ON USING ADAPTIVE SURROGATE MODELING IN DESIGN FOR EFFICIENT FLUID POWER

Lakshmi Gururaja Rao, Jonathon Schuh, Randy H. Ewoldt, James T. Allison
University of Illinois at Urbana-Champaign, Urbana, IL, 61801
Email: {lrao2,schuh4,ewoldt,jtalliso}@illinois.edu

ABSTRACT
In the last several decades fluid power has been used extensively in diverse industries such as agriculture, construction, marine, offshore resource extraction, and even entertainment. With a vast and ever-increasing spectrum of potential applications, the design of efficient and leak-free components in fluid power systems has become essential. Previous experiments and studies have shown that the use of microtextured surfaces in hydraulic components achieves performance enhancement by reducing friction and leakage. This article aims to build on this recent work through a systematic optimization-based study of performance improvement through microtexture surface design. These studies evaluate the potential of Newtonian fluid properties, coupled with varying surface features, to achieve design objectives for efficiency. This early-stage design strategy aims to find optimal surface features that minimize apparent fluid viscosity (low friction) and the area of the microtexture. The resulting multi-objective optimization (MOO) problem involves a computationally intensive simulation of the system based on computational fluid dynamics (CFD). As a strategy to reduce overall computational expense, this paper describes the development of a new adaptive surrogate modeling strategy for multi-objective optimization. Two case studies are presented: a simple analytical case study illustrating the details of the method and a more sophisticated case study involving the two-dimensional CFD simulation of Newtonian fluids on symmetric surface textures. This design approach embraces the potential of using rheologically complex fluids in engineering system design and optimization. This study can be further extended to a more generalized problem by coupling both fluid and geometrical design decisions.

1 INTRODUCTION
Many engineering applications involve multiple design objectives, typically centered on enhancing performance and reducing cost. The trade-off between conflicting objectives can be investigated using Multi-Objective Optimization (MOO) tools. The solution of a MOO problem involves a set of solutions, known as the Pareto-optimal set (or Pareto set), that consists of non-dominated optimal points and quantifies design trade-offs in the objective function space. Many strategies exist for addressing MOO design problems [1, 2], some of which include condensing the multiple objectives into a single objective function by using a weighted sum or a geometric mean. An alternative treats all but one objective as a constraint. In either case multiple optimization solutions are required to build up the Pareto set, each time varying objective weights or constraint bounds. Another class of MOO solution methods are evolutionary algorithms such as the Multi-Objective Genetic Algorithm (MOGA) and the Non-dominated Sorting Genetic Algorithm (NSGA-II) [3, 4], both of which produce multiple Pareto-optimal solutions in a single optimization execution.

Evaluating design performance of engineering systems as part of design optimization studies often involves computationally intensive simulations. Despite advances in computing power in the recent past, computationally intensive analysis methods (e.g., Computational Fluid Dynamics (CFD)) can be impractical to use with optimization directly [5]. Important fluid power system design objectives, such as friction across sliding contacts, often require an expensive CFD simulation of the Navier-Stokes equation for each new design candidate. One effective strategy for reducing the total number of high-fidelity simulations...
required to solve a design optimization problem is optimize using an approximate surrogate model (SM)—sometimes called a meta-model—of the high-fidelity model [6, 7]. Accurate solution requires acceptable SM accuracy in the neighborhood of the optimum. Adaptive surrogate modeling methods iteratively assess SM accuracy, and evaluate the high-fidelity model at new sample points to improve SM accuracy if needed in strategic regions of the design space [8, 9]. The cost of evaluating sample points using the high-fidelity model must be accounted for when comparing SM to direct optimization.

SM has been investigated for specific application domains, such as structural design [10] and injection molding [11]. While most studies have focused on single-objective problems (e.g., Ref. [12]), extensions of SM to MOO have been made [11]. This article presents advancements of adaptive SM for MOO that aim to reduce the number of required simulations further. New features of this method include enforcement of constraints during high-fidelity model sampling. The algorithmic approach is elucidated using a mathematical example, followed by a physical engineering application that employs a CFD simulation of Newtonian fluids on microtextures.

2 ADAPTIVE SM METHODOLOGY

In this section the adaptive surrogate modeling (ASM) method is explained in detail, including steps for formulating the multi-objective optimization problem and implementing the adaptive strategy for incrementally improving surrogate model accuracy in strategic design space regions.

2.1 MOO Problem Formulation

Formulating an underlying engineering design problem as a mathematical optimization problem involves identifying design objectives, variables, and constraints based on a firm understanding of the physical system and design requirements and intent. A multi-objective optimization problem can be formulated as follows:

\[ \begin{align*}
\min_{\mathbf{x} = [x_1, x_2, \ldots, x_n]^T} & \quad f(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T, \\
\text{subject to} & \quad g_i(\mathbf{x}) = [g_{i1}(\mathbf{x}), \ldots, g_{ip}(\mathbf{x})]^T \leq 0 \\
& \quad h_i(\mathbf{x}) = [h_{i1}(\mathbf{x}), \ldots, h_{iq}(\mathbf{x})]^T = 0 \\
& \quad \mathbf{x}_{LB} \leq \mathbf{x} \leq \mathbf{x}_{UB}
\end{align*} \]  

(1)

where \( f(\mathbf{x}) \) is a vector-valued objective function, \( \mathbf{x} \) is the vector of design variables, \( g(\cdot) \) and \( h(\cdot) \) are constraint functions, and \( \mathbf{x}_{LB} \) and \( \mathbf{x}_{UB} \) are design variable bounds.

2.2 SM-Based Multi-Objective Optimization

The next step in solving Prob. (1) using a surrogate modeling strategy is to construct a model based on training data, i.e., a set of design points and the corresponding objective and constraint function values based on high-fidelity simulation. The SM is computationally inexpensive, allowing many function evaluations required for identifying an optimal design. The resulting design solution is only optimal with respect to the SM, so SM accuracy at the predicted optimum must be assessed by checking high-fidelity model outputs at this point. One of three strategies typically are used for ensuring accurate solution (Fig. 1).

![FIGURE 1: Strategies for SM-based design optimization: (a) Sequential, (b) Adaptive, (c) Direct Sampling [6]](image)

The first (Fig. 1(a)) is a sequential approach with a single validation check after SM construction but before optimization. This requires error assessment across the entire modeling domain to ensure accurate solution, involving a large number of validation points that must be evaluated using the high-fidelity model. Sequential sampling methods that treat the model development and optimization as independent tasks. Efficiency can be improved by adopting an iterative method where additional training data is obtained via additional sampling to improve SM accuracy if model validation fails (Fig. 1(b)). Wang et al. studied adaptive sampling as a means to improve a surrogate model prior to (single objective) optimization [13]. In this case, both the optimization and model validation points (a more global approach) are used to form a new sample set. High-fidelity model evaluations can be reduced further by using a direct sampling approach where new sample points are concentrated near the optimum predicted via SM optimization (Fig. 1(c)).

Additional challenges are introduced for MOO problems. Instead of needing to ensure SM accuracy in the neighborhood of a single predicted optimal point, the SM must be accurate in the neighborhood of the set of Pareto-optimal designs. Wilson et al. introduced an approach that ensured model validation and accuracy prior to identifying the Pareto-set [14]. Li et al. approximated Pareto sets using a hyper-ellipse [15]. Shan and Wang developed a new method—Pareto-Set Pursuing (PSP)—that ad-
dresses the challenges of solving MOO problems based on computationally expensive models [16]. The Adaptive Surrogate Modeling method for Multi-Objective Optimization (ASM-MOO) methodology presented here is a direct sampling approach. The case study involves a challenging fluid-flow system design problem, but the methodology can be extended to a general class of design problems. We introduce design constraint enforcement at the sampling stage, and study the resulting efficiency of this modified sampling method.

3 ASM-MOO Algorithm Framework

A more detailed illustration of the ASM-MOO methodology is shown in Fig. 2. The modeling domain $D_m$ is the region of the design space over which the SM is constructed. This may be fixed, or may contract and shift to conserve modeling efforts. The superscript $k$ is the iteration counter. The sampling domain $D_s$ is the design space region over which new sample points are defined; design sample points and their corresponding simulation outputs are used for the training dataset $T$.

![ASM-MOO algorithm](image)

**FIGURE 2:** ASM-MOO algorithm

During initialization the sampling domain $D_s$ is defined to be the same as the modeling domain, as no information is available yet to help focus sampling efforts near the predicted Pareto-optimal points. It is desirable to avoid using sample points that violate design constraints for two reasons: 1) Often infeasible points cannot be simulated [17], and 2) even if simulation is possible, simulation of infeasible points is wasted effort as SM accuracy is unimportant in infeasible regions. Here a set of preliminary samples are generated using one of several appropriate space-filling sampling techniques (such as Latin Hypercube) in the preliminary sampling domain $D_s$. This domain may include infeasible points. This set of sample points is then filtered to remove infeasible points, i.e., retained sample points belong to the set $D_r = D_s \in F$, where $F$ is the feasible domain (i.e., the set of points that satisfy design constraints). In the studies presented here design constraints are analytical, so SMs for design constraints are not needed. Future work will address the case where design constraints depend on simulation outputs.

$T$ is then formed by evaluating each retained sample point using the high-fidelity model, and organizing these data pairs. The SM is then constructed using $T$. Here the specific approximation Here radial basis functions are used to create the model [18]. The MOO problem is then solved using an appropriate algorithm (MOGA here) and function evaluations based on the SM to obtain a set of approximately Pareto-optimal points $x_k$.

Model validation is performed by assessing SM error at points in the Pareto set (analogous to assessing SM error at the predicted optimum for single-objective problems). If the selected error metric does not satisfy a specified error tolerance, the process is repeated by generating new sample points near the current Pareto set, running the high-fidelity model at these new points, and adding these points to $T$. A new (more accurate) SM is formed, and optimization is repeated. Additional details for these steps are presented below.

3.1 Sampling

The modeling domain $D_m$ is defined here by the design variable bounds. Due to the SM type and the nature of the case studies, $D_m$ was not adjusted during the ASM-MOO process. Here $D_r = D_m$, and a suitable design of experiments (DOE) technique is used to generate sample points that lie in $D_r$. The appropriate choice of sampling technique depends on factors such as (i) design space dimension, (ii) estimated level of response noise, and (iii) SM type. For noisy data (e.g., probabilistic models), DOEs that reduce the sensitivity to noise are desirable, such as central composite designs, face-centered cubic designs, factorial designs, and Box-Behnken designs [19]. For systems with less noise (such as deterministic models), Latin-Hypercube Sampling (LHS), minimum bias designs, and Orthogonal Arrays (OAs) are preferred. Here LHS is used, which is an example of a space-filling design that is often effective for computer experiments without variance [20–22], including adaptive surrogate modeling [23] and in conjunction with OAs [24].
3.2 Filtering Infeasible Sampling Points

Engineering design problems often involve inequality constraints $g(\cdot) \leq 0$. The sampling procedure described above ensures that only feasible design points are evaluated using the high-fidelity model, improving overall efficiency. In some cases this strategy offers another important advantage: sample points that violate constraints may not be physically meaningful, or for other reasons would cause simulation failure. Removing infeasible sample points preemptively avoids these simulation problems that are otherwise difficult to manage in an optimization implementation [17]. In this article we investigate how filtering out infeasible sample points influences solution efficiency, which is a new concept in ASM.

While this article addresses only analytically defined linear and non-linear constraints, we acknowledge the need to extend this study to constraints that require the simulation of the high-fidelity model. Filtering out infeasible points based on approximated constraint functions is an important topic for future work. It is expected that additional challenges will be encountered as there is uncertainty in whether sample points are actually infeasible during the ASM solution process.

3.3 Surrogate Model Construction

Once the set of feasible sample points is defined, these design points are evaluated using the high-fidelity model. This produces input-output pairs that are added to the set of training data $T$ for use in SM construction. A wide variety of approximation function types may be used for SM construction. Several general-purpose options are described below. Application-specific SMs may help improve solution efficiency further.

General purpose approximation functions for SMs help to estimate simulation output at design points other than those in $T$. Well-known options include Kriging [22, 25], artificial neural networks (ANNs) [26], and radial basis functions [27, 28]. Other methods include multivariate adaptive regression splines (MARS) [29], least interpolating polynomials [30], and inductive learning [31]. Jin et al. compared several different surrogate models and concluded that the appropriate model choice depends on the degree of system nonlinearity [32]. Stander et al. compared polynomial response surface approximation, Kriging, and neural networks [33], and discovered that while neural networks and Kriging models require a larger number of initial points, their algorithmic efficiencies are comparable to a polynomial model. The general conclusion from the literature demonstrates that there is no single best SM type that satisfies all problems. Haftka et al. address this issue by exploring the use of an ensemble or a weighted average SM in place of individual surrogates [34]. The best model choice depends on the nature of the problem and the sampled data set.

Radial basis functions have been used extensively as SMs for engineering design optimization [18, 35]. Fang et al. compared polynomial response and radial basis functions and found that RBFs are more effective at handling non-linearities in some cases [36]. RBF models have been selected for use in this article. Comparison with other model types is an important topic for future work.

When using an RBF approximation for multi-objective problems, an approximate function is defined for each of the objectives. An approximate objective function can be defined: $\hat{f}_j(x) = \sum_{i=1}^n w_j^i \psi(||x_i - c_i||)$, $i = 1, 2 \ldots n_i$, $j = 1, 2 \ldots m$

where $\psi(.)$ is the radial basis function, $w_j^i$ are unknown weighting coefficients, $x_i$ is the $i$th training point, and $c_i$ is the $k$th basis function center. The RBF used here is the thin plate spline function [37]: $\psi(r) = r^2 \ln r$, where $r$ is the Euclidean distance between the training point and function center: $r = (||x_i - c_i||)$. 

During SM construction we seek to find the coefficients $w_j^i$ that solve $\psi w^j = f_j$, where $\psi_{j,k} = \psi(||x_i - c_i||)$, $i, k = 1, 2 \ldots n_i$ is the Gram matrix, and the vector of training outputs for the $j$th objective function is $f_j = [f_j(x_1), f_j(x_2), \ldots, f_j(x_n)]^T$ [37]. This equation has a unique solution since the Gram matrix is square. To reduce problem complexity here we assume that the RBF centers coincide with training points, i.e., $c_i = x_i$. Once the coefficients $w_j^i$ are identified for each objective function the approximate objective functions can be defined: $\hat{f}(x) = [\hat{f}_1(x), \ldots, \hat{f}_m(x)]^T$. For a given point in the design space $x$, we can compute the row vector $\psi(x) = [\psi(||x - c_1||), \ldots, \psi(||x - c_n||)]$. The surrogate model for the $j$th objective function is $\hat{f}_j(x) = \psi(x) w^j$.

3.4 Surrogate Model Optimization

The multi-objective design optimization problem is then solved by replacing the original nonlinear functions in Prob. (1) with approximate functions obtained via surrogate modeling. This produces an approximate Pareto set that is only accepted as the solution if the validation step is passed. If all objective and constraint functions are approximated using surrogates, the updated problem formulation is:

$$
\min_{x=[x_1, x_2, \ldots, x_n]^T} \hat{f}(x) = [\hat{f}_1(x), \ldots, \hat{f}_m(x)]^T,
$$

subject to

$$
\begin{align*}
\hat{g}(x) &= [\hat{g}_1(x), \ldots, \hat{g}_p(x)]^T \leq 0 \\
\hat{h}(x) &= [\hat{h}_1(x), \ldots, \hat{h}_q(x)]^T = 0 \\
x_{LB} &\leq x \leq x_{UB}
\end{align*}
$$

This MOO problem was solved here using a Multi-Objective
3.5 Surrogate Model Validation

Surrogate models used in design optimization need to be accurate in the region of the optimum. Guaranteeing accuracy within a tolerance at other points in the design points is wasteful of resources. Validation is straightforward for single objective problems where SM accuracy must be verified near the single predicted optimal point. This can be accomplished by evaluating the high-fidelity model at the predicted optimum, and comparing the result to the SM output at this point. SM validation is more involved for MOO problems. We need to ensure that the SM is accurate for points in the design space that are Pareto-optimal. Instead of validating the SM at a single point, it must be validated across a bounded hypersurface. For this purpose we define the validation domain \( D_v \) to be the set of points over which the SM must be validated. If the design space is in \( \mathbb{R}^n \), the hypersurface that defines \( D_v \) is a manifold of dimension \( n - 1 \).

Validating over bounded hypersurface introduces two core challenges: 1) Instead of validating a single point in the design space, validation must be performed across an infinite number of points on a hypersurface, and 2) The boundaries of this hypersurface may be complicated (possibly non-convex or even disconnected), and we should focus validation efforts on points within the hypersurface bounds. The first challenge is addressed by using regression to fit a surface through Pareto-optimal points in the design space to define an approximate (unbounded) validation domain. A uniform sampling technique is then used to select a finite set of validation points from \( D_v \) for validation. Generating these new validation points in \( D_v \) is required because simply using the Pareto-optimal points identified by the MOGA as validation points may not be adequate due to non-uniform distribution in the design space. The error between the SM and the high-fidelity model is evaluated at these validation points, and these error quantities are combined into a single error metric for SM validation.

In addressing the second challenge we recognize that not every point on an unbounded regression surface that is fit through Pareto-optimal points is in the Pareto set. We must identify and stay within hypersurface bounds to ensure that validation points are (approximately) in the Pareto set.

Several options exist for defining the validation hypersurface \( D_v \), including simple regression, non-uniform rational basis splines (NURBS) [40], or T-splines. Quadratic regression performed well for the small-dimension case studies used here. Support Vector Data Description (SVDD) could be used to define a precise hypersurface boundary that includes all known Pareto-optimal points [17]. Rather than defining a precise boundary, we developed an algorithm that generated random points on the hypersurface that were within a convex hull of a subset of the known Pareto-optimal points. This provided an efficient means of generating validation points that were approximately in the true \( D_v \).

Once validation points are chosen, the high-fidelity model is evaluated at these points for comparison to the SM to evaluate accuracy. Several options for SM validation exist, but standard methods are intended primarily for cases where accuracy is being assessed across \( D_m \). Cross-validation is a well-known model validation method, but is computationally cumbersome. Meckesheimer suggested a leave-one-out cross-validation strategy, applicable to radial basis functions [41, 42]. The method used here is to evaluate the standard normal error (SNE) using points in \( D_v \). For a set of \( n_v \) validation points, the norm of the difference between the actual function value and the approximated function value is defined as the SNE:

\[
SNE = \sum_{i=1}^{n_v} || \left(f(x_i) - \hat{f}(x_i)\right)||
\]

where \( x_i \) is the design vector value for the \( i \)th validation point.

3.6 Sampling Domain Update

If the \( SNE \geq \varepsilon_v \), where \( \varepsilon_v \) is a specified validation tolerance, model accuracy must be improved in the neighborhood of the Pareto set. This is done by generating new sample points that are in \( D_u \), which is the updated sampling domain that includes a small neighborhood of the design space near \( D_v \). The high-fidelity model is evaluated at these new sample points, and these data pairs (along with the validation points from the previous set) are appended to \( T \) for use in constructing the updated SM with improved accuracy near the estimated Pareto set.

Sample points need to be defined that lie within \( \bar{D}_u \), which is a hypervolume. One approach would be to define the boundary of this hypervolume directly (using SVDD, level set methods, etc.) and filter out sample points that are not within this boundary. A classification method such as a Support Vector Machine (SVM) could be used to determine whether candidate sample points are within \( \bar{D}_u \). Alternatively, force-directed layout or field functions could be used to increase the density of sample points near \( D_v \).

The method used here is an Octree-based sampling method [43] that generates uniformly distributed points in the neighborhood of validation points as shown in Fig. 3. Sample point density can be adjusted as the ASM-MOO algorithm progresses. Given a specified distance tolerance, points lying far from \( D_v \) are eliminated to produce \( \bar{D}_u \). In addition, only feasible sample points are retained. \( D_u = \bar{D}_u \in F \) defines the updated feasible sampling domain that contains new sample points to be simulated.
FIGURE 3: Octree-based sampling algorithm (two design variable example: \(x_1\) and \(x_2\)). With an initial set of points to define the domain to sample from as shown in (a), the octree-based algorithm uniformly samples points within these bounds sparsely in (b) and more densely in (c) depending on sampling requirements.

Figure 4 illustrates conceptually the validation and sample point generation strategies described above using a two-dimensional design space. Yellow markers are MOGA Pareto points. The solid black line is the validation domain \(D_v\) obtained via regression. Dark blue markers are the uniformly distributed validation points. The red curve is the design constraint boundary (up and to the right in infeasible). Red markers are new (infeasible) Octree sample points. Magenta markers are feasible sample points that are too far away from \(D_v\) to use. Cyan markers are the feasible sample points in \(D_u\).

3.7 Modeling Domain Update

The modeling domain \(D_m\) can either remain static or can be contracted and re-centered to ease model construction. \(D_m\) contraction can be accompanied through a simple iterative formula:

\[
\Delta_{k+1}^m = \beta \Delta_k^m
\]

where \(\Delta_k^m\) is the range of \(x_i\) at the \(k\)th ASM-MOO iteration, and \(\beta < 1\) is a fixed contraction parameter. One strategy for re-centering the modeling domain would be to align its center with the centroid of \(D_v\). Rapid changes in \(D_m\) center may cause numerical instabilities, which could be ameliorated with numerical damping (e.g., the heavy ball method). Here \(D_m\) remains fixed. Variable bounds are determined by experimentally appropriate limits.

3.8 Training Data Set Update

Before construction of the updated SM, the training data set is augmented with both the validation data and new sample points: \(T^{k+1} = T^k \cup D_u^k \cup D_v^k\), where \(k\) denotes the \(k\)th iteration of the ASM-MOO algorithm.

4 NUMERICAL RESULTS AND DISCUSSION

The ASM-MOO algorithm is demonstrated first using a simple analytical multi-objective constrained optimization problem, and then using a more sophisticated case study involving design of efficient fluid power systems.

4.1 Analytical Example

Consider the following MOO problem:

\[
\begin{align*}
\min_{x=[x_1,x_2]^T} & \quad f(x) = [f_1(x), f_2(x)]^T, \\
\text{where} & \quad f_1(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \\
& \quad f_2(x) = (x_1)^2 + (x_2 - 6)^2 \\
\text{subject to} & \quad -2.5x_1 + x_2 \leq 1 \\
& \quad (x_2 - 2)^2 \leq 2 \\
& \quad 0.4 \leq x_1 \leq 1.6, \quad 2 \leq x_2 \leq 5
\end{align*}
\]

(5)

This problem was solved in two ways: 1) using the MOGA algorithm available in MATLAB® with direct objective function evaluation, and 2) using the ASM-MOO with the same MOGA algorithm. The result of the first solution approach is the baseline or ‘true’ solution. Both Pareto-MOO sets are shown in Fig. 5, and the error values are presented in Table 1.

If we allow the same number of total original function evaluations for the ASM-MOO method, using sample constraints to eliminate infeasible sample points results in slightly improved solution accuracy based on SNE. The total number of function evaluations was made approximately equal by increasing the number of initial sample points for the case with sample constraints. Additional benefits of sampling constraints may be realized with simulation-based problems (e.g., preventing simulation failure by avoiding infeasible designs).

4.2 Design for Efficient Fluid Power

A hydraulic system design study motivated the development of the ASM-MOO method introduced in this article. This study
involves the investigation of different surface texture designs for components used in fluid power systems where metal parts are in sliding contact separated by hydraulic fluid. The competing design objectives investigated here include reducing effective viscosity (related to friction) and the cross-sectional area of the texture (related to cost).

Previous studies have shown that the use of surface textures aids in decreasing friction in lubricated sliding contact [44]. Schuh and Ewoldt [45] have experimentally examined the effects of using symmetric and asymmetric textures to decrease viscous friction and increase normal force production with Newtonian lubricants. For the experiments, a DHR-3 gap controlled rotational rheometer was used with the textured plates mounted to a temperature controlled Peltier plate with Crystalbond, a thermoreversible adhesive. A schematic of the experimental set up is given in Fig. 7, and the measured geometric properties of the textures tested are given in Fig. 8.

FIGURE 5: Comparison between Pareto sets obtained via direct optimization and via ASM-MOO

FIGURE 6: Pareto-values and the update domain in the design space.

FIGURE 7: Experimental setup. Design variables are defined in Eqn. (6).

In their experiments, Schuh and Ewoldt found that symmetric textures decreased the viscous friction more than the asymmetric textures, but the asymmetric textures produced larger normal forces than the symmetric ones (important for sealing).

Since each texture profile met both of the given design ob-
FIGURE 8: Measured geometric properties

Training data was obtained via FLUENT, a finite volume-based CFD solver. Symmetric and asymmetric texture profiles were simulated at steady state in FLUENT using a Newtonian lubricant with a viscosity of 0.140 Pa s and a density of 866.8 kg/m$^3$. The simulations were performed using double precision and parallel computing with two processes. The pressure was solved using the SIMPLE algorithm and the momentum was solved using the second order upwind technique. A hybrid initialization scheme was used, and the simulation was run until all the residuals for the system were less than 1 $\times$ 10$^{-5}$. $h$ was fixed at 0.25 mm. Several different values of $W$, $D$, and $L$ were simulated in order to examine different texture profiles. The results from FLUENT were given as a normal force and viscous force, and the viscous force was converted to an apparent viscosity using:

$$\eta_a = \frac{F_v}{WLU}$$  \hfill (7)

where $U$ is the velocity of the top plate, which for these simulations was set as $U = 1.425$m/s.

The design constraint $x_1 \leq 0.95 x_3$ ensures that texture width is no greater than 95% of the length of the moving plate. The simulation would be infeasible (or produce insensible output) if the constraint is violated. RBFs were used as the surrogate model here. ASM-MOO was then applied to the design system to identify the optimal texture features. In this case, the validation domain $D_v$ is a hypersurface as we have a 3-dimensional design space (Fig. 9).

FIGURE 9: The Pareto-points, validation points, and updated sample points for the case study. The surface shown represents the validation domain, $D_v$. Results presented here are preliminary and limited to the case of symmetric textures. The result is visualized as a Pareto set in Fig. 10. The Pareto-set obtained is compared with the CFD points (obtained while building the training data set). We can see that the Pareto-set is approaching the true Pareto-frontier of the simulated points in Fig. 10. It is insightful to see that $\eta_a / \eta_0 < 1$ can be achieved with a very small texture area. The extreme Pareto-point (corresponding to $\eta_a / \eta_0 > 1$) is not physically meaningful, but it is expected that the ASM-MOO algorithm will allow for its correction as the iterations proceed and the surrogate model accuracy is improved. Table 2 reports the Pareto-set values for both CFD data points and the ASM-MOO
algorithm. This tradeoff information is valuable for designers who are interested in determining a texture design that best meets overall system design requirements.

![FIGURE 10: The CFD simulation outputs (gray markers) are shown along with the Pareto-set that is identified by the ASM-MOO algorithm (orange circles). By plotting all CFD simulation results we can qualitatively identify the location of the true Pareto set at the lower-left boundary of the attainable set. The Pareto-set shown here is the result of an initial set of six ASM-MOO iterations, and does not represent the converged optimal solution. Nonetheless, it is clear that the ASM-MOO algorithm is moving toward the true Pareto-set.]

![TABLE 2: The labeled points in Fig. 10 are shown with numeric values of both the design variable and the objective functions. Rows 2 and 3 are values obtained from the ASM-MOO algorithm. Rows 1 and 4 are the Pareto-values that one can identify by plotting the CFD data points (shown as grey markers in Fig. 10).](image)

<table>
<thead>
<tr>
<th>$\eta_a/\eta_0$</th>
<th>Area (mm²)</th>
<th>W (mm)</th>
<th>D (mm)</th>
<th>L (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.518</td>
<td>10.719</td>
<td>5.797</td>
<td>1.849</td>
<td>6.260</td>
</tr>
<tr>
<td>0.527</td>
<td>8.000</td>
<td>4.202</td>
<td>1.978</td>
<td>4.432</td>
</tr>
<tr>
<td>0.775</td>
<td>0.578</td>
<td>3.453</td>
<td>0.090</td>
<td>3.772</td>
</tr>
<tr>
<td>1.00</td>
<td>1.4e-5</td>
<td>0.0039</td>
<td>0.0036</td>
<td>3.312</td>
</tr>
</tbody>
</table>

5 CONCLUSIONS AND FUTURE WORK

Engineering design optimization problems are often characterized by multiple objectives and expensive simulations. As a strategy for reducing computational expense, surrogate modeling strategies have been developed and applied across a variety of engineering domains. In this article, one such strategy, Adaptive Surrogate Model coupled with Multi-Objective Optimization (ASM-MOO), was presented. A systematic formulation of the multi-objective problem and a detailed framework for the adaptive surrogate modeling was outlined. This algorithm also incorporated the use of design constraints in the sampling phase, which is an essential feature particularly in modeling systems that cannot be simulated in analytical example and is further extended to a fluid power component problem.

Future investigations will involve nonlinear sample constraints and design constraints evaluated using a SM. Alternative strategies for generating feasible samples should be explored (e.g., mapping a hypercube to $F$ in the neighborhood of $D_v$). The case study provides great insights into plausible microtexture geometries for different design needs. However, it is limited currently to symmetric textures; ongoing work is addressing the case of asymmetric textures as well. Aside from texture geometry, we envision studying a larger optimization problem by including fluid properties such as viscosity, density etc., in the design space (i.e., simultaneous texture and fluid design). Other improvements include extension to 3D problems and non-Newtonian hydraulic fluids. The ASM-MOO method defined here is a local search algorithm; samples are taken from the neighborhood of the approximate Pareto set. As an extension, we would like to expand it to a global method by also sampling in regions of low information to improve global search properties.

6 ACKNOWLEDGEMENTS

This work was supported by the Engineering Research Center for Compact and Efficient Fluid Power (CCEFP), supported by the National Science Foundation under Grant no. EEC-0540834. The co-authors thank Mr. Yong Hoon Lee for his assistance with numerical studies, and Mr. Nathaniel Bristow and Ms. Madeline Popelka for their assistance with the experimental setup and data collection.

REFERENCES


[33] Stander, N., Roux, W., Giger, M., Redhe, M., Fedorova,


