

Constrained Bayesian Optimization for Automatic Underwater Vehicle Hull Design

Harsh Vardhan
harsh.vardhan@vanderbilt.edu
Vanderbilt University
Nashville, TN, USA

Peter Volgyesi
peter.volgyesi@vanderbilt.edu
Vanderbilt University
Nashville, TN, USA

Will Hedgecock
ronald.w.hedgecock@vanderbilt.edu
Vanderbilt University
Nashville, TN, USA

Janos Sztipanovits
janos.sztipanovits@vanderbilt.edu
Vanderbilt University
Nashville, TN, USA

ABSTRACT

Automatic underwater vehicle hull Design optimization is a complex engineering process for generating a UUV hull with optimized properties on a given requirement. First, it involves the integration of involved computationally complex engineering simulation tools. Second, it needs integration of a sample efficient optimization framework with the integrated toolchain. To this end, we integrated the CAD tool called FreeCAD with CFD tool openFoam for automatic design evaluation. For optimization, we chose Bayesian optimization (BO), which is a well-known technique developed for optimizing time-consuming expensive engineering simulations and has proven to be very sample efficient in a variety of problems, including hyper-parameter tuning and experimental design. During the optimization process, we can handle infeasible design as constraints integrated into the optimization process. By integrating domain-specific toolchain with AI-based optimization, we executed the automatic design optimization of underwater vehicle hull design. For empirical evaluation, we took two different use cases of real-world underwater vehicle design to validate the execution of our tool. The code for running the experimentation and installing the toolchain can be found here https://github.com/vardhah/ConstraintBOUUVHullDesign.

KEYWORDS

Bayesian Optimization, underwater vehicles, inequality constraints, computational fluid dynamics, CAD

ACM Reference Format:
Harsh Vardhan, Peter Volgyesi, Will Hedgecock, and Janos Sztipanovits. 2023. Constrained Bayesian Optimization for Automatic Underwater Vehicle Hull Design. In *Cyber-Physical Systems and Internet of Things Week 2023 (CPS-IoT Week Workshops '23), May 09–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3576914.3587530

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CPS-IoT Week Workshops '23, May 09–12, 2023, San Antonio, TX, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0049-1/23/05…$15.00
https://doi.org/10.1145/3576914.3587530

1 INTRODUCTION

Artificial Intelligence (AI) and Machine Learning (ML) are becoming increasingly useful for both system-level design [3, 28, 2, 24] and control [23, 1, 25]. However, its application to real-world designs is still nascent for a variety of reasons. The first challenge is that these areas involve complex tools and require cumbersome, integrated toolchains. Integrating these tools requires domain knowledge, as well as the development of tool-level integration platforms. The next challenge involves the complexity of carrying out evaluations due to the high cost of data labeling. In this work, we aimed to address both challenges in the context of an underwater vehicle hull design optimization problem. The design of an underwater vehicle hull involves multiple steps: CAD design of the hull, generation of stereolithography (STL) files, generation of a computational fluid dynamics (CFD) simulation environment, creation of volume meshing, fixing of the initial and boundary conditions, and solution of Navier-Stokes equations at the mesh level. For a fully automated design optimization process, it is necessary to integrate the execution of all of these processes in a single tool.

Another aspect of this work relates to optimization algorithms. There are many gradient-free and gradient-based algorithms available; however, in recent times, Bayesian Optimization (BO) [4, 16, 30] has emerged as a well-established paradigm for optimizing expensive-to-evaluate functions in a sample-efficient manner, and it has been successfully applied to many scientific domains. Bayesian Optimization is a complex optimization process that provides the benefits of both model-based and simulation-based optimization approaches. To this end, BO creates a probabilistic model of an unknown function during the optimization process and uses this model to simulate and search for the best candidate sample to be used in the next evaluation. Constraints may exist which make some of the designs and design space infeasible, and these must be handled appropriately during the optimization process. In this work, we have made the following contributions:

(1) A ready-to-use underwater vehicle design tool using a Myring hull-based parametric CAD seed design,
(2) Integration of a constrained Bayesian Optimization framework for UUV hull design problem.

This fully automated optimization toolchain can be used as a good starting point to study and test different optimization methods on

real-world complex problems, in addition to providing a useful tool for underwater vehicle hull designers.

