Synchronization of human assembly task models in virtual manufacturing systems using deep active learning

The effective and accurate modeling of human actions in Cyber Physical Systems is one of the key technologies in virtual/smart manufacturing systems. One challenge is obtaining data, but virtual reality has the potential to make human manufacturing experiments more practical. In this paper we propose a framework to simplify human assembly task experiments by reducing the data required and automating the adjustment of experimental factors. In VR experiments involving throughput rate a deep active learning model was used to reduce the amount of data required, thereby speeding up the experiment. The proposed method can yield quick/accurate generation of human action models in virtual systems. Potential human errors and delays in assembly processes are successfully reproduced and demonstrated in this concept.

Keywords - Human modeling, Virtual reality, AI and machine learning, Digital transformation

1. Introduction

Modern manufacturing systems include physical system, data-acquisition, and simulation components [1]. Human integration is a key factors affecting adoption [1], [2]. There has been a desire to move towards human-centric production for years [3]–[5], but gathering data and modeling behavior of humans is substantially more difficult than their machine counterparts due to practical and ethical reasons.

1.1 Human data acquisition

Data acquisition for human processes is useful for safety, prediction, and diagnosis. Industries where human operators are static (e.g. long distance drivers and pilots) has seen commercial success in applying sensor-based systems, most likely due to sensor placement. Some examples are sensors placed in chairs, steering wheels, operator facing cameras, etc [6]–[13].

Manufacturing typically requires operators operate dynamically in unstructured environments and therefore has seen less commercial advancements despite attempts. Some work using medical equipment is promising but prohibitive due to sensor practicality and cost (EEG or EKG) [14]. Taking biological samples offers valuable insights but poses practical challenges [15]. Investment into sensors can be a risk as it is not clear what the data will reveal. [16] mentions that modeling can address this by providing sample data, but also says these models must be compared against empirical human-in-the-loop data, creating the chicken and egg dilemma.

Virtual reality (VR) has beneficial properties that address many of these issues. Using head mounted displays and controllers we can simulate rich interactions between humans and the manufacturing environment without the need for additional sensors. VR has been used extensively for product development [17] and visualizing and planning manufacturing systems and layouts [18]. Here, the interest is in planning data acquisition systems for human assembly tasks.

[19], [20] developed VR frameworks for psychology trails. [20] provided a framework for planning experimental trails based on the selected factors, with the additional functionality of conducting remote experiments by storing data on a remote database. [19] developed a user friendly framework for experimental design and conducting experiences, requiring little knowledge of computer programming. Where both of this work simplifies experimental design, this work investigates outsourcing this responsibility to a model.

1.2 Human modeling

Human performance models (HPM) predict human behavior in a task or system. Since human behavior is complex, simplified models are constructed in specific domains. These models are useful for designing systems [16]. Historically, these models where used to identify factors affecting performance, thereby enabling ergonomists to design systems that optimize human performance [21].

Computational models like ACT-R [22] have shown to be successful in predicting human performance, but require years of programming experience. Therefore, analytical models are more common. For example, Wright’s learning curve models how human learning reduces the task duration using logarithmic/exponential functions. It was initially used in predicting the throughput rate in human assembly tasks [23] and later received numerous additions/modifications considering work induced fatigue, and rest schedules [24]–[27].

Recently, an interest to incorporate technologies like artificial intelligence and big data due to the changing nature of work [28]. [29] highlights automatic extraction of practical information from heterogeneous data (a capability of deep learning) and real-time data collection as key enablers of including digital human models into manufacturing CPSs. [30] used K-nearest neighbors to classify tasks by skill level requirement, allowing hiring managers to select operators with the appropriate skills. [31] used an Artificial Neural Network (ANN) to model the relationship between the work environment, worker personalities, and their subsequent performance, but did not provide a quantifiable measure of the accuracy. An ANN was explored in this research for the convenience of modern tools and trends.

Since modeling and data-acquisition for experimentation are interconnected, using the model to actively suggest the experimental conditions can be beneficial.

1.3 Models

A number of previous works look into selecting the next experimental data-point intelligently.

1.3.1 Design of experiments

Experimental design is the selection of factor-level combinations to be tested and the conscious exclusion/blocking of others. This is mostly concerned with the measuring process. Design of experiments (DOE) builds on this by including a protocol for analyzing the data and selecting the next experimental point [32]. DOE methods are well known for their usefulness. Some examples are factorial design, ANOVA, and response surface. These methods are limited to situations with few factors, few factor-levels, and constant random error. Traditionally, linear regression models are used. More recently, iterative design [33] has shown promising results. The decision to use a multi-layer-perceptions in this work was motivated by the potential to lift the limitation regarding the number of factors.

1.3.2 Pool-based active learning

Active learning refers to having the model choose the next action. In pool-based active learning, samples from an unlabeled pool are labeled for classification. Gathering unlabeled images from the internet is easy, but the act of labeling the data is “expensive” as it requires human effort. The goal is to maximize a models performance with the fewest labeled samples possible [34]. Two pools of data are used; the training data pool consists of labeled data, and a data-bank pool consists of unlabeled data. Algorithms search through the unlabeled pool selecting a sample that will most improve the models performance. The acquisition function returns the samples usefulness. A (human) oracle labels the selected sample.

Pool-based active learning for classification problems are by far more popular than experimental intervention, with modern literature equating active learning with pool-based algorithms [34]–[36]. There is a comparatively less work considering regression [37].

One should note that regression problems can fit into pool-based active learning, where labeling data refers to populating real-valued targets and the unlabeled data-bank would be an infinite pool in the input space.

1.3.3 Reinforcement learning

Another relevant framework is reinforcement learning, which is concerned with solving sequential decision-making problems. However, this is more general and not often related to experimental design. [38] investigated a tightly related topic “The design of experiment using reinforcement learning”. This work was limited to the use of computer simulations.

Some notable work uses reinforcement learning to search for neural-network architecture [36], [39], [40]. This is particularly applicable here because (1) the model is often retrained between sampling/experiments, and (2) the model may need to change as more data becomes available.

1.3.4 Acquisition functions

The acquisition function returns a samples usefulness, selecting which sample to select from the pool. The acquisition function balances a number of concerns when selecting the next point. Optimal experimental design [41] formalizes selecting the next point that minimizes the variance of estimators, in-turn selecting the point with the most information-content. One interpretation of this is selecting the point with the highest expected model change [42]. Another is computing variance query by committee [43], where multiple models predict the same value.

Another concern is diversity of samples. [37] applied a two stage process to regression prediction of driver drowsiness. Using a two stage process that (1) sampled point “far apart”, then (2) recalling the previous points, selects a new point based on the centroid of the distributed points.

[44] proposes passive sampling where the acquisition function is based on a separate (non-learning) model, not requiring re-training at each iteration. This work also found that passive sampling achieves more stable performance, avoiding fluctuations from selecting samples with the highest regression errors.

Most work assume constant noise, in human modeling that is not always a valid assumption.

1.3.5 Model uncertainty

There are three factors that contribute towards uncertainty: poor model specification, training data variance (noise), and sparsely represented data (epistemic uncertainty) [45]. We can ignore the first factor assuming our (ANN) model can sufficiently model the unknown system.

The distinction between noise and epistemic uncertainty is important. Given an infinite amount of data, the (epistemic) uncertainty will tend to zero, while the noise will not. Noise does however affect the amount the data required to reduce uncertainty. Typically, the noise is assumed to be constant but this is not always the case. There are a number of regression models that estimate uncertainty, these include Bayesian neural-networks [46], [47] , bagging/bootstrap [48], dropout-techniques [49] and Gaussian process regression [46], [50] and ensembles (Query by committee) [43].

This work uses the acquisition functions to address the issue of high-noise regions.

1.4 Summary

Using VR to conduct experiments allows data-acquisition during the prototyping phase, reducing the risk of investing into sensors. Using active learning the model decides the next experimental point-of-interest intelligently, reducing the amount of data required. Due to the VR experiment being software defined, the experiment factors can be automatically adjusted by the model, reducing the labor required and errors introduced by the experiment conductor.

The rest of this paper is organized as follows, section 2 covers the experimental methods, namely the VR simulation used to acquire data, the selected case study, and the data experiment used to show the reduction in required data; section 3 covers the algorithm and some common choices that addresses the issues raised by this case study; section 4 provides the results of the VR simulation and the data experiment; section 5 provides a briefly summary.

2. Virtual simulation and experiments

The aim of the experiment was to show that active-sampling will require less data than random sampling. To this end human operators completed a series of assembly tasks in VR where the data was gathered. Ideally one would conduct two separate VR simulations, but this would require more data. Instead a two step process was used. Firstly, the data was gathered from a VR simulation. Next, in a sampling experiment the aforementioned data represents a data-bank where training samples are selected using active-sampling or random sampling. This approach reuses the VR simulation data, thereby reducing the experimental labor.

2.1 Virtual reality simulation

2.1.1 Case study selection

Wright-learning was selected as the case study for the experimental task. As mentioned, the Wright-learning curve models the duration of a task.

This case study was selected because it presents an approachable model, that is easy to depict graphically due to low dimensionality, easy to relate to since we all experience its effects in daily life, and it is useful for predicting throughput rate in manufacturing systems. The challenges it presents due to having varying noise is interesting because it may not be visible in higher dimensional problems. Recall that if this process was modeled for a typical machine, the throughput rate could simply be approximated with a constant mean value, further illustrating how much more challenging modeling human performance is compared to machine performance.

To our knowledge no-one has previously looked at the data-variance of Wright-learning. We suspect due to the impracticality of gathering low-level data.

2.1.2 Experimental design details

The experimental procedure is described in appendix A for interested readers. An explanatory video is provided here. In summary the subjects performed four common assembly tasks in VR and the duration of each task was measured.

2.2 Sampling data experiment

The main objective is to reduce the number of experiments conducted. To do this we compare random sampling using a Uniform distribution across the input space, with active learning, having the model select the next experimental point. It was assumed that random samplings performance would be representative of simpler experimental methods such as Latin squares.

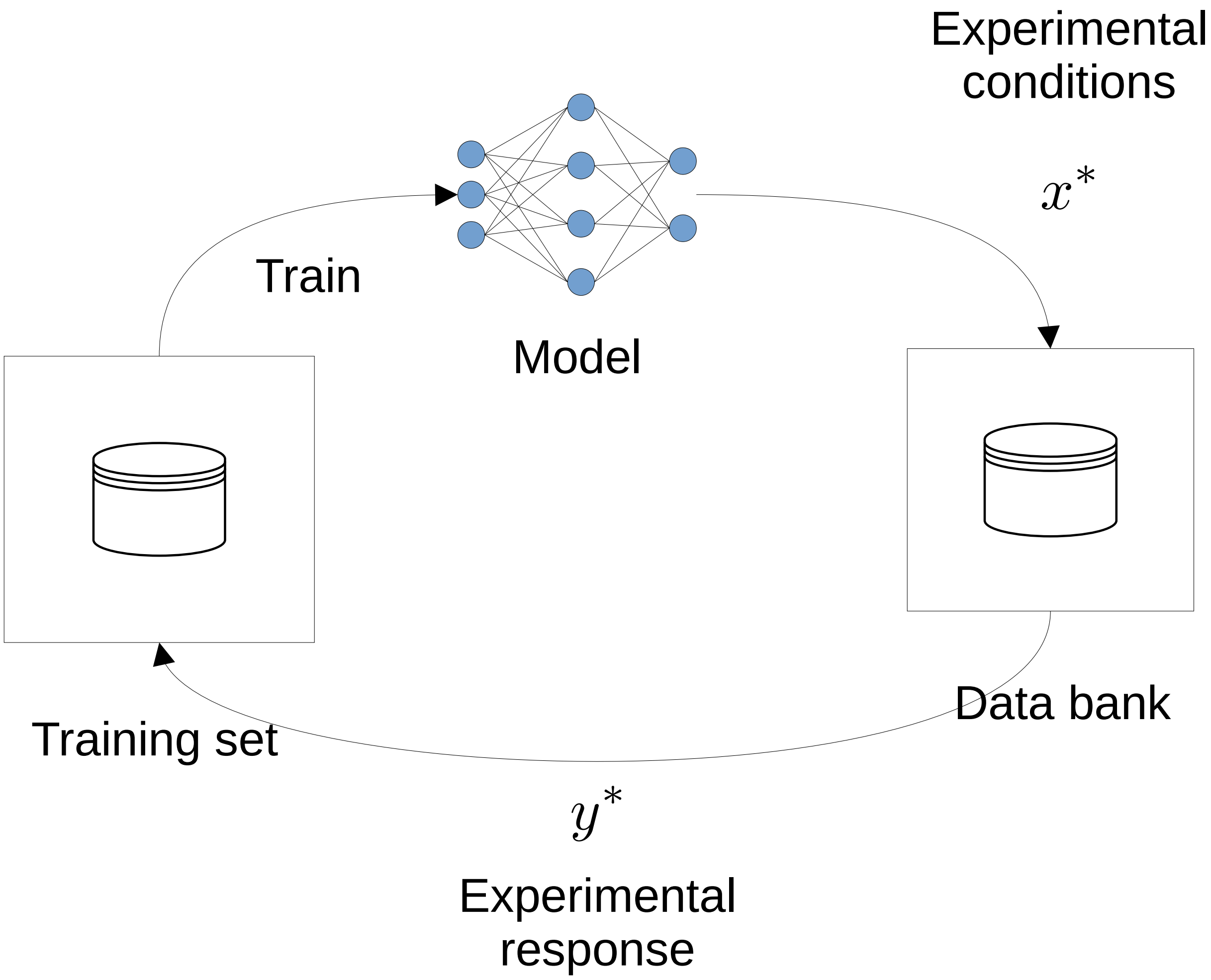


Figure 2: The sampling experiment. The data bank represents the experimental data.

Active experimental sampling was simulated by separating a training data bank and a bank of experimental data. Initially, the model is trained on a small training set, having the model select the next experiment's conditions (data-point). We then append the selected point to the training bank.

In this experiment we start with having all data-points from one task and no data-points for the other tasks. We then sample points from these unknown tasks. We measure the performance using the MSE error of a 30% cross validation set at each iteration. As more data from the unknown tasks are sampled, the model accuracy for these tasks increase, reducing the error. We expect to see active sampling converge to the lower error limit with less data than random sampling. Due to the random nature of the algorithm we conduct a number of experimental runs. Due to the high noise region, we assume that some data from a similar assembly process is available to locate these regions and improve training data.

3. The algorithm

The active learning algorithm used, follows a straight forward loop. Starting at some initial point-of-interest x^*, it samples from the unknown system. The sequence is as follows:

1. Conducting an experiment at inputs x^*.
2. The data is then appended to our data-set.
3. The model is trained using the current training pool.

4. The next sampling point-of-interest x^* is determined.

5. This process is repeated until we reach a termination condition.

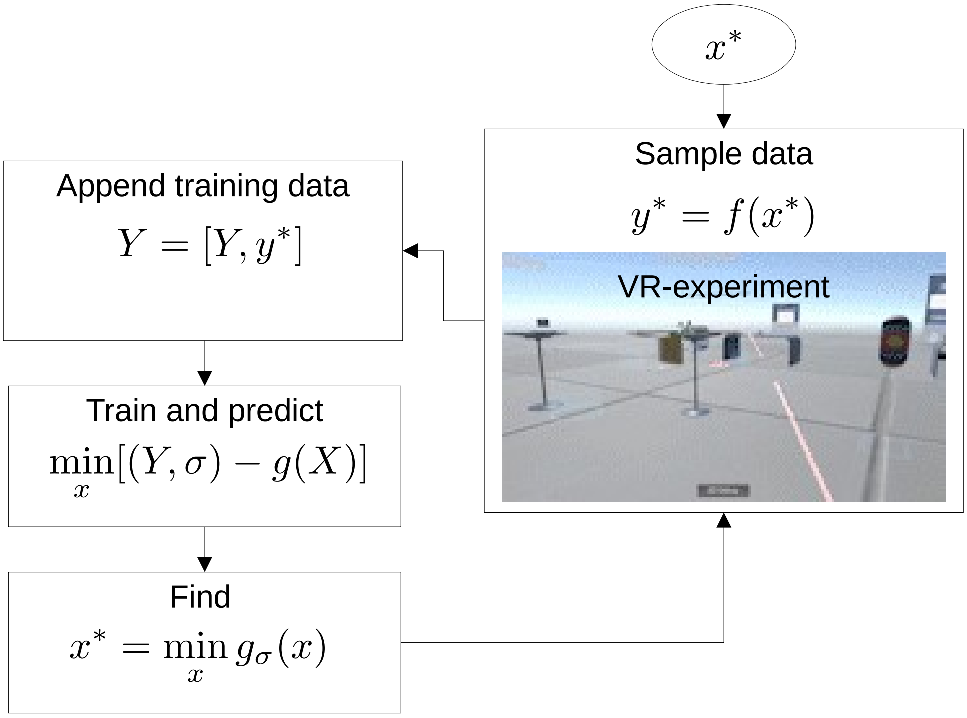


Figure 4: The main loop of the algorithm as applied here. We exclude the termination condition. Later we replace our acquisition function g_σ.

3.1 Training the model

We require a model that predicts the mean response and epistemic uncertainty .

$$(y, \sigma) = g(x)$$

3.1.1 Ensemble models

Ensembles are based on Query by committee [43] where multiple models predict the same value. The main idea is that if the data sufficiently describes the behavior, all models will predict the same outcome, if there is some discrepancy in the results, it can be interpreted as uncertainty. The results are then combined to attain the mean and variance [51], [52]. See [53] for a recent review.

The mean and uncertainty can be determined using the equations below. The mean being the average of the model prediction, and the uncertainty being the mean distance between the mean-prediction and the model prediction.

$$\mu(x) = \frac{1}{B} \sum g(x)$$

$$\sigma^2(x) = \frac{1}{B-1} \sum \mu(x) - g_i(x)$$

In this work, a more convenient (local) ensemble method with two main differences is used. Firstly, instead of using multiple networks, a single neural network predicts multiple mean-predictions. Secondly, there is no need to split the data. We suggest using this model as an entry point for ease of use, but suggest other ensemble methods for practical applications, and more experienced deep learning practitioners. The figure below shows the two configurations.

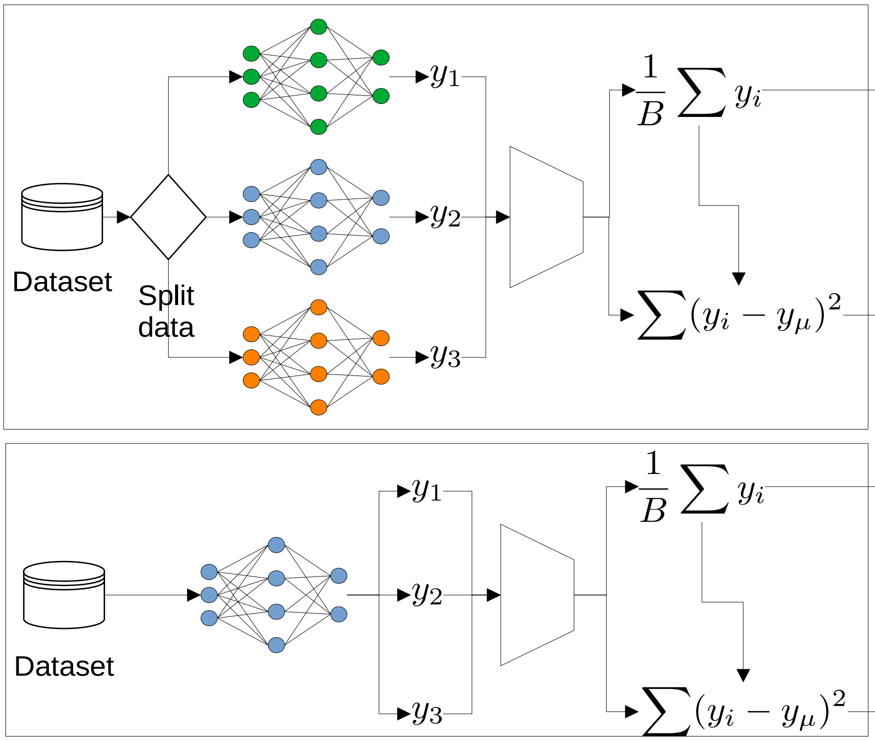


Figure 5: Conventional (Global) ensemble methods (top) train multiple models on random subsets of the data. Below, the local ensemble method is simpler and more convenient. Note that local ensemble can be used within Global ensemble methods, hence the two are not exclusive. We do not claim that this method yields better results, and suspecting this model will not scale well with dimensionality, number of ensembles, inputs, and outputs due to the interaction between neurons.

Recall that each model is randomly initialized, so given the same data and training process, they may predict different outcomes.

3.1.2 Loss function

The loss function must now quantify the mean error and uncertainty. The well known negative log likelihood [54] loss function was used with minor modifications. Gaussian noise is assumed.

$$L = \frac{n}{2}[log(\sigma^2) + |\frac{y_\mu - y_t}{\sigma^2}|^2]$$

The algorithm that follows show that the loss is calculated for ensembles.

Return Loss L_e :

$$y_\mu = \frac{1}{B}\sum y_i$$
$$\sigma^2(x) = \frac{1}{B-1}\sum y_\mu - y_i$$
$$L_e = \lambda log(\sigma^2) + \sum|\frac{y_\mu - y_i}{\sigma^2}|^2$$

Given the model and loss function we now have a model that predicts both mean and uncertainty $(y, \sigma^2) = g(x)$. Occasionally we represent the mean-prediction as $y = g_y(x)$ and the uncertainty prediction as $\sigma = g_\sigma(x)$. These refer to the same model $g(x)$.

3.2 Sampling strategies

Sampling strategies refer to how we capture the data from our unknown system. In practice we acquire data from experiments, but during development other sampling strategies are helpful. In the general case we know x^* and sample output (y) from the unknown system f.

$y^* = f(x^*) + \epsilon$

3.2.1 Experiment sampling

Usually an experiment is conducted with the selected experimental factors (x^*). Here we need to consider the experimental constraints, typically expressed as some range of inputs e.g. $x \in (0,60)\ lumens$. Instead of addressing constraints here, we delegate the responsibility of constraints to the acquisition function.

3.2.2 Function sampling

A generative function was used to sample without restriction. This allowed testing different noise conditions, systems of different "*curviness*", and edge cases. This strategy can be used for illustrative purposes.

3.2.3 Data-bank sampling

A third but necessary sampling technique is sampling from an existing data-bank of previous experimental data. This may be useful for testing the framework using existing data sets. This resembles the aforementioned DAL framework by using a pool of unlabeled data. However, we search the input-space and not the pool of examples.

These techniques are particularly useful because often we want to know more about our system before conducting experiments. The case study illustrates why we cannot rely on a fire-and-forget experiment but need to first explore existing data.

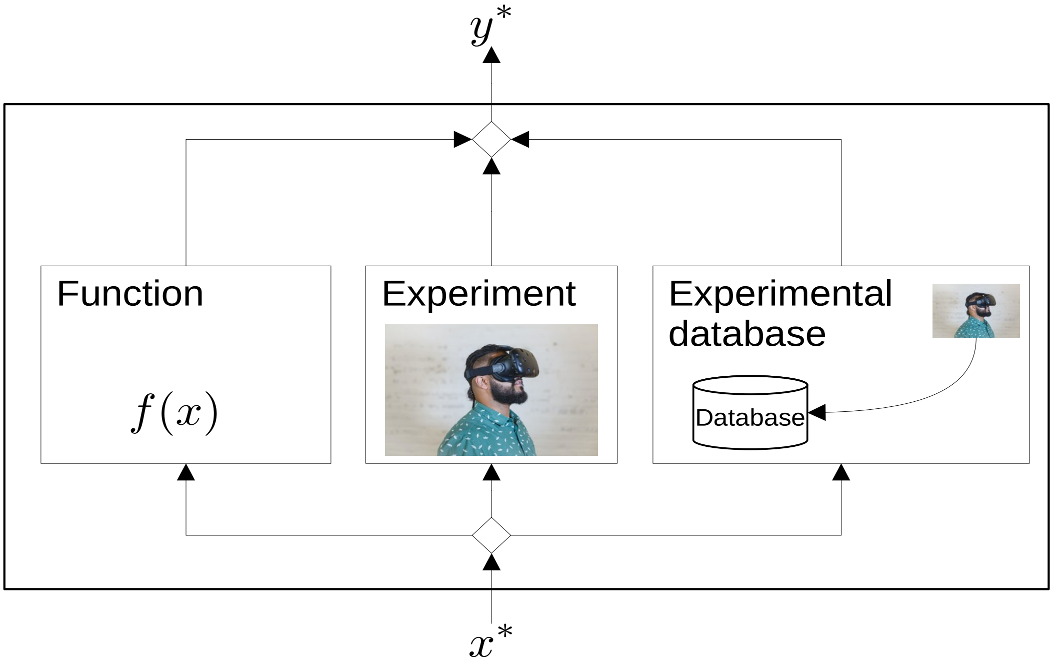


Figure 6: Sampling strategies. Starting from left, we can sample from a "noisy" system function, then from the experiment itself, finally from a bank of previous experimental data.

3.2.4 Other sampling considerations

Whether to sample once every loop or gather multiple samples, depends on the model training time and the effort of configuring the experiment. We leave those up to the practitioner(s).

Naturally, x^* is a random variable, and here we simply transformed it into a uniform distribution with a small width, leaving yet another tuning parameter. A more intelligent solution would use the information (height of $\sigma(x)$ on m-dimensional surface) to determine the characteristics of the sample distribution.

This solution presents many tuning-parameters, but a current trend in research is to automate the selection of these parameters [35], [39].

3.2.5 Retraining every iteration

A question that arises is whether to reset the model or continue training between iterations? In our experience, resetting the model leads to more diverse results due to random initialization of the networks. This comes at the cost of increased training times with each iteration. As the training time cost is often small compared to the experiment costs, this was an obvious choice in this case.

As the data set grows larger, resetting the model will require more careful consideration due to an increase in training times. [34] highlights the ability to continue training as a feature of some algorithms. This should be considered in time sensitive applications. On the other hand, [36], [40] show that active learning can be used to optimize the model architecture, which may be applicable at these iterations.

3.3 Determining the sample point

A naive acquisition function would determine x^* to be where the uncertainty is the highest. This turns out to be a reasonable approach but does not account for experimental design constraints, high-noise regions, and regions of interest.

3.3.1 Search and utility

In optimization we formulate this as a search problem. The search is not governed by our constraints and may suggest points outside of our experimental range. Utility was used to incorporate experimental constraints into the acquisition function. $Util(x) = \sigma(x) * c_1(x) * c_2(x) * ...$ from uncertainty and constraints. We maximize our utility instead of uncertainty using optimization $x^* = max(Util(x^*))$. The figure below shows the use of step functions to incorporate constraints. Utility allows the balancing of multiple concerns within the acquisition function.

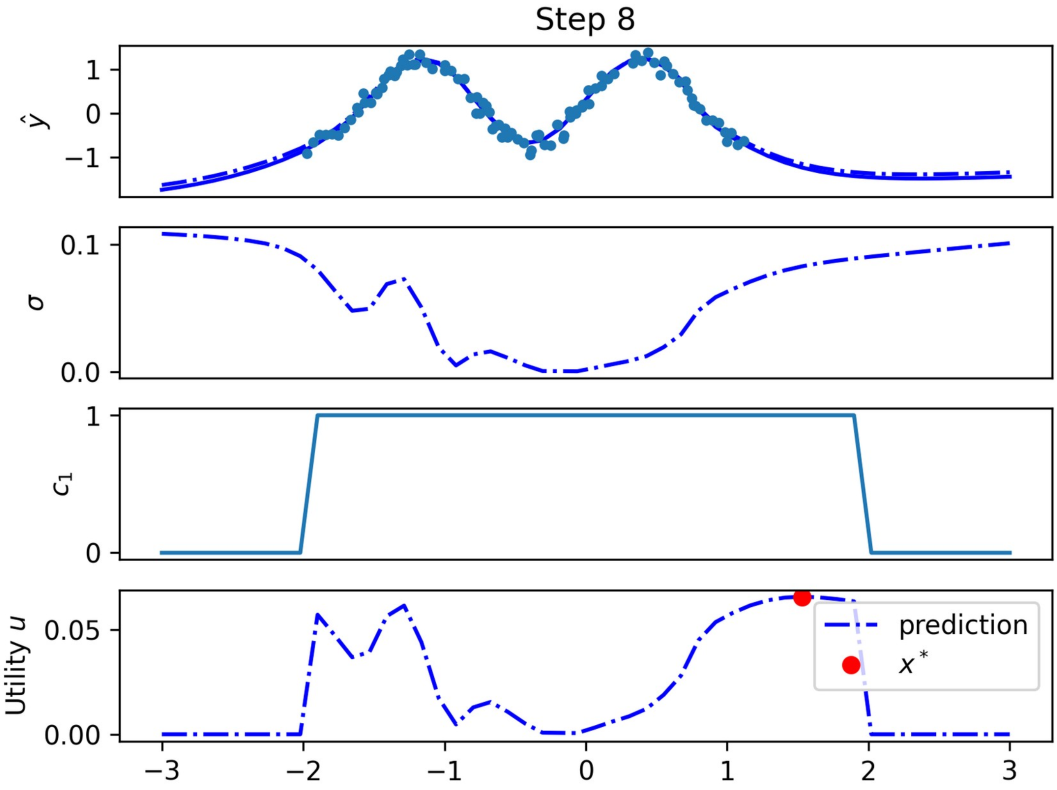


Figure 7: The use of step function in utility to constrain our selection of x^* within the experimental design range. $f(x) = sin(4x) + \sigma$

It can be seen that the utility is shaped by multiplying these functions. We typically design the experimental consideration functions to be in a range of 0 to 1, e.g. $c_i(x) \in [0, 1]$. One can also see that due to our constraints the maximum utility will always occur within our experimental range[1].

3.3.2 Some more interesting uses of utility

Utility is used to retain a level of control by restricting the experimental range, or biasing the decisions of the algorithms next point. Below we include some examples.

3.3.2.1 Region of interest

There is often a region of interest where we want to bias the selection of x^* to areas near this region. During testing we found this useful for biasing x^* away from regions of high noise. A candidate function is shown below, where p is the point of interest. Note that a region need not be a point but could be a plane or other geometric feature. Refer to the figure that follows.

$c(x) = \frac{\beta^2}{\beta^2 + (p-x)^2}$

These functions are based on the geometric position and therefore require the user to have knowledge or insight of the process. A modern trend in artificial intelligence is to reduce the decision burden from the practitioner, making this less attractive.

[1] Excluding when the utility function or constraints are zero functions.

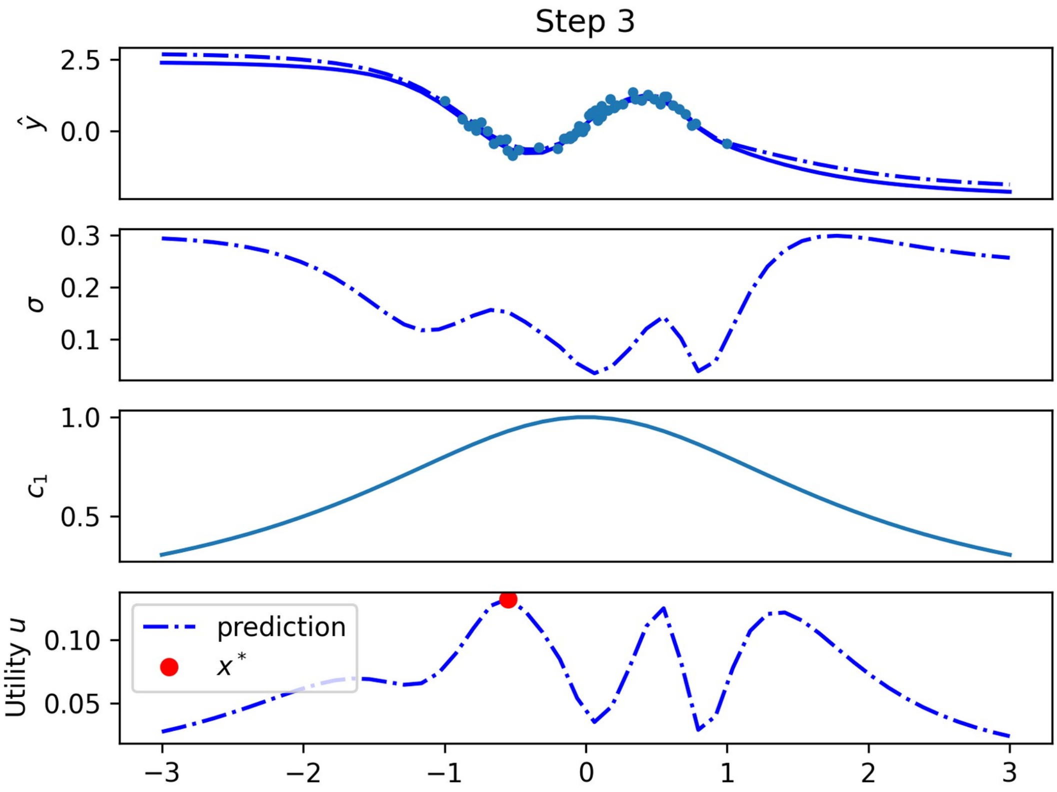


Figure 8: The utility is biased to sample nearer to the region of interest (0). Points far from this area will not be considered.

3.3.2.2 *Limiting the change*

In some cases we would like to bias the next sample to be geometrically near to the previous sample. Consider the example of a process involving varying the temperature or feed speed to find ideal operating conditions. One would prefer to change these factors only slightly to avoid complications as a result of irregular throughput, heating energy costs, and temperature fluctuations. In this case we encourage points that are near to the current operating point. e.g $c(x) = \frac{\beta^2}{\beta^2+(x_c-x)^2}$

3.3.2.3 *Diversity inclusion*

On the other hand, we may want to sample points that are further away for some reason. In this case we penalize points that are close together. e.g $c(x) = C^{-\frac{\beta^2}{\beta^2+(x_c-x)^2}}$.

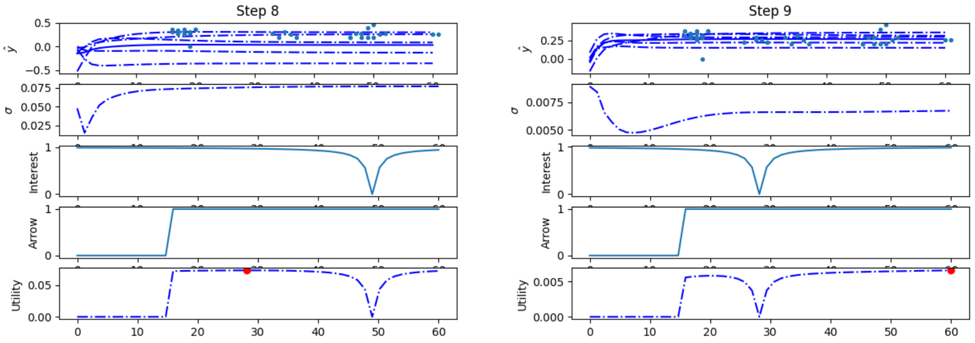


Figure 11: Shows how we can encourage diversity in samples by encouraging distance between consecutive points.

3.4 Loop termination

Some good choices on when to stop sampling experimental data are after a predetermined number of iterations, or once the uncertainty is below a threshold. Alternatively, some data can be selected for cross validation and an error threshold can be used. When selecting a threshold for termination it should be noted that results will differ between runs. This is due to random initialization of the models and random training batches. In addition, a models uncertainty will not necessarily decrease monotonically between iterations.

4. Results

The question we attempt to answer here is “Does active sampling reduce the amount of experiments required to model the throughput rate of an unknown human assembly process, given data from a similar task?” After conducting simulations using the data acquired from VR experimental trials, random and active sampling were compared to see which converges to the minimum error with fewer data-points.

4.1 Virtual manufacturing experiment

When examining the data gathered from the VR experiments (figure below), we see the variance is largest in the beginning when the subject is first learning the task. This is true for both inter-subject and intra-subject (inter-repetition) variance and can be interpreted as “operators start at different levels, but after practice tend towards the same performance” and “operator tend to take more risks at the start, but find reliable methods with experience”, respectively. Also note that the data is not normally distributed as a task is more likely to take longer.

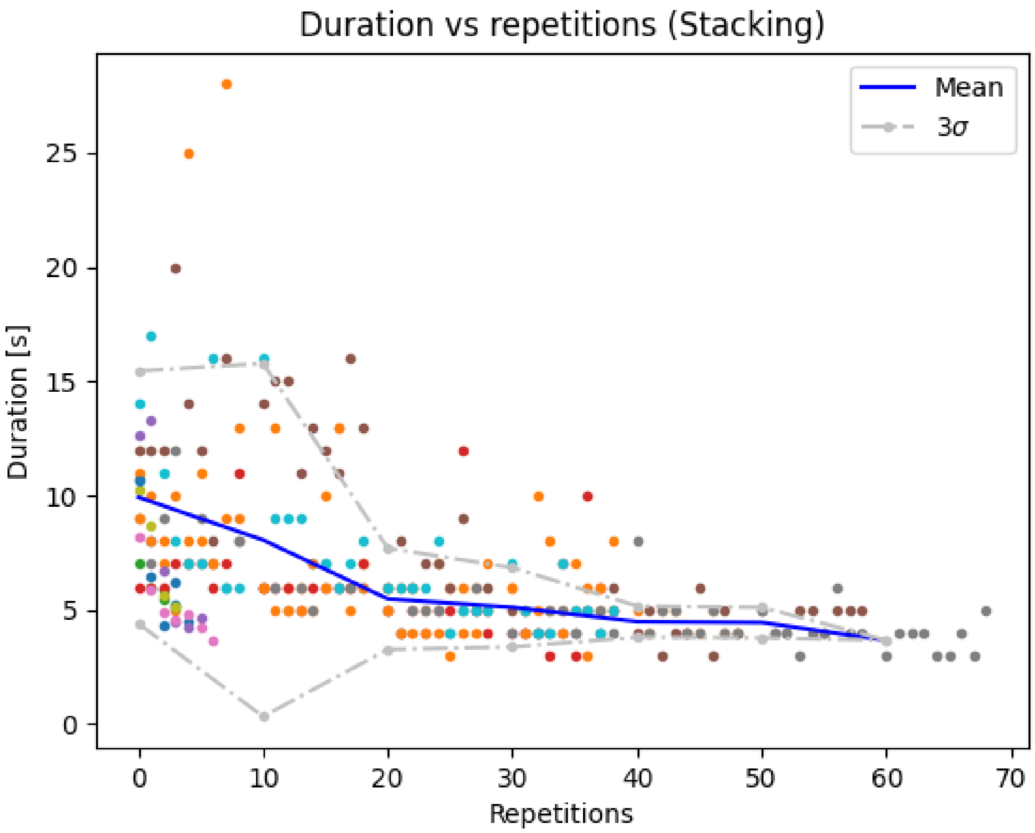


Figure 1: The durations for the stacking, the mean and variance is taken across each trial (10 repetitions). The different color dots correspond to human subjects. An operator is likely to have a task take longer than shorter, therefore a Gaussian distribution is not accurate. We make this assumption for convenience.

This large variance at the start means more data will be sampled in this region. During development we noticed the algorithm would stall, only sampling points within this region. Utility was introduced to address this by influencing the algorithms decision.

As previously stated, the model is given all data-points from one task (task 0) and no data-points for the other unknown tasks. It then samples points from the unknown tasks.

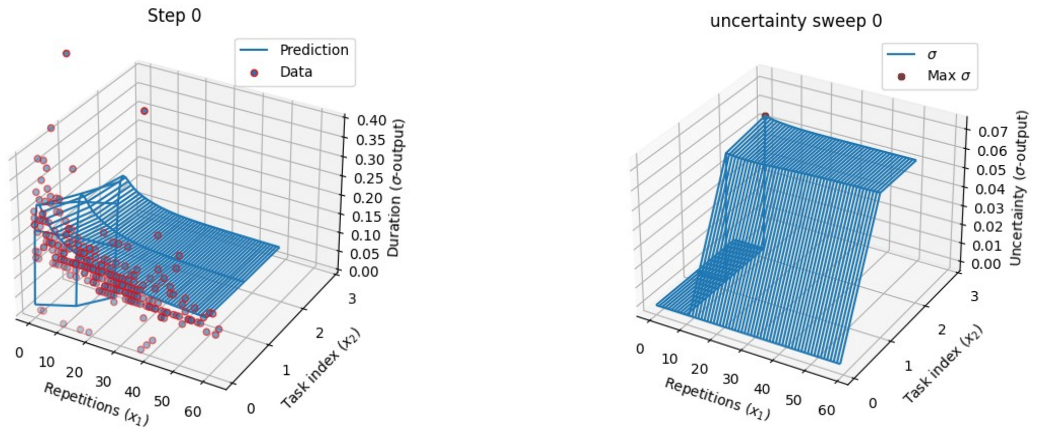


Figure 8: The predicted mean for each task (left) and utility (right), show the exclusion of the “data hungry” region, setting these values to zero. The scale gives the impression of a flat plateau, but that is not the case.

4.2 Sampling experiment results

When comparing active sampling and random sampling we found active sampling converged to a lower error with less data than random sampling, as expected. The figure that follows illustrates this. Additionally, we see that active learning has a lower variance between multiple runs, resulting in more stable learning.

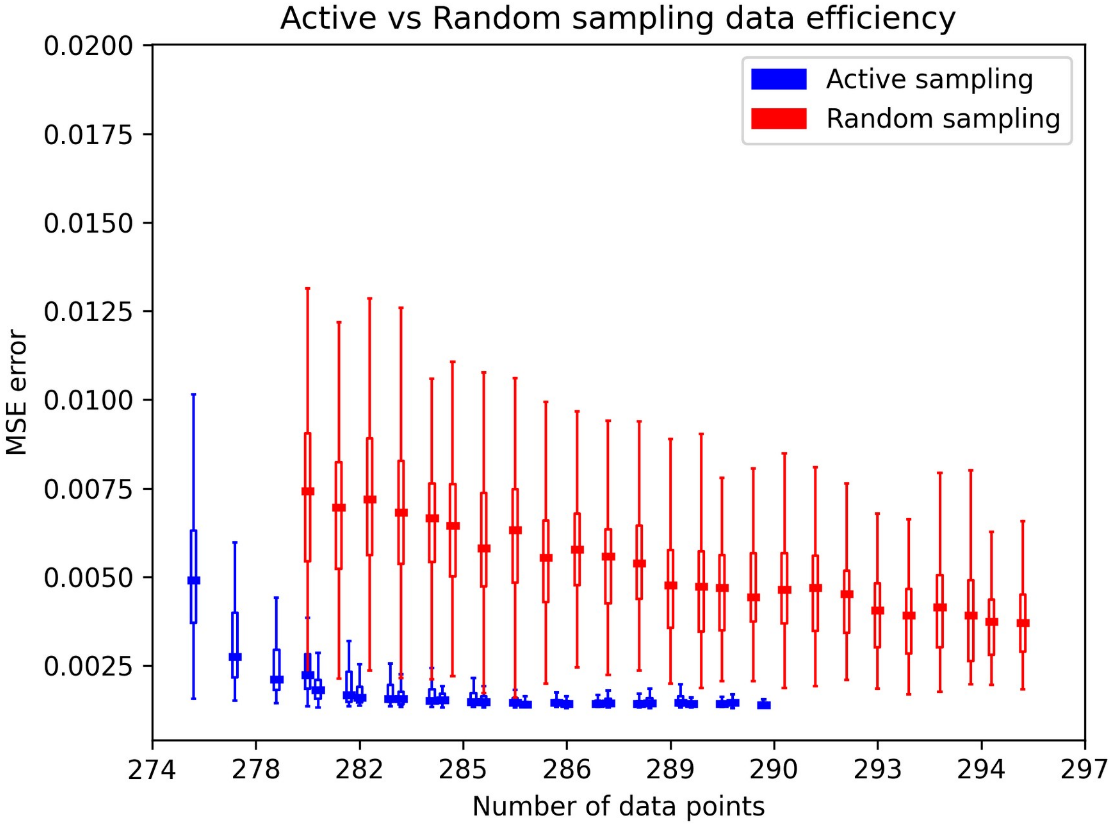


Figure 9: The experimental results of cross validation error of the new tasks for random vs active data-point sampling. Note how active sampling converges to a low error quicker than random sampling. Also notice random sampling has a substantially higher variance.

4.3 Discussion

4.3.1 Active sampling results

Deep-active sampling attained lower prediction errors with similar data samples when compared to random sampling. This implies that it could reduce the amount of experimental trials conducted.

Active sampling resulted in lower variance between multiple runs, implying more stable learning. This is likely due to uncertainty-based sampling.

Utility based acquisition functions were successful at balancing multiple sampling concerns, including experimental design constraints and high-noise regions.

4.3.2 Virtual reality simulation results

It was shown that VR simulations can measure the wright-learning effect. To our knowledge this has not been done before. Due to skepticism in VR experiments, first confirming that the phenomena can be observed is necessary.

VR simulated the process and gathered the data at a high-fidelity, logging the duration of each component and not the mean duration across the entire batch. This allowed exploring data variance in this research. An alternative to VR would be investing in sensors (electronic devices to gather and record data) and hardware/props for assembly tasks. This would also require manually configuring the experiments between trials.

Combining active learning with the software nature of VR simulations we are able to further reduce the burden by reconfiguring experimental factors at runtime, adjusting for ergonomic factors, and measure data with high fidelity. For instance, the point-of-interest could contain task-repetitions and lighting conditions that could automatically be adjusted by the model.

4.3.3 An ideal solution

Let us take this opportunity to speculate on how a future human experiment application could look, using our findings here and tools from literature.

There is less need for a domain expert as deep learning models can extract useful predictions from heterogeneous data. Design of experiments could be replaced by an active learning model selecting the experimental conditions at runtime. Prototyping human assembly systems using VR would postpone the need for sensors and hardware. This is also true for retrofitting existing systems.

Remote VR experiments will alleviate some of the effort of experiment conductors. A new level of scalability is possibly with remote VR experiments. Developers are not expected to be modeling experts when reinforcement learning can be used for neural architecture search.

This makes it a useful tool as it substantially reduces the burden of experimental trials and requirements of its users.

4.3.4 Limitations and future work

The case study illustrated that active sampling should not be used blindly, due to high-noise or “data hungry” regions. This would result in a worse performance than random sampling. In most cases, existing data from a similar process should be available to determine whether these regions are present within in the system.

The current model does not evaluate how much a factor contribute to the response. Measuring this effect size is a desirable feature that ergonomists typically use to improve the systems performance. The data acquired here can be used with a separate model to investigate the contributing factors.

This naturally leads to the question of whether active learning can be used to optimize a process, a topic which requires further investigation.

5. Conclusions

This work investigated whether active sampling can be used to efficiently build models of human assembly processes, by reducing the amount of experiments conducted in a VR environment.

We saw that human systems can have high noise regions, causing active sampling to stall. Approaches to mitigate this issue were discussed. We suggest a practitioner investigate the noise characteristics of a similar process, to determine which strategies work. A good addition to this work would be the automatic realization and mitigation of “data hungry” regions.

Deep active sampling is particularly exciting in software based environments like VR, where adjustment of the experimental configurations can be automated. This may also be the case for other human software interfaces.

Acknowledgments

This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government (MOTIE, 20193310100030), and the Nuclear Safety Research Program through the Korea Foundation Of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea. (No. 2003020)

6. Works Cited

[1] P. Leitão, J. Barbosa, A. Pereira, J. Barata, and A. W. Colombo, “Specification of the PERFoRM architecture for the seamless production system reconfiguration,” *IECON Proc. (Industrial Electron. Conf.*, pp. 5729–5734, 2016, doi: 10.1109/IECON.2016.7793007.

[2] S. Karnouskos and P. Leitao, “Key Contributing Factors to the Acceptance of Agents in Industrial Environments,” *IEEE Trans. Ind. Informatics*, vol. 13, no. 2, pp. 696–703, 2017, doi: 10.1109/TII.2016.2607148.

[3] A. Kolus, R. Wells, and P. Neumann, “Production quality and human factors engineering: A systematic review and theoretical framework,” *Appl. Ergon.*, vol. 73, no. October 2017, pp. 55–89, 2018, doi: 10.1016/j.apergo.2018.05.010.

[4] F. Fruggiero, S. Riemma, Y. Ouazene, R. Macchiaroli, and V. Guglielmi, “Incorporating the Human Factor within Manufacturing Dynamics,” *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1691–1696, 2016, doi: 10.1016/j.ifacol.2016.07.825.

[5] M. Yung, A. Kolus, R. Wells, and W. P. Neumann, “Examining the fatigue-quality relationship in manufacturing,” *Appl. Ergon.*, vol. 82, no. August 2018, p. 102919, 2020, doi: 10.1016/j.apergo.2019.102919.

[6] S. Kerick, J. Metcalf, T. Feng, A. Ries, and K. McDowell, “Review of Fatigue Management Technologies for Enhanced Military Vehicle Safety and Performance,” *Tech. Report; U.S. Army Res. Lab.*, no. September, 2013.

[7] C. C. Liu, S. G. Hosking, and M. G. Lenné, “Predicting driver drowsiness using vehicle measures: Recent insights and future challenges,” *J. Safety Res.*, vol. 40, no. 4, pp. 239–245, 2009, doi: 10.1016/j.jsr.2009.04.005.

[8] H. B. Kang, “Various approaches for driver and driving behavior monitoring: A review,” *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 616–623, 2013, doi: 10.1109/ICCVW.2013.85.

[9] A. Sahayadhas, K. Sundaraj, and M. Murugappan, “Detecting driver drowsiness based on sensors: A review,” *Sensors (Switzerland)*, vol. 12, no. 12, pp. 16937–16953, 2012, doi: 10.3390/s121216937.

[10] K. Ransikarbum, N. Kim, S. Ha, R. A. Wysk, and L. Rothrock, “A Highway-Driving System Design Viewpoint Using an Agent-Based Modeling of an Affordance-Based Finite State Automata,” *IEEE Access*, vol. 6, pp. 2193–2205, 2017, doi: 10.1109/ACCESS.2017.2782257.

[11] D. F. Dinges, G. Maislin, R. M. Brewster, G. P. Krueger, and R. J. Carroll, “Pilot test of fatigue management technologies,” *Transp. Res. Rec.*, no. 1922, pp. 175–182, 2005, doi: 10.3141/1922-22.

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

[13] H. J. Baek, G. S. Chung, K. K. Kim, and K. S. Park, "A smart health monitoring chair for nonintrusive measurement of biological signals," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 1, pp. 150–158, 2012, doi: 10.1109/TITB.2011.2175742.

[14] Z. Sedighi Maman, M. A. Alamdar Yazdi, L. A. Cavuoto, and F. M. Megahed, "A data-driven approach to modeling physical fatigue in the workplace using wearable sensors," *Appl. Ergon.*, vol. 65, pp. 515–529, 2017, doi: 10.1016/j.apergo.2017.02.001.

[15] E. Bal, O. Arslan, and L. Tavacioglu, "Prioritization of the causal factors of fatigue in seafarers and measurement of fatigue with the application of the Lactate Test," *Saf. Sci.*, vol. 72, pp. 46–54, 2015, [Online]. Available: http://dx.doi.org/10.1016/j.ssci.2014.08.003

[16] A. Sebok, C. Wickens, and R. Sargent, "Using meta-analyses results and data gathering to support human performance model development," *Proc. Hum. Factors Ergon. Soc.*, pp. 783–787, 2013, doi: 10.1177/1541931213571171.

[17] G. Lawson, D. Salanitri, and B. Waterfield, "Future directions for the development of virtual reality within an automotive manufacturer," *Appl. Ergon.*, vol. 53, pp. 323–330, Mar. 2016, doi: 10.1016/J.APERGO.2015.06.024.

[18] W. Dangelmaier, M. Fischer, J. Gausemeier, M. Grafe, C. Matysczok, and B. Mueck, "Virtual and augmented reality support for discrete manufacturing system simulation," *Comput. Ind.*, vol. 56, no. 4, pp. 371–383, May 2005, doi: 10.1016/J.COMPIND.2005.01.007.

[19] J. Grübel, R. Weibel, M. H. Jiang, C. Hölscher, D. A. Hackman, and V. R. Schinazi, "EVE: A Framework for Experiments in Virtual Environments," 2017, pp. 159–176. doi: 10.1007/978-3-319-68189-4_10.

[20] J. Brookes, M. Warburton, M. Alghadier, M. Mon-Williams, and F. Mushtaq, "Studying human behavior with virtual reality: The Unity Experiment Framework," *Behav. Res. Methods*, vol. 52, no. 2, pp. 455–463, Apr. 2020, doi: 10.3758/s13428-019-01242-0.

[21] J. L. Harbour, "Human performance modeling: A case study," *Perform. Improv.*, vol. 49, no. 8, pp. 36–41, Sep. 2010, doi: 10.1002/PFI.20171.

[22] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An integrated theory of the mind," *Psychol. Rev.*, vol. 111, no. 4, pp. 1036–1060, Oct. 2004, doi: 10.1037/0033-295X.111.4.1036.

[23] T. P. Wright, "Factors Affecting the Cost of Airplanes," *J. Aeronaut. Sci.*, vol. 3, no. 4, pp. 122–128, Feb. 1936, doi: 10.2514/8.155.

[24] E. M. Dar-Ei, *HUMAN LEARNING: From Learning Curves to Learning Organizations*, vol. 29. Boston, MA: Springer US, 2000. doi: 10.1007/978-1-4757-3113-2.

[25] G. Mummolo, S. Digiesi, and G. Mossa, "Learning and Tiredness Phenomena in Manual Operation Performed in Lean Automated Manufacturing Systems: a Reference Model," 2004.

[26] M. Y. Jaber, Z. S. Givi, and W. P. Neumann, "Incorporating human fatigue and recovery into the learning–forgetting process," *Appl. Math. Model.*, vol. 37, no. 12–13, pp. 7287–7299, Jul. 2013, doi: 10.1016/j.apm.2013.02.028.

[27] N. Asadayoobi, M. Y. Jaber, and S. Taghipour, "A new learning curve with fatigue-dependent learning rate," *Appl. Math. Model.*, vol. 93, pp. 644–656, May 2021, doi: 10.1016/j.apm.2020.12.005.

[28] N. A. Stanton, "Special issue on human factors and ergonomics methods," *Hum. Factors Ergon. Manuf. Serv. Ind.*, vol. 32, no. 1, pp. 3–5, Jan. 2022, doi: 10.1002/hfm.20943.

[29] G. Paul and L. Briceno, "A Conceptual Framework of DHM Enablers for Ergonomics 4.0," in *Lecture Notes in Networks and Systems*, vol. 223 LNNS, 2022, pp. 403–406. doi: 10.1007/978-3-030-74614-8_50.

[30] N. Li, H. Kong, Y. Ma, G. Gong, and W. Huai, "Human performance modeling for manufacturing based on an improved KNN algorithm", doi: 10.1007/s00170-016-8418-6.

[31] T. S. Baines and J. M. Kay, "Human performance modelling as an aid in the process of manufacturing system design: A pilot study," *Int. J. Prod. Res.*, vol. 40, no. 10, pp. 2321–2334, Jul. 2002, doi: 10.1080/00207540210128198.

[32] R. A. Fisher, *The design of experiment*, 1st ed. Edinburgh, Scotland: Oliver and Boyd, 1935.

[33] M. Wantawin, W. Yu, S. Dachanuwattana, and K. Sepehrnoori, "An Iterative Response-Surface Methodology by Use of High-Degree-Polynomial Proxy Models for Integrated History Matching and Probabilistic Forecasting Applied to Shale-Gas Reservoirs," *SPE J.*, vol. 22, no. 06, pp. 2012–2031, Dec. 2017, doi: 10.2118/187938-PA.

[34] P. Ren *et al.*, "A Survey of Deep Active Learning," *ACM Computing Surveys*, vol. 54, no. 9. Association for Computing Machinery, Dec. 01, 2022. doi: 10.1145/3472291.

[35] S. Ren, Y. Deng, W. J. Padilla, and J. Malof, "Hyperparameter-free deep active learning for regression problems via query synthesis," Jan. 2022, [Online]. Available: http://arxiv.org/abs/2201.12632

[36] M. Haußmann, F. Hamprecht, and M. Kandemir, "Deep active learning with adaptive acquisition," in *IJCAI International Joint Conference on Artificial Intelligence*, 2019, vol. 2019-Augus, pp. 2470–2476. doi: 10.24963/ijcai.2019/343.

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

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

[39] Y. Geifman and R. El-Yaniv, "Deep active learning with a neural architecture search," 2019.

[40] P. Ren *et al.*, "A Comprehensive Survey of Neural Architecture Search," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–34, May 2022, doi: 10.1145/3447582.

[41] A. N. Donev and A. C. Atkinson, *Optimum Experimental Designs*. Clarendon Press, Oxford Statistical Science Series, 1992.

[42] W. Cai, Y. Zhang, and J. Zhou, "Maximizing expected model change for active learning in regression," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 51–60, 2013, doi: 10.1109/ICDM.2013.104.

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

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

[45] T. Pearce, M. Zaki, A. Brintrup, and A. Neely, "High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach," Feb. 2018, [Online]. Available: http://arxiv.org/abs/1802.07167

[46] 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.

[47] V. Mullachery, A. Khera, and A. Husain, "Bayesian Neural Networks," Jan. 2018, [Online]. Available: http://arxiv.org/abs/1801.07710

[48] T. Hothorn and B. Lausen, "Double-bagging: combining classifiers by bootstrap aggregation," *Pattern Recognit.*, vol. 36, no. 6, pp. 1303–1309, Jun. 2003, doi: 10.1016/S0031-3203(02)00169-3.

[49] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.02142

[50] C. Fiedler, C. W. Scherer, and S. Trimpe, "Practical and Rigorous Uncertainty Bounds for Gaussian Process Regression," 2021. [Online]. Available: www.aaai.org

[51] T. Heskes, "Practical confidence and prediction intervals".

[52] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Transactions on Neural Networks*, vol. 22, no. 9. pp. 1341–1356, Sep. 2011. doi: 10.1109/TNN.2011.2162110.

[53] M. Abdar *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Inf. Fusion*, vol. 76, pp. 243–297, Dec. 2021, doi: 10.1016/J.INFFUS.2021.05.008.

[54] J. Quiñonero-Candela, C. E. Rasmussen, F. Sinz, O. Bousquet, and B. Schölkopf, "Evaluating Predictive Uncertainty Challenge," 2006, pp. 1–27. doi: 10.1007/11736790_1.

7. Appendix A: Virtual manufacturing experiment design

The experiment involved human operators performing manufacturing assembly tasks and we recorded data relevant to the task duration. Other data was also recorded but not used in this case study.

7.1 Experiments

We conducted four experiments where subjects were required to perform simple assembly tasks and data was recorded. The assembly tasks involved material handling. This video [video here] demonstrates the tasks executed. The figure below shows schematics of the tasks but they are best explained via a video.

7.1.1 Task description

Subjects are required to place components in specific location and hit a submission button once they are complete. An audio and visual prompt informs them whether they have completed the task correctly. They perform the task for a number of repetitions and then move on to the next task.

We call these tasks: placement, stacking, sorting, and joining. We have a subjective notion of them increasing in complexity. The task sequence was not randomized, but could easily be done. We chose a fixed sequence of increasing complexity because it was the first time many of the subject have used virtual reality and the more complex tasks can be frustrating for beginners.

The first two tasks (placement and stacking) are repeatable. These tasks require the same components to be placed at the same points every repetition. In placement two bars are placed in a required pose, in stacking three cylinders are stacked on top of each other.

The next two tasks (sorting and joining) are random tasks which we regard as more complex. Every repetition a new random formula is given in the form of a simple schematic and the subject must select the appropriate components from bins. The formula is presented using primary shapes that correspond to assembly components cylinder, square-bar, triangular, and cross-shaped (X). In the joining task 5 components are fixed (welded/tacked) together before submitting.

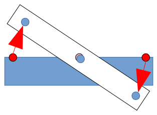
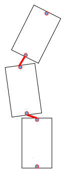
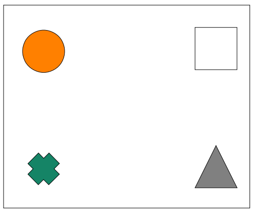
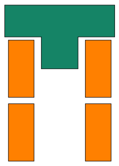
Placement	Stacking	Sorting	Joining
			
Repeatable	Repeatable	Random	Random

Figure 13: Task matrix

Errors occurred when a subject placed/stacked the incorrect number of components, or placed the incorrect component for a recipe. Errors did not count toward the repetitions in a trial, so if 2 errors occurred in a 10 repetition trial, the two error reps are repeated, totaling 12.

7.1.2 Experimental procedure

In trials for placement, stacking, and sorting (1-3) subjects completed 10 repetitions, assembling 10 components. In task 4, subjects only assembled 5 components due to the duration and challenge of the final task.

Subjects completed up to 7 trials over the span of five days. Trials were spaced randomly, with a minimum 5 hours space between trials of the same subject. All subjects were between the age of 20-30[2]. There were 11 males and 1 female who took part in the study[3]. Not all subjects completed 6 sequential trails. The table below shows the frequency of trails completed. All recipients had little previous exposure to using VR. No compensation was given for this experiment.

7.1.3 Ergonomic calibration

We noticed in pre-testing that the table height should be adjusted for each individual. This was particularly evident for taller individuals who sometimes had to bend to reach the work surface causing irritation due to extra fatigue.

We introduced a calibration phase in which the table height was adjusted based on the individuals limb length. Individuals assumed a series of poses and we calculated the limb length, using this to adjust the workstation. This blocked ergonomic factors between subject.

[2] There tends to be a substantial mature population in manufacturing, not represented in this study.

[3] This is not an unusual distribution of sexes in manufacturing environments.

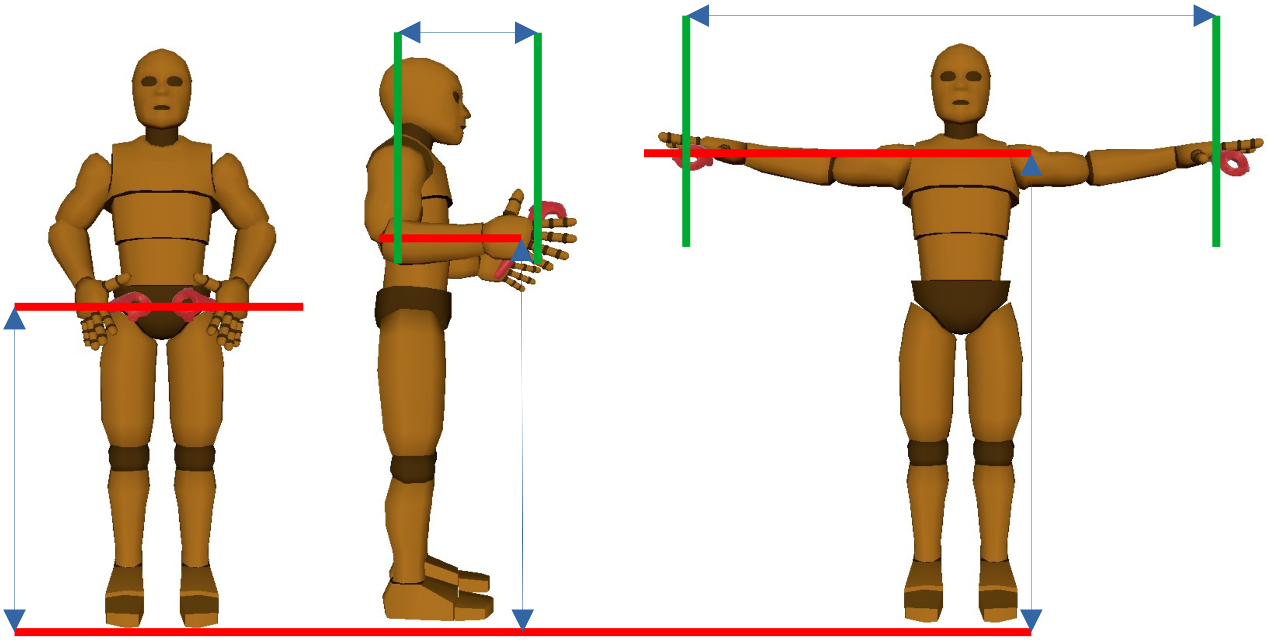


Figure 14: Limb length measurement of individuals, used to adjust the workspace.

This hints that VR can be used to design ergonomic workstations without the need for actual hardware. This is outside the scope of our work and so was not explored much further.

7.1.4 Spaced learning

In this trail we assumed spaced-learning, where subjects had a period between each trial, usually a day. We where faced with a single-case where one subject was available for only one day and decided run 4 trials on one day. The results were inconsistent with the others so we removed this subject from the trial.

Table 1: Frequency of trials completed.

Trials	Number of subjects that completed
2	1
3	5
4	4
5	0
6	1
7	1

7.2 Experimental implementation

The VR environment was developed in Unity3D using the SteamVR plugin. Custom C# code was developed in house. The experiments were conducted using the HTC Vive Cosmos head mounted display and controllers.

8. Appendix B: supporting plots

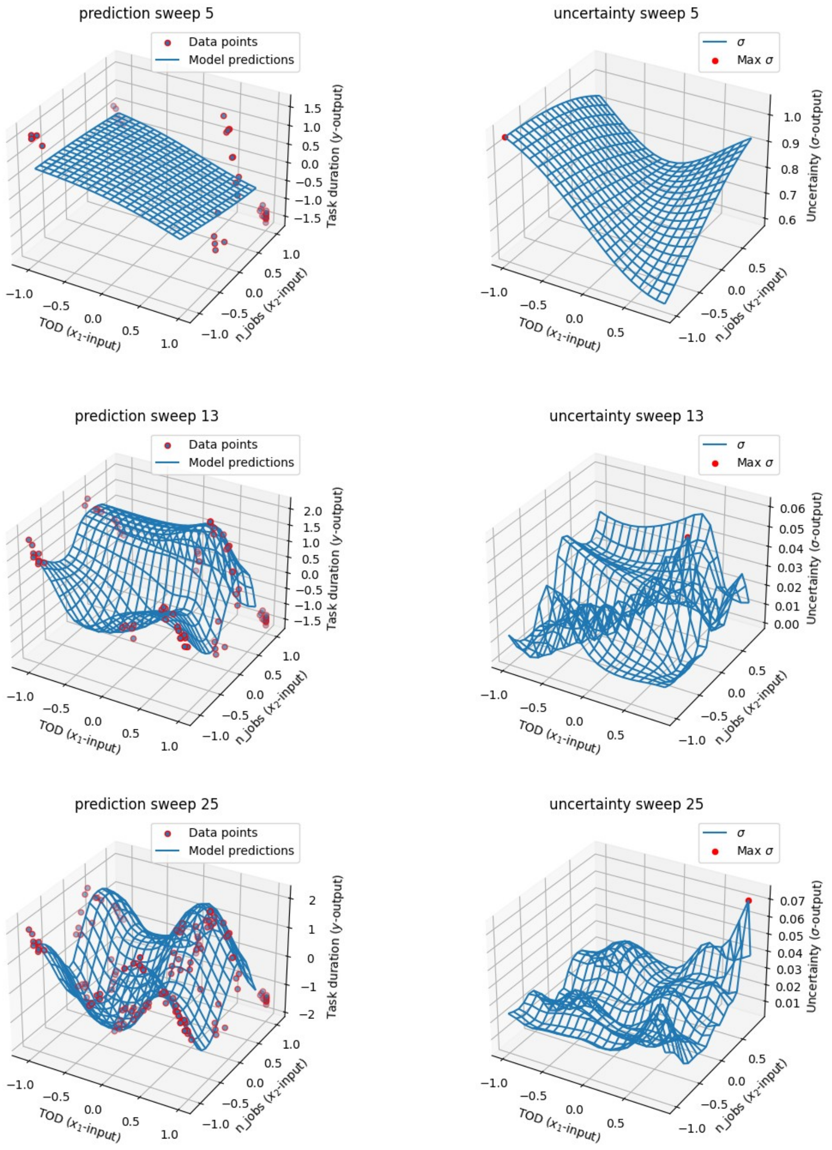


Figure 12 : A toy problem with 2 input features $(y = sin(x_1) + sin(x_2) + N(0, q^2))$. On the left the mean predictions [gif here], on the right the uncertainty [gif here]. Note how the plot axis limit the selection of the maximum σ.