# Wilson Score Kernel Density Estimation for Bernoulli Trials

Lars Carøe Sørensen, Simon Mathiesen, Dirk Kraft and Henrik Gordon Petersen
*SDU Robotics, University of Southern Denmark, Campusvej, Odense, Denmark*

Keywords: Iterative Learning, Statistical Function Estimators, Binomial Trials.

Abstract: We propose a new function estimator, called Wilson Score Kernel Density Estimation, that allows to estimate a mean probability and the surrounding confidence interval for parameterized processes with binomially distributed outcomes. Our estimator combines the advantages of kernel smoothing, from Kernel Density Estimation, and robustness to low number of samples, from Wilson Score. This allows for more robust and data efficient estimates compared to the individual use of these two estimators. While our estimator is generally applicable for processes with binomially distributed outcomes, we will present it in the context of iterative optimization. Here we first show the advantage of our estimator on a mathematically well defined problem, and then apply our estimator to an industrial automation process.

## 1 INTRODUCTION

Optimization of stochastic processes is a common task in industrial robotics. This includes a wide range of processes like peg-in-hole and screwing operations, but also design of feeding solutions as we briefly touch later in this paper as test case. Such processes are likely to be influenced by uncertainties, which need to be handled to achieve a successful execution. However, many experiments are normally required to obtain reliable estimates of stochastic functions, and each evaluation is often seen as being expensive (e.g. costly or time-consuming). Hence, making a sampling of the entire parameter space in such cases is not feasible, since this Naive sampling is sample-inefficient. The problem becomes even more severe when the stochastic process is defined in multiple dimensions with wide parameter ranges, which results in a large parameter space, and when an evaluation of the function is limited to a binary outcome, which only reveal whether the experiment succeeded or failed.

One way to approach this problem is by taking the uncertainty of the function estimate into account during the optimization of a stochastic function and thereby obtaining a proper estimate of the unknown underlying function. This can be done by both calculating statistical estimates on the true mean and the surrounding confidence interval (e.g., using Normal Approximation (Ross, 2009)). In addition, Kernel Density Estimation (Härdle et al., 2004) can be used to account for the likely local smoothness in the parameters of these stochastic problems. As a result, this makes the selection more effective, since an experiment also expresses information about the neighboring region.

In our previous work (Sørensen et al., 2016), we have shown that by actively using both the mean estimate and the associated uncertainty in an iterative learning setting, the number of function evaluations required can be drastically reduced. The purpose of the iterative learning is to make an effective sampling of the parameter space. However, each decision on which part of the parameter space to explore next is in the beginning being hindered by the sparse amount of data. Decisions based on little data will often become unreliable in such situations. The common function estimators (e.g. Normal Approximation) often require a significant amount of experiments to obtain a usable estimate on the true function, which makes them inapplicable due to sample-ineffectiveness.

As a discrete function estimation, Wilson Score (Agresti and Coull, 1998) has the property of making a reasonable estimate when having few samples compared to Normal Approximation. Moreover, regression by Kernel Density Estimation (Härdle et al., 2004) is a continuous function estimator that generalizes the outcomes of the function evaluations to the neighboring region by kernel smoothing. The novelty of this paper is the derivation of a new statistical function estimator, which both has the smoothing property from Kernel Density Estimation and the few



Sørensen, L., Mathiesen, S., Kraft, D. and Petersen, H.
Wilson Score Kernel Density Estimation for Bernoulli Trials.
DOI: 10.5220/0009816503050313
In *Proceedings of the 17th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2020)*, pages 305-313
ISBN: 978-989-758-442-8

samples correction from Wilson Score while also being continuous.

The paper is structured as follows: Section 2 starts by defining the overall goal of the optimization, and then describing our iterative learning approach. Section 3 briefly recaps some methods, namely Normal Approximation and Kernel Density Estimation, and discusses in further details why these function estimators are unusable in an iterative setting when having a limited number of samples. Section 4 includes our main contribution which is the derivation of the new function estimator "Wilson Score Kernel Density Estimation" which combines the properties of Wilson Score and Kernel Density Estimation regression. We show the advantages of our new function estimator by applying it on a simple mathematical problem in Section 5, but we also use our function estimator in an iterative learning setting for optimizing a real industrial case in Section 6. Finally, we conclude the paper in Section 7 and then propose future work in Section 8.

# 2 APPROACH, ASSUMPTIONS AND CURRENT WORK

The overall goal is to gain the best execution of a given industrial process based on only binary outcome (success or failure). This is achieved by optimizing the process parameters and thereby finding the highest probability of success for the process:

$$x_{opt} = \underset{x \in \mathcal{X}}{\operatorname{argmax}}(p(x)), \tag{1}$$

where $x$ is an arbitrary parameter set in a metric parameter space, $\mathcal{X} \in \mathbb{R}$, and $x_{opt}$ denotes the parameter set that gives the highest probability of success, $p(x)$. We assume that the function $p(x)$ is continuous.

What we have to our disposal for performing the optimization is a manual limitation of the parameter space to ensure that $\mathcal{X}$ is bounded, and the possibility to perform experiments, i.e. executions of the process, with a chosen parameter set. An experiment with parameter set $x$ can be described as a Bernoulli trial with (unknown) probability $p(x)$ which generates an outcome defined as $y \in \{0,1\} = \{f,s\}$ corresponding to failure and success, respectively. In the iterative learning described below, we perform a sequence of experiments with different parameter sets, where the $i$-th experiment is defined as $\{x_i, y_i\}$. We assume that the underlying probability of success for an experiment with parameter set $x$ is independent of when the experiment is carried out (i.e. independent of the placement $i$ in the sequence).

## 2.1 The Iterative Learning Approach

For iterative learning in our setting, an efficient approach is required to reduce the number of experiments needed for the optimization of the process parameters. In each iteration a well-considered choice must be made on which parameter set to investigate next. In the literature, the choice is realized through the use of statistical calculations which are based on all the experiments performed in previous iterations. Hence, each iteration of the learning approach uses the principles of Bayesian Optimization (Brochu et al., 2010) which generally is constructed as:

1. **Selection:** Select the parameter set, $x_i \in \mathcal{X}$, for the next experiment based on the statistical measures calculated from all the previous experiments, $\mathcal{D}_{i-1}$.
2. **Experiment:** Perform an experiment with the parameter set $x_i$ and obtain the outcome $y_i \in \{0,1\}$.
3. **Save:** Save the experiment $\mathcal{D}_i = \{\mathcal{D}_{i-1}, \{x_i, y_i\}\}$.

For Bayesian Optimization, the selection in each iteration is conducted by maximizing an acquisition function by $x_i = \operatorname{argmax}(acq(x))$. There exist a variety of acquisition functions (see e.g. (Brochu et al., 2010; Sørensen et al., 2016)). Most of the acquisition functions require estimates of the mean, $p(x)$, and the variance $\sigma^2(x)$ at any $x$, which must be reliable for the iterative learning to efficiently select proper parameter sets. An often used acquisition function is the Upper Confidence Bound (UCB) (Tesch et al., 2013), which is defined as $ucb(x) = p(x) + \kappa\sqrt{\sigma^2(x)}$, where $\kappa$ defines a trade-off between exploration and exploitation. In the next section, it is explained how this trade-off can be automatically adjusted by using the confidence interval.

# 3 EXISTING STATISTICAL ESTIMATORS

In this section, we discuss different existing function estimators for estimating the true mean, $p(x)$, and variance, $\sigma^2(x)$, for any $x$ based on a set of prior experiments $\mathcal{D}$. All the function estimators will for convenience be described in terms of the confidence interval, and we therefore introduce the common definition of the true confidence interval (see e.q. (Ross, 2009)) as:

$$\left[ p(x) \pm z\sqrt{\sigma^2(x)} \right], \tag{2}$$

where $z$ is defined as the $(1 - \frac{\alpha}{2})$ quantile for a two-sided interval.

We start by defining the simple Normal Approximation (NA), which acts as a basis for Kernel Density Estimation (KDE). After defining KDE, we then explain the problem which arises when having a sparse sampling of the parameter space, $\mathcal{X}$, and how Wilsom Score (WS) can correct for this problem.

## 3.1 Normal Approximation

Assume that the parameter space, $\mathcal{X}$, is tesselated into a finite set of representative points. Consider an arbitrary $x_i$, and assume that we have performed $n_i$ experiments with that parameter set. The straightforward function estimator to use is NA (Ross, 2009). For NA the true probability, $p(x_i)$, for a Bernoulli distribution can be estimated by:

$$\hat{p}_{na}(x_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} y_j \,, \tag{3}$$

where $y_j$ is the outcome of the $j$-th experiment in the $i$-th point. Moreover, it can be proven that the NA estimate converges towards true mean such that $\hat{p}_{na} \to p$ when $n \to \infty$ (Ross, 2009).

Likewise the variance is defined as:

$$\sigma^2_{na}(x_i) = \frac{1}{n_i} \hat{p}_{na}(x_i)(1 - \hat{p}_{na}(x_i)) \,. \tag{4}$$

The confidence interval for NA can be obtained by substituting (3) and (4) into (2).

The problem with NA is that the mean estimate is very dependent on the individual outcomes for a low $n$. This means that a large number of experiments are typically needed to cover the parameter space and to obtain reliable statistics. This makes the NA function estimator ill suited in combination with an iterative learning method due to effectiveness, since choices made in the beginning of the iterative process will be based on unreliable (and potentially wrong) estimates. Moreover, the probability estimates for neighboring parameter points will in particular for a relative dense tessellation be correlated as $p(x)$ is continuous. An efficient function estimator needs to exploit this, which is not the case for NA which is a discrete estimator.

Several approaches utilized smoothing principles to let the neighboring experiments influence the probability estimate such as Gaussian Processes (Rasmussen and Williams, 2006) or K-nearest neighbors regression (Härdle et al., 2004). Our previous work (Laursen et al., 2018) showed how Gaussian Processes applied to a binomial setting[1] lacks the ability to properly explore the parameters space. The paper also shows that including the number of samples in the calculation of the confidence interval instead of only variance improves the performance the acquisition function when used for iteratively selecting the next parameter set to explore (see also Section 2.1). Despite the improved performance, this variations only mimics the true calculation of the confidence interval in (2) without being theoretically defined. Furthermore, note that the approach used later in this paper to develop our new function estimator named WSKDE cannot directly be transferred to Gaussian Processes Classification due to their derivation. We will in this paper restrict ourselves to the generic non-parametric Kernel Density Estimation (KDE) regression, which previously has been shown to be very suitable for process optimization (Sørensen et al., 2016).

[1]Formally known as Gaussian Processes Classification.

## 3.2 Kernel Density Estimation

The first step in Kernel Density Estimation (KDE) is to define an estimate of the density of experiments, $f(x)$, at an arbitrary parameter set, $x$. This estimate is in (Härdle et al., 2004) defined as:

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^{n} K_{h,x_i}(x) \,, \tag{5}$$

where $n$ is the total number of experiments in the entire parameter space $\mathcal{X}$, and where $x_i$ is the parameter set applied in the $i$-th experiment. Moreover, $K_{h,x_i}(x)$ is the smoothing kernel located in $x_i$ with a bandwidth of $h$.

The estimate of the success probability $p(x)$ by KDE is defined in (Härdle et al., 2004) as:

$$\hat{m}_h(x) = \frac{\hat{f}_{h,Y}(x)}{\hat{f}_h(x)} = \frac{n^{-1} \sum_{i=1}^{n} K_{h,x_i}(x)\, y_i}{n^{-1} \sum_{j=1}^{n} K_{h,x_j}(x)} \,, \tag{6}$$

where $\hat{f}_{h,Y}(x)$ is the estimated density weighted by the outcome $y$. Hence, for experiments with a binomial outcome (success or failure), $\hat{f}_{h,Y}(x)$ will simply be the estimate density of successful experiments. Accordingly to (Härdle et al., 2004) the KDE regression estimate converges towards true mean such that $\hat{m}_h(x) \to m(x)$ when $h \to 0$ and $nh \to \infty$.

We can also rewrite (6) as:

$$\hat{m}_h(x) = \frac{1}{n} \sum_{i=1}^{n} W_{h,i}(x)\, y_i \,, \tag{7}$$

where $W_{h,i}(x)$ is referred to as the weighting:

$$W_{h,i}(x) = \frac{K_{h,x_i}(x)}{n^{-1} \sum_{j=1}^{n} K_{h,x_j}(x)} \,. \tag{8}$$

Hence, the confidence interval for KDE regression can be estimated in (Härdle et al., 2004) as:

$$\left[ \hat{m}_h(x) \pm z \sqrt{\frac{||K||_2^2 \hat{\sigma}^2(x)}{nh\hat{f}_h(x)}} \right] , \tag{9}$$

where $||K||_2^2$ is the squared $L_2$ norm of an identity kernel ($\int\{K(u)\}^2du$), and thereby a scalar value only dependent on the chosen kernel type. Additional, $\hat{\sigma}_h^2(x)$ is the estimated variance which is given in (Härdle et al., 2004) as:

$$\hat{\sigma}_h^2(x) = \frac{1}{n}\sum_{i=1}^{n} W_{h,i}(x)(y_i - \hat{m}_h(x))^2 . \quad (10)$$

Please note that the term under the square root in (9) differs from the original definition of the confidence interval in (2), since the variance is scaled by $||K||_2^2/(nh\hat{f}_h(x))$.

It is important to add, that the KDE regression in (6) and the confidence interval in (9) estimates both suffers from a bias and variance error. The bias error arises from the kernel smoothing and can be eliminated by letting $h \to 0$, whereas the variance error is eliminated by letting $nh \to \infty$. To make the KDE regression confidence interval in (8) calculable, it is derived under the assumption that $h$ has been chosen small enough so that the bias can be neglected. In Appendix "The Effect of the Bias and Variance Error in relation to KDE and WSKDE" this assumption and the effect of the bias is discussed.

### 3.3 Wilson Score

The aim of our approach is to reduce the number of samples needed by the iterative learning approach by focusing on the promising regions of the parameter space. However, for this approach to obtain good performance, the accuracy of mean estimate and the confidence interval are important. We considered to begin with the NA function estimator. It is well known that NA needs (as a rule of thumb) at least five experiments leading to each of the two outcomes in order to achieve a robust confidence interval (Brown et al., 2001). Therefore, for parameter points where there are very few experiments, NA typically provides unrealistic confidence intervals. Unfortunately, the KDE confidence interval estimate in (9) suffer from the same problem if there are an insufficient amount of samples in the neighbor region.

To deal with the disadvantages of NA, the Wilson Score (WS) can be used for estimating on the confidence interval. The estimate of the mean is for WS defined in (Agresti and Coull, 1998) as:

$$\hat{p}_{ws}(x_i) = \alpha_1 \hat{p}_{na}(x_i) + \frac{1}{2n_i}z^2 , \quad (11)$$

where $\hat{p}_{na}(x_i)$ is the mean estimated by NA from (3), $n_i$ is number of experiments performed in the $i$-th parameter point $x_i$, and $\alpha_1 = 1/(1+n_i^{-1}z^2)$.

Moreover, the estimated variance is by WS defined in (Agresti and Coull, 1998) as:

$$\sigma_{ws}^2(x_i) = \alpha_1 z\sqrt{\frac{1}{n_i}\hat{p}_{na}(x_i)(1-\hat{p}_{na}(x_i)) + \alpha_2} , \quad (12)$$

where $\alpha_2 = z^2/(4n_i^2)$.

As for NA, the confidence interval for WS can be obtained by substituting (11) and (12) into (2). Studying the WS confidence interval shows that when $n_i \to 0$ then the interval becomes $[0;1]$, or equivalent $[0.5 \pm 0.5]$, and when $n_i \to \infty$ then the WS interval becomes equal to the NA interval including that $\hat{p}_{ws}(x_i) \to \hat{p}_{na}(x_i)$ which converges towards the true mean. By these two properties, WS eliminates the disadvantage of NA when having a sparse sampling. However, as for NA, WS is also a discrete function estimator opposed to KDE which takes the neighboring samples into account. In the next section, we present a novel Wilson Score inspired estimate of the confidence interval for KDE that is more robust than the classical KDE estimate.

## 4 WILSON SCORE KERNEL DENSITY ESTIMATION

Even though Wilson Score (WS) gives a proper function estimate when having a low number of samples, it is a discrete function estimator as Normal Approximation (NA), and we therefore want to combine WS with the smoothing property from regression by Kernel Density Estimation (KDE). To derive our new statistical KDE confidence interval estimator, we start with expanding the KDE variance from (10) as:

$$\hat{\sigma}_h^2(x) = \frac{1}{n}\left(\beta_1 + \beta_2 - \beta_3\right) , \quad (13)$$

which by considering a Bernoulli distribution where $y \in \{0,1\}$ corresponding to failure and successful outcomes can be simplified to:

$$\beta_1 = \sum_{i=1}^{n} W_{h,i}(x)y_i^2 \overset{Ber}{=} n\hat{m}_h(x) , \quad (14)$$

$$\beta_2 = \sum_{i=1}^{n} W_{h,i}(x)\hat{m}_h(x)^2 \overset{Ber}{=} n\hat{m}_h(x)^2 , \quad (15)$$

$$\beta_3 = 2\sum_{i=1}^{n} W_{h,i}(x)\hat{m}_h(x)y_i \overset{Ber}{=} 2n\hat{m}_h(x)^2 , \quad (16)$$

which can be substituted into (13) and simplified to:

$$\hat{\sigma}_h^2(x) \overset{Ber}{=} \hat{m}_h(x)(1-\hat{m}_h(x)) . \quad (17)$$

This result can be inserted into (9) to obtain the KDE confidence interval for Bernoulli trials as:

$$\left[ \hat{m}_h(x) \pm z\sqrt{\frac{||K||_2^2}{nh\hat{f}_h(x)}\hat{m}_h(x)(1-\hat{m}_h(x))} \right]. \quad (18)$$

Comparing this with the NA estimates from (3) and (4):

$$\left[\hat{p}_{na}(x_i) \pm z\sqrt{\frac{1}{n_i}\hat{p}_{na}(x_i)(1-\hat{p}_{na}(x_i))}\right], \quad (19)$$

allow us to identify the mean and in particular the KDE sample size at $x$ as:

$$\hat{p}_{na}(x_i) = \hat{m}_h(x) \quad \text{and} \quad n(x) = \frac{nh}{||K||_2^2}\hat{f}_h(x), \quad (20)$$

where $nh/||K||_2^2$ scales the estimated sample density $\hat{f}_h(x)$ based on the total number of samples, $n$, and the chosen bandwidth of the kernel, $h$.

Hence, the two expressions from (20) can be substituted into (11) and (12) to obtain the estimated mean and variance for our new Wilson Score Kernel Density Estimation (WSKDE) function estimator. The estimated mean is then:

$$\hat{p}_{wskde}(x) = \gamma_1\hat{m}_h(x) + \frac{1}{2n(x)}z^2, \quad (21)$$

where $\gamma_1 = 1/(1+n(x)^{-1}z^2)$, and the estimated variance is:

$$\hat{\sigma}_{wskde}(x) = \gamma_1 z\sqrt{\frac{1}{n(x)}\hat{m}_h(x)(1-\hat{m}_h(x)) + \gamma_2}, \quad (22)$$

where $\gamma_2 = 1/(4n(x)^2)z^2$.

The result of the WSKDE derivation in (20) implies that our WSKDE estimate also converges towards the true mean when $n \to \infty$ under the conditions $h \to 0$ and $nh \to \infty$. Moreover, the WSKDE confidence interval has the same properties as WS by approaching $[0.5 \pm 0.5]$ when $n \to 0$. Note that the neglection of the bias error for KDE does not effect the derivation of WSKDE.

In Appendix "Generalization to Multiple Dimensions" it is briefly explained how the KDE and WSKDE function estimators can be generalized to multiple dimensions.

## 5 EXPERIMENTAL VALIDATION

An experimental validation is conducted to show the performance difference between the KDE and WSKDE function estimators. The performance of KDE or WSKDE is in this experiment defined as how often their confidence interval includes the underlying function, $p(x)$. We will not compare WSKDE against NA or WS, since these are discrete estimators. In relation to the iterative learning approach (see Section 2.1), it is of interest to iteratively conduct experiments so the convergence in performance can be examined. For our experiment, we use the following underlying test function:

$$p_{test}(x) = 0.5\,(1+\sin(x)) \quad (23)$$

where $x \in [0; 2\pi]$.

In each test a total of 100 iterations are conducted. For each iteration an experiment is carried out by picking a random position $x_i$ from the uniform distribution on the interval of $x$ (i.e. $[0; 2\pi]$) and picking a random number $r$ uniformly distributed in the interval $[0; 1]$. We then define the outcome as $y_i \equiv s$ if $r \leq f(x_i)$ and otherwise $y_i = f$, where $s$ and $f$ are success and failure respectively. The performance of both KDE and WSKDE is calculated by tessellating the $x$-axis into $n_{tes} = 101$ discrete points, and testing wherever their respective confidence interval encapsulates the true function $f(x)$. A confidence of 95% is used for the intervals which corresponds to $z \approx 1.95$.

Hence, the average performance is in the $i$-th iteration calculated as:

$$p_{avg,i} = \frac{1}{n_{tes}}\sum_{j=1}^{n_{tes}}\delta(x_j), \quad (24)$$

where $\delta(x_j)$ is 1 if both $lcb(x_j) < f(x_j)$ and $ucb(x_j) > f(x_j)$ and otherwise 0. Moreover, $x_j$ is the $j$-th tessellation point and $lcb(x_j)$ and $ucb(x_j)$ is the lower and upper bound of the confidence interval of either KDE or WSKDE (see (9), (21), and (22)).

To illustrate the difference between KDE and WSKDE, Figure 1 shows three plots of the underlying test function $p_{test}(x)$ and the estimated mean and confidence interval of both the KDE and WSKDE at different iterations. Also the performance measure in each of the tessellation points is shown. The figure clearly shows how KDE (blue curve) struggles to properly estimate the true function (green curve). The plots also show how the confidence interval of WSKDE (red curve) takes advantage of the few samples correction property of WS by adjusting the estimate from $[0.5 \pm 0.5]$ towards the true function. Hence, WSKDE includes the true function significantly better than KDE and it is therefore producing more reliable results when having a sparse sampling of the parameter space.

To obtain statistics on the results, the procedure explained above is repeated 50 times and the average of $p_{avg,i}$ is calculated. The results are presented in Figure 2. It clearly shows that the KDE confidence interval rarely includes the true function in

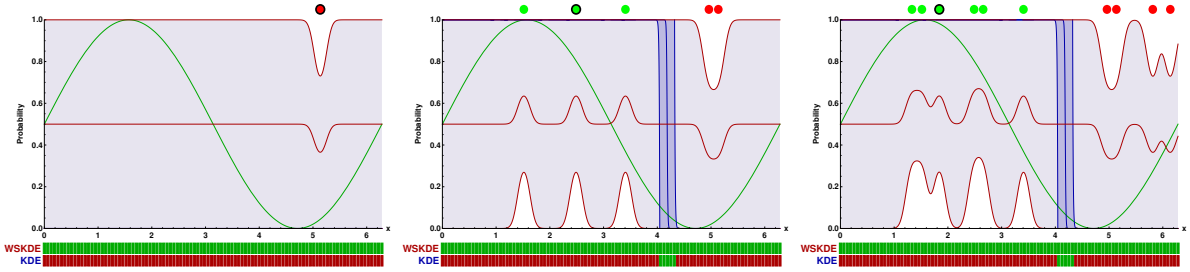


Figure 1: The figure shows a plot of underlying test function $p_{test}(x)$ (green curve), the estimated mean and confidence interval of the KDE (blue curve) and of WSKDE (red curve) for iteration 1, 5, and 10. The two bars below each plot show the performance of KDE (upper bar) and WSKDE (lower bar) for each of the $j$ tessellation points where green and red respectively means that confidence interval includes the underlying test function or not. The green and red disks above each plot represents successful and failed samples. Note the estimated mean and confidence interval KDE in iteration 1 is zero in the entire parameter space.

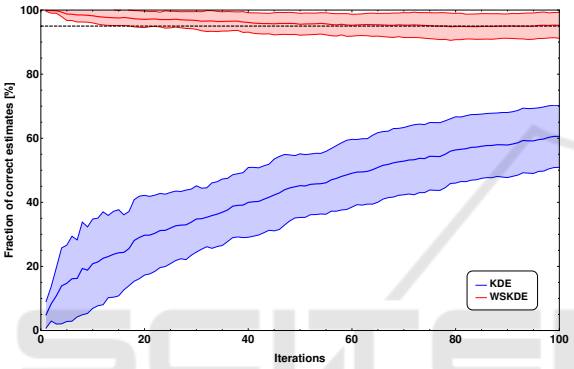


Figure 2: The figure shows how well the confidence interval of the KDE (blue) and WSKDE (red) functions estimators includes the underlying function. The procedure has been repeated 50 times. Hence, the two solid lines show the percentage of how often the confidence interval on average includes the underlying function, and the hatched area around the lines represent one standard deviation. The dashed line shows the 95% performance.

the beginning. KDE gradually improves its performance during the iterations, however, it does on average only reach 60 % in iteration 100. Inspecting the WSKDE result shows that it on average includes 95 % of the true function, which is also expected since the function estimators use a 95 % confidence interval. Note, the WSKDE confidence interval does include the whole underlying function in the beginning (performance of 100 %), which was also expected since no or only few neighbor samples exist.

# 6 OPTIMIZATION OF AN INDUSTRIAL ASSEMBLY CASE

In addition to the experimental validation on the simple mathematical function in the previous section, we will in this section apply our iterative learning approach to a real industrial case. For this test case, we first carry out the iterative learning process using dynamic simulations, and then test the best solution in real-world. We have in previous work (Mathiesen et al., 2018) shown that our dynamic simulations align very well with real-world experiments and produce reliable results. We limit the experiments to the use of our Wilson Score Kernel Density Estimation function estimator, since the previous section showed the problems with the pure Kernel Density Estimation function estimator. In this section, we first explain the case, the scenario and which parameters we want to optimize. We then briefly explain how we select the sample in each iteration and finally present the results.

## 6.1 Part Feeding with Vibratory Bowl Feeders

Vibratory Bowl Feeders (VBFs) is still today an important part in industrial assembly. The purpose of the VBFs are to orient parts (which typically come in bulk) into a desired orientation, so these parts easily can be handled by subsequent automation system. VBFs can be used to feed a multitude of parts where a typical use case is feeding screws. A VBF works by vibrating parts forward from the bottom of the bowl along a track on which orienting mechanisms called traps are located. For our test case we optimize a rejection trap for a brass cap (see Figure 3). The purpose of a rejection trap is to reject wrongly oriented caps for recirculation (position B) and let correctly oriented caps pass (position A). The figure also shows the four parameters which control the performance of the trap and are described in Table 1. These parameters are today tuned manually by human experts, typically in a trial-and-error process, even though some guidelines do exist (Boothroyd, 2005).

Table 1: The parameters for the chosen rejection trap along with their bounds, discretization. The standard deviation of the kernel is bandwidth of the kernel, $h$, which in multiple becomes a bandwidth matrix, $H$. All values are in millimeters.

| Parameters | | Range | | | Kernel |
|---|---|---|---|---|---|
| Name | Description | Min | Max | Disc. | Std. |
| w | Width of track | 0.0 | 12.0 | 1.0 | 1.00 |
| d | Distance to cut-out | 0.0 | 8.0 | 1.0 | 1.00 |
| r | Radius of cut-out | 3.0 | 15.0 | 0.5 | 0.25 |
| p | Width of protrusion | 0.0 | 11.0 | 1.0 | 1.00 |

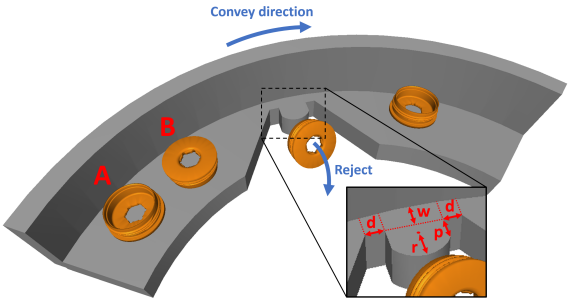


Figure 3: The object and rejection trap used in our test case. The object is a brass cap which can be oriented in one of two stable poses (A or B). The purpose of the traps is to reject caps in orientation B and let caps in orientation A pass. This trap has four parameters which are optimized to gain the best performance. Rejected parts fall to the bottom of the bowl and are thereby recirculated.

## 6.2 Experimental Setup and Choices

We use the iterative learning approach described in Section 2.1 for optimizing the chosen parameters in our test case. For the iterative selection of the next parameter set, we use a refined version of the Upper Confidence Bound (UCB) as acquisition function. Instead of letting $\kappa$ define the trade-off between exploration and exploitation, we let the upper bound of the confidence interval automatically control this adjustment so $acq(x) = p(x) + z\sqrt{\sigma^2(x)}$. We name this acquisition function the Upper Confidence Interval Bound (UCIB). As function estimator we use our WSKDE (see (20) and (22)) and we utilize a 95% confidence interval which result in $z \approx 1.96$.

We choose to discretize the parameter space, $\mathcal{X}$, since the selection of the next parameter set then becomes as simple as iterating though all sample points and picking the one with the highest upper confidence bound (in opposition to finding the maxima in a large continuous parameter space often consisting of multiple maximums). It also allows for pre-calculating the kernel mask instead of calculating all the kernel contributions individually. Moreover, we choose a Gaussian kernel with a diagonal kernel matrix consisting of the standard deviation values shown in Table 1, which are set to the discretization of parameters to allow for smoothing. A total of 1500 iterations are conducted.

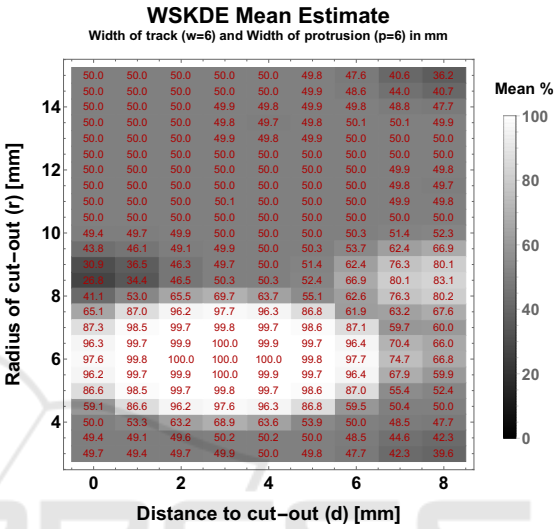


Figure 4: A cross-sectional view of the mean estimated with WSKDE after 1500 iterations where the parameters $w$ and $p$ have been fixed. The parameter set with the highest mean estimate is located at $d = 3.0$ and $r = 6.0$ (with $w = 6.0$ and $p = 6.0$).

## 6.3 Result and Discussion of Test Case

As an example, Figure 4 shows a 2D plot of the parameter $d$ and $r$ where the parameters $w$ and $p$ both have been fixed to $6\,[mm]$. Due to space constraints it is not possible to show 2D plots of the entire parameter space, since we consider four parameters with a wide range. The result shows that all parameters have an influence on the trap performance.

For the first 91 iterations, the iterative learning approach explores the parameter space and obtains both successes and failures. Hereafter, the iterative learning finds one parameter set which is exploited for the majority of the remaining 1409 iterations without any failures. The reason why the iterative learning keeps selecting this one parameter set is because the UCIB is slightly higher than for other parameter sets. Moreover, the UCIB of WSKDE does not get lower if only successes are obtained, and this parameter set will therefore be chosen continuous. Only seven time a different parameter set is selected, but this due to machine precision and each time iterative learning im-

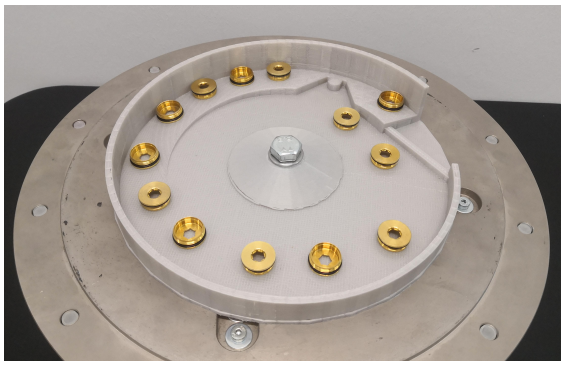

Figure 5: The 3D-printed bowl with the optimized parameters mounted on a VBF drive. The optimized rejection trap is located in top of the bowl just before the outlet. See Figure 3 for details on the trap parameters.

mediately returns because a failure is obtained.

After the 1500 iterations, the parameter set with the highest estimated mean is selected. This parameter set has the values of $w = 6.0$, $d = 3.0$, $r = 6.0$, and $p = 6.0$ (all values in millimeters), and has a mean value of 99.97 % and a confidence interval of $[99.95; 100.00]$ % when calculated by the WSKDE function estimator. The mean is 100 % when calculated by Normal Approximation (see (3)) since only successes are obtained in this parameter set and only these are considered by this estimator. The reason why WSKDE has a slightly lower mean estimate is because of the few samples correction from Wilson Score. The many successes in this parameter set lead to that the few failures close by do not have a significant impact and the kernel smoothing does therefore not the cause of this lower mean estimate.

For our real-world test, we 3D-printed a bowl with the parameters found above which is shown in Figure 5. The bowl has been tested 200 times for each of the two stable poses of the brass cap (see Figure 3). The result shows that all the brass caps starting in stable pose B were rejected and those starting in stable pose A all passed the trap. This yields a success rate of 100%, and with a total of 400 experiments, the resulting design is therefore found to be robust.

# 7 CONCLUSION

This paper presents a new function estimator denoted Wilson Score Kernel Density Estimation (WSKDE) for experiments with binary outcomes. The estimator has been theoretically derived and has the few samples correction from Wilson Score and the smoothing property from Kernel Density Estimation regression. The estimator is especially suited for iterative learning methods since their sampling strategy often requires efficient and trustworthy estimators in the beginning of the learning process where decisions are based on sparse information. The benefit of this estimator has been visualized on a mathematically defined problem and shown to work on a real industrial use case.

# 8 FUTURE WORK

Future work could both include topics related to the WSKDE function estimator and the iterative learning approach. We will below present some of the most relevant topics for these two subjects.

Categorizing the outcome of an experiment as being either success or failure is often the most convenient, and sometimes the only possibility, whether experiments are conducted in simulation or real-world. This makes the presented approach generally applicable. However, further information about the experiment is for some applications available. Therefore, it would be beneficial to extent the current WSKDE function estimator for utilizing outcomes in more categories or even as a continuous value from 0 to 1 representing how successful an experiment was.

Other topics worth investigating related the WSKDE function estimator could be the pros and cons for using a discrete and continuous parameter space, but also how the kernel sizes adaptively can be adjusted. The latter could potentially lower the effect from smoothing as more samples are taken and thereby improve the function estimates.

For the iterative learning, a future topic could be to implement and compare other acquisition functions to gain other behaviors . This could include studying the influence of selecting $z$-score differently than a 95 %-percent confidence interval. Moreover, the iterative learning is currently terminated after an user-defined number of iterations. I could be beneficial to expose other criteria for termination as when the lower confidence bound of one parameters set is above a acceptable threshold. This would make the termination criteria more intuitive to choose.

# ACKNOWLEDGMENT

This work was supported by Innovation Fund Denmark as a part of the project “MADE Digital”.

# REFERENCES


Agresti, A. and Coull, B. A. (1998). Approximate Is Better than "Exact" for Interval Estimation of Binomial Proportions. *The American Statistician*.

Boothroyd, G. (2005). *Assembly Automation and Product design*. CRC Press, 2nd ed. edition.

Brochu, E., Cora, V. M., and de Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR*.

Brown, L. D., Cai, T. T., and Dasgupta, A. (2001). Interval estimation for a binomial proportion. *Statistical Science*.

Härdle, W., Werwatz, A., Müller, M., and Sperlich, S. (2004). *Nonparametric and semiparametric models*. Springer Berlin Heidelberg.

Laursen, J., Sorensen, L., Schultz, U., Ellekilde, L.-P., and Kraft, D. (2018). Adapting parameterized motions using iterative learning and online collision detection. pages 7587–7594.

Mathiesen, S., Sørensen, L. C., Kraft, D., and Ellekilde, L.-P. (2018). Optimisation of trap design for vibratory bowl feeders. pages 3467–3474.

Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA.

Ross, S. M. (2009). *Introduction to Probability and Statistics for Engineers and Scientists*. Acedemic Press, 4th edition.

Sørensen, L. C., Buch, J. P., Petersen, H. G., and Kraft, D. (2016). Online action learning using kernel density estimation for quick discovery of good parameters for peg-in-hole insertion. In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics*.

Tesch, M., Schneider, J. G., and Choset, H. (2013). Expensive function optimization with stochastic binary outcomes. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*.


# APPENDIX

## The Effect of the Bias and Variance Error in Relation to KDE and WSKDE

The true confidence interval consists of both a bias and variance error, however, the bias term has to be neglected to make confidence interval calculable (see (9)). The variance term includes $f(x)$ which can be approximated by $\hat{f}(x)$, but, unfortunately, the bias term also includes $m'(x)$, $m''(x)$ and $f'(x)$, which cannot be approximated properly. Note, the bias and variance errors can be suppressed by letting $h \to 0$ and $nh \to \infty$ respectively.

In general, the bias is the vertical difference between the estimate and the true function and arises from smoothing effect. This smoothing effect drags down maxima and pulls up minima of the function estimate, $\hat{m}(x)$, compared to $m(x)$. In addition, the bias is proportional to only $m''(x)$ in extrema. Hence, neglecting the bias error but assuming that $m''(x)$ does not displace the optimum with respect to $x$, then $\hat{x}_{opt} = x_{opt}$ even though $\max(\hat{m}(x)) < \max(m(x))$. This assumption requires that important function details are not smoothed-out and is acceptable when choosing $h$ appropriately. Furthermore, neglecting the bias error will offset the confidence interval estimate compared to the true confidence interval such that the estimated bounds are raised at minima and lowered at maxima. For further details see (Härdle et al., 2004).

Neglecting the KDE regression bias error will also be reflected in the WSKDE mean and confidence interval estimates, since the KDE regression mean, $\hat{m}_h$, directly replaces the Normal Approximation mean, $\hat{p}_{na}$, as shown in (20). However, the bias error will be suppressed in sparsely sampled regions due to the few samples correction of WS (the WS confidence interval goes towards $[0;1]$ with mean of 0.5 when $n \to 0$). Regardless the neglection of the KDE regression bias error, our derivation of WSKDE is still valid since it is only based on a comparison of the variance terms of WS and KDE.

## Generalization to Multiple Dimensions

The equations of KDE and WSKDE can be generalized to multiple dimensions. Hence, the kernel, $K$, becomes a multi-dimensional kernel with bandwidth matrix $H$, which must be symmetric and positive definite. Whenever the bandwidth, $h$, is used as a scalar as in (9) or (20), this becomes the determinant of the bandwidth matrix $|H|$. For a multi-normal Gaussian kernel, $||K||_2^2$ is calculated as $1/(2^d\sqrt{\pi^d})$ where $d$ is the number of dimension, and this constant scalar is therefore not dependent on the bandwidth of the kernel. Note, the discrete function estimators NA and WS do not change when going to multiple dimensions, since these are only related to a certain parameter set without the influence of experiments made in neighboring region as when using kernel smoothing.