2 PROBLEM FORMULATION AND APPROACH

In this section, we formulate the UUV hull design problem as a constrained optimization problem. The hull shape of an underwater vehicle is indicated as Ω and can be defined using a multivariate parameter x, that is, $x \triangleq \Omega$. If f is the objective function that maps a 3D shape Ω with a complex property of coupled two-way solid-fluid dynamics, that is, drag force (F_d) ($f : \Omega \mapsto F_d$.), then the optimization problem can be formulated as:

$$\Omega^* = \underset{x \in DS}{\operatorname{argmin}} f(x) \tag{1}$$

Here, DS is our design search space. The UUV hull contains electronics, sensors, and other mechanical and electrical components. Packing them into the hull imposes a non-linear constraint on the optimization process. The hull design problem can then be formulated as a constrained optimization problem defined as follows:

$$\Omega^* = \underset{x \in DS}{\operatorname{argmin}} f(X) \tag{2}$$

$$s.t.\ g(x) \leq 0 \tag{3}$$

Here, constraint function $g(x)$ ensures that all selected components can be packed inside the designed UUV hull. To solve this optimization problem, we utilize a constrained Bayesian Optimization framework as formulated by [8].

2.1 Constrained Bayesian Optimization

Bayesian Optimization relies on a probabilistic model of the system of interest during optimization, and the fidelity of the model is the most decisive factor in the optimization process. We use the Gaussian process [18] defined below to model system behavior (f):

$$f \sim \mathcal{GP}(\mu(.), \kappa(.,.)) \tag{4}$$

Here $\mu(.)$ is the mean function and $\kappa(.,.)$ is the covariance kernel. For any given pair of input points $x, x' \in R^d$, these are defined as:

$$\mu(x) = \mathbb{E}[f(x)] \tag{5}$$

$$\kappa(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x')] \tag{6}$$

In the Bayesian sequential design optimization process, a crucial step at each iteration is to select the most promising candidate x^* for evaluation in the next iteration. In the BO setting, this is done by defining an acquisition function. The design of an acquisition function is a critical component in the performance efficiency of the BO. Let x^+ be the best-evaluated sample so far. To select a candidate point $\hat{x}$ in the next iteration, an improvement is defined according to Mockus et al. [16] as follows:

$$I(\hat{x}) = max\{0, f(\hat{x}) - f(x^+)\} \tag{7}$$

The expected improvement in such a case is defined as an EI acquisition function, which has a closed-form solution for estimating it from a new candidate point, as given by Mockus et al. [16] and Jones et al. [14]:

$$EI(\hat{x}) = \mathbf{E}\big[I(\hat{x})|\hat{x})|\big] \tag{8}$$

$$EI(x^+) = (f(x^*) - \mu^+)\Phi(\frac{f(x^*) - \mu^+}{\sigma^+} + \sigma^+\phi(\frac{f(x^*) - \mu^+}{\sigma^+}) \tag{9}$$

Here, ϕ is the standard normal cumulative distribution and Φ is the standard normal probability density function. Using this EI function, the most promising candidate sample is selected by choosing x^+ that has the maximum EI value.

$$x^* = \underset{x^+ \in DS}{\operatorname{argmax}}\ EI(x^+) \tag{10}$$

The newly selected sample x^* is evaluated and is included in the evaluated data set, called X. Accordingly, the posterior probability distribution is estimated by the conditioning rules for Gaussian random variables, as below:

$$\mu^* = \mu(x^*) + \kappa(x^*, X)\kappa(X, X)^{-1}(f(X) - \mu(X)) \tag{11}$$

$$(\sigma^*)^2 = \kappa(x^*, x^*) - \kappa(x^*, X)\kappa(X, X)^{-1}\kappa(X, x^*) \tag{12}$$

Constrained BO, which is an extension to standard BO meant to model infeasibility during the inequality-constrained optimization routine, is formulated and proposed by [8]. We use this formulation for our experimentation, and it models both function and constraint as Gaussian processes. Let g be the constraint function that is unknown *a priori*; the first step in this setting is to model f and g as Gaussian processes:

$$f \sim \mathcal{GP}(\mu_1(x), \kappa_1(x)) \tag{13}$$

$$g \sim \mathcal{GP}(\mu_2(x), \kappa_2(x)) \tag{14}$$

$$\mu_1(x) = \mathbf{E}[f(x)] \tag{15}$$

$$\kappa_1(x, x') = \mathbf{E}[\{f(x) - \mu_1(x)\}\{f(x') - \mu_1(x')\}] \tag{16}$$

$$\mu_2(x) = \mathbf{E}[g(x)] \tag{17}$$

$$\kappa_2(x, x') = \mathbf{E}[\{g(x) - \mu_2(x)\}\{g(x') - \mu_2(x')\}] \tag{18}$$

The improvement function in this case is modified as:

$$I_C(x^+) = \Delta(x^+)max\{0, f(x^*) - f(x^+)\} \tag{19}$$

$$\Delta(x^+) \in \{0, 1\} \tag{20}$$

$\Delta(x^+)$ is a feasibility indicator function that is 1 if $g(x^+) \leq 0$, and 0 otherwise. It causes $\Delta(x^+)$ to be a Bernoulli random variable whose probability of getting a feasible design is:

$$PF(x^+) := Pr(g(x) \leq 0) = \int_{-\infty}^{0} P(g(x^+)|x^+, X)dg(x^+) \tag{21}$$

Due to the Gaussian behavior of $g(.)$, $\Delta(x^+)$ would be a univariate Gaussian random variable. The modified expected improvement to include the effect of infeasibility gives a joint acquisition function:

$$EI_C(x^+) = \mathbf{E}[I_C(x^+)|x^+] \tag{22}$$

$$= \mathbf{E}[\Delta(x^+)I(x^+)|x^+] \tag{23}$$

$$= PF(x^+)EI(x^+) \tag{24}$$

This joint acquisition function can be further optimized using standard optimization algorithms. Since our acquisition function has the property of being smooth and continuous, we used a two-step optimization to find x^*. The first step is Monte Carlo optimization and the second step is limited memory BFGS [7] (see Figure 1).

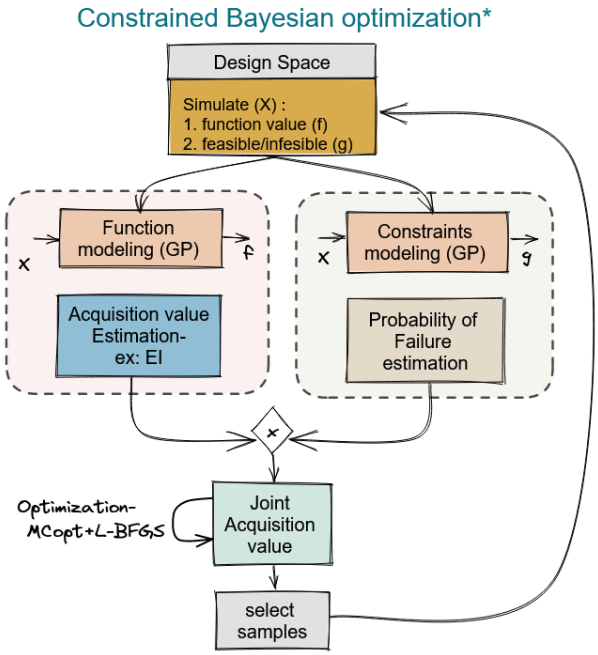


Figure 1: Constrained Bayesian Optimization - Overview

2.2 Integrated Toolchain

For design automation, exploration, and optimization, we integrated the necessary simulation tools for completely automated execution. To this end, we integrated the CAD design tool FreeCAD [19] with the CFD simulation tool OpenFoam [13]. FreeCAD is used to design a parametric CAD model and generate the 3D CAD geometry from a given set of parameters along with its stereolithography (STL) file without any manual intervention. OpenFoam uses this STL file to conduct fluid physics simulations via finite volume discretizations. Volume meshing is done using a castellated 3D parametric volumetric mesh which is further split and refined in the vicinity of the body surfaces. Other CFD simulation requirements, like solver settings and initial and boundary conditions, can also be set up from a Python environment. In the meshed volume, RANS with kw-SST turbulence fluid physics is solved, and the output of interest is fetched from OpenFoam and transferred back into the Python environment. Accordingly, it is possible to both control and run the entire design optimization pipeline from a single Python environment. This integration of tools and capability to control the parameters and environmental conditions gives us the flexibility to run an optimization framework with design tools in the loop without human intervention (refer to Figure 2).

2.3 Parametric CAD Model and Baseline Packing Geometry

For automatic design optimization, it was imperative to design a parametric CAD model with the flexibility and adaptability to be able to create a 3D model from a given set of parameters without manual intervention. The parametric CAD design should maintain the experimenter's assumptions in order to generate a *valid* CAD design based on the given parameters. To ensure this, we use a stringent design methodology for completely constrained designs [11]. For the parametric CAD seed design, we used a Myring hull[17] as our baseline architecture. It is the most commonly used axisymmetrical hull shape, due to a number of advantages such as streamlined flow behavior and satisfactory geometry for both hydrodynamic and hydrostatic pressure. A Myring hull has three different body sections: nose, tail, and cylindrical center:

The nose and tail equations for a Myring hull are given by:

$$r_1(x) = \frac{1}{2}D\left[1 - \left(\frac{x-a}{a}\right)^2\right]^{\frac{1}{n}} \tag{25}$$

$$z = (x - a - b) \tag{26}$$

$$r_2(x) = \frac{1}{2}D - \left[\frac{3D}{2c^2} - \frac{\tan\theta}{c}\right]z^2 + \left[\frac{D}{c^3} - \frac{\tan\theta}{c^2}\right]z^3. \tag{27}$$

Here, r_1 and r_2 define the radius of the nose and tail of the hull at a distance, x, measured from the tip of the nose. Other body parameters are: a, b, c, D, n, and θ, corresponding to nose length, body length, tail length, cylindrical body diameter, nose shaping parameter, and tail shaping parameter, respectively (see Figure 3).

The baseline design used for internal component packing and placement comes from another automated tool [15]. Component selection and packing are not within the scope of this paper; however, three-dimensional packing of components in an arbitrary shape is an NP-complete problem. Based on the capabilities of our external component selection and packing tool, we utilized a simple design with conical end caps (nose and tail) and a cylindrical body to determine the exact required hull dimensions. These conical-shaped parametric designs are optimized to minimize internal hull volume while ensuring that packed components have no interferences. These baseline designs, however, are not optimal from the perspective of producing minimal-drag designs. Once components are packed and the parameters of a baseline design are found, this fixed geometry will act as a minimal constraint in the optimization of the hull design. (see Figures 4 and 5).

2.4 Infeasible Design Heuristics

Since a parametric Myring hull can assume a wide range of shapes [17], the generated hull shape needs to be tested for interference with the baseline packed design. Any Myring hull parameters that cause interference with the baseline design are deemed to be infeasible. Since the computational cost of CAD assembly and running an interference test is much less than CFD simulation, the in-feasibility test on an optimized design is conducted during the CAD modeling and assembly stage (refer to Figure 2). Running full CFD analysis on an infeasible design is a waste of computational time and resources, and it delays the optimization process. To address this situation, we implemented a heuristic that works as follows (for a minimization problem): for an infeasible design that is detected during CAD assembly, return the maximum drag value to the optimizer for all evaluated samples up to that point, instead of running a full CFD analysis. The opposite can be done for a maximization problem. However, for starting the experiment we need at least one drag value of in-feasible design. Accordingly, we run the first infeasible design and store its drag value.

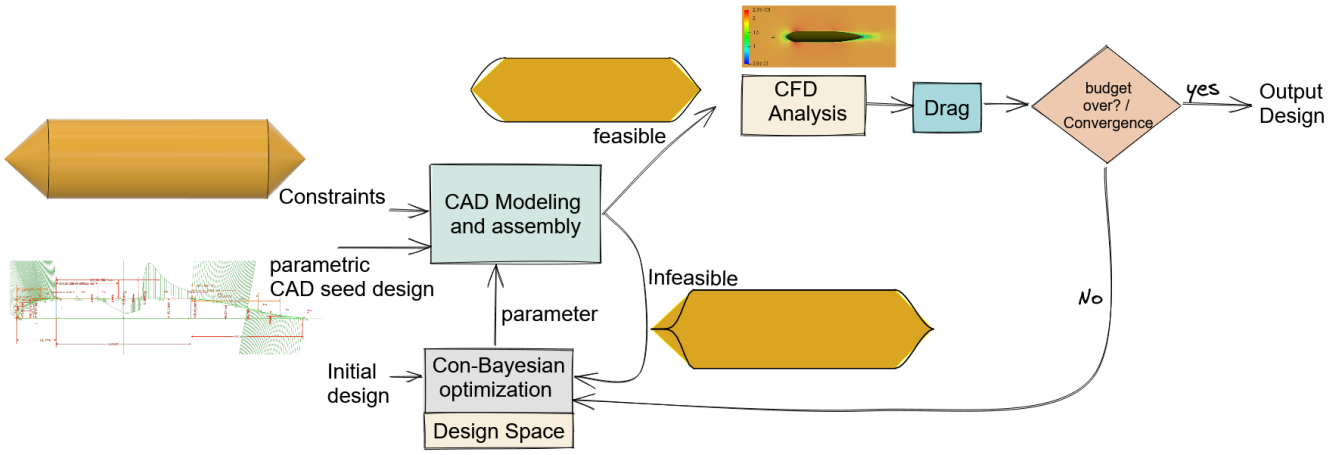


Figure 2: Optimization pipeline using integrated CAD and CFD tools with Bayesian Optimization

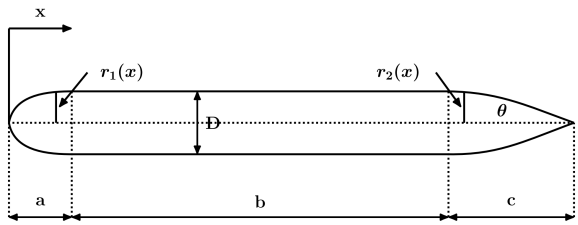


Figure 3: Myring hull: Geometry and parameters

Symbol	Minimum	Maximum
a	a_B	$a_B + 2500$ mm
c	c_B	$c_B + 2500$ mm
n	0.1	5.0
θ	0^o	50^o
$l = a + b + c$		

Table 1: Range of design parameters for optimization

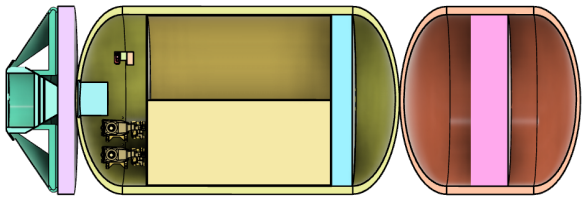


Figure 4: Selected components in the UUV in a specific packing configuration

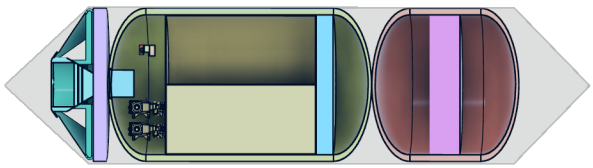


Figure 5: Selected components in baseline packed geometry

3 DESIGN EXPERIMENTATION AND RESULTS

In this section, we present two different experiments carried out using our optimization pipeline. In both cases, selected components are the same and consequently, the baseline packing geometry is identical. The operating conditions (i.e., the velocity of operation, initial and boundary conditions) and environmental conditions (e.g., turbulence intensity) are kept constant based on mission requirements. The baseline packing geometry is as shown in Figure 6. The design space (DS) of the search process is selected as shown in Table 1.

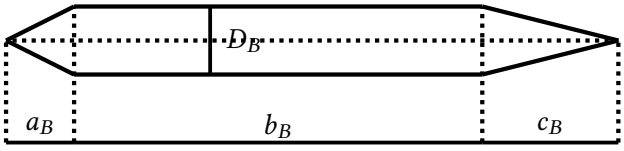


Figure 6: Baseline 3D hull design with $a_{baseline}$ = 555 cm, $b_{baseline}$ = 2664 cm, $c_{baseline}$ = 512 cm, $D_{baseline}$ = 1026 cm

3.1 Experiment 1

In this experiment, we only optimize the nose and tail shapes, defined by parameters n and θ. The range of the design space for optimization of parameters n and θ is given in Table 1. Due to it being a computationally costly process, we run 50 iterations of optimization using our optimization pipeline. The most optimal design (shown in Figure 8) has a drag value of approx 69 Newtons.

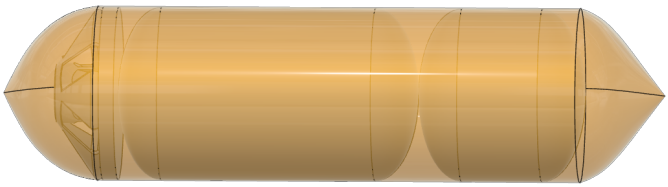


Figure 7: Optimal UUV hull shape with fixed nose and tail length. Optimal design parameters: $n = 1.0$; $\theta = 50.0$

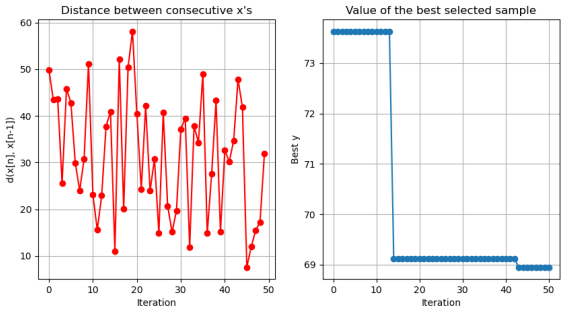


Figure 8: Optimization process vs number of evaluation/iteration: L2 distance between successive selected samples (left), Drag value of best-selected sample in Newton (right)

3.2 Experiment 2

In this experiment, we also optimized the nose and tail length (parameters a and b) in addition to their shapes (parameters n and θ). The design space for optimization of all four variables is given in Table 1. Again, we run 50 sequential optimization steps using our optimization pipeline. The most optimal design is shown in Figure 8 and had a drag value of approximately 36 Newtons. This is a 50% reduction in drag due to the streamlined nose and tail shapes and would save a large amount of energy consumption during real-world operation of the vehicle.



Figure 9: Optimal UUV hull shape with nose and tail length as a free parameter. Optimal design parameters: $a = 2643.86$; $c = 1348.72$; $n = 1.144$; $\theta = 22.03$

3.3 Analysis of result

In both experiments, the allocated budget was 50 evaluations since the evaluation time was tens of minutes. But BO converges to optimal/near-optimal design in a few iterations. In exp1 (refer to right side plot in figure 8) even in 12 iterations a near-optimal design was found and no significant further improvement is observed. In

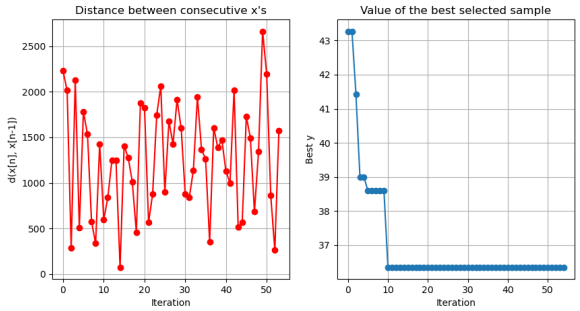


Figure 10: Optimization process vs number of evaluation/iteration: L2 distance between successive selected samples (left), Drag value of the best-selected sample in Newton (right)

exp2 (refer to right plot in figure 10) only in 10 iterations the optimal design was found and no further improvement was observed. This sample efficiency is due to the dynamic probabilistic modeling of the design space on labeled samples and state-of-the-art acquisition functions and accordingly costly optimization calculation. However, with the current multi-core implementation of BO, it takes milli-second to seconds for finding a new sample to evaluate and it is prudent to use BO in use cases where sample labeling and evaluation time can not be reduced beyond seconds.

4 RELATED WORK

One of the first well-known studies on optimizing UUV hull design for low drag was conducted by Gertler [10] in 1950. Later in 1976, Myring [17] studied viscous-inviscid flow interaction methods to predict drag and concluded that there is low variability in body drag force when the nose or tail varies from slender to stout within a specific range, but it increases dramatically outside that range. To design shapes for better performance, bio-inspired hull shapes for UUVs are becoming popular [5]. To this end, Dong et al. [5] designed a gliding robotic fish with the streamlined shape of a whale shark. Stewart et al. [22] designed a hybrid UAV-UUV system inspired by seabirds. A four-fin bio-inspired UUV was studied by Geder et al. [9]. They also showed that fish can achieve both high maneuverability and excellent gliding performance by equipping themselves with controllable fins and tails.

More recently, with extraordinary developments in computing capability and the maturity of mesh-based analysis tools, computational fluid dynamics (CFD) simulations are now widely applied to analyze UUV hydrodynamic performance. Most research is based on either the Reynolds-averaged Navier-Stokes (RANS) formulation or the large eddy simulation (LES). Since RANS treats viscous effects much better than potential flow theory and needs fewer computational resources, it is more frequently used than LES. With the advent of computer-aided design, traditional CFD can be leveraged inside an optimization loop to seek an optimal UUV design for given flow conditions; see, e.g., Alam et al. [2]. Many different optimization algorithms have been considered; for example, adjoint methods [12] and genetic algorithms [21]. [29] integrated

machine learning model with numerical simulation to get optimized design faster than the traditional methods. Schweyher et al. [20] used an evolutionary strategy- Genetic Algorithm to obtain a minimum-drag body. Application of Bayesian Optimization to find a minimum-drag shape was studied in 2D small arbitrary shapes by Eismann et al. [6] and an axisymmetric body of rotation by Vardhan et al [26]. Both works did not consider constraint modeling during the design process. A deep neural network-based approach is used to study the effects of UUV shape on drag force by [27].

5 CONCLUSION AND FUTURE WORKS

In this work, we developed an end-to-end design automation toolchain for constrained optimization problems for underwater vehicle hull design. We integrated the state-of-the-art AI-based optimization algorithm called Bayesian optimization along with the capability to handle constraints during the optimization process. Since this integrated tool is generic, the most interesting future work of interest is the extension and integration of other optimization algorithms with the current evaluation toolchain and comparing the performance of AI-based bayesian optimization with these optimization methods that are currently used in the domain of CFD-based optimization. To this end, it would be interesting to compare BO with other simulation-based optimization methods like GA, Nelder Mead, and Particle Swarm Optimization along with the most used gradient-based optimization method i.e. adjoint-assisted optimization. A detailed comparative study can give us a better picture of the standing of AI-based optimizers in comparison to other existing optimization methods.

ACKNOWLEDGMENTS

This work is supported by DARPA through contract number FA8750-20-C-0537. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the sponsor.

REFERENCES

[1] Pieter Abbeel, Adam Coates, and Andrew Y Ng. 2010. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research* 29, 13 (2010), 1608–1639.

[2] Khairul Alam, Tapabrata Ray, and Sreenatha G Anavatti. 2015. Design optimization of an unmanned underwater vehicle using low-and high-fidelity models. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47, 11 (2015), 2794–2808.

[3] Yannick Allard and Elisa Shahbazian. 2014. *Unmanned underwater vehicle (UUV) information study.* Technical Report. OODA Technologies Inc Montreal, Quebec Canada.

[4] Charles E Clark. 1961. The greatest of a finite set of random variables. *Operations Research* 9, 2 (1961), 145–162.

[5] Huijie Dong, Zhengxing Wu, Di Chen, Min Tan, and Junzhi Yu. 2020. Development of a whale-shark-inspired gliding robotic fish with high maneuverability. *IEEE/ASME Transactions on Mechatronics* 25, 6 (2020), 2824–2834.

[6] Stephan Eismann, Stefan Bartzsch, and Stefano Ermon. 2017. Shape optimization in laminar flow with a label-guided variational autoencoder. *arXiv preprint arXiv:1712.03599* (2017).

[7] Roger Fletcher. 2013. *Practical methods of optimization.* John Wiley & Sons.

[8] Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. 2014. Bayesian optimization with inequality constraints.. In *ICML*, Vol. 2014. 937–945.

[9] Jason D Geder, Ravi Ramamurti, Marius Pruessner, and John Palmisano. 2013. Maneuvering performance of a four-fin bio-inspired UUV. In *2013 OCEANS-San Diego*. IEEE, 1–7.

[10] Morton Gertler. 1950. *Resistance experiments on a systematic series of streamlined bodies of revolution: for application to the design of high-speed submarines.* Navy Department, David W. Taylor Model Basin.

[11] Christoph M Hoffmann and K-J Kim. 2001. Towards valid parametric CAD models. *Computer-Aided Design* 33, 1 (2001), 81–90.

[12] Antony Jameson. 2003. Aerodynamic shape optimization using the adjoint method. *Lectures at the Von Karman Institute, Brussels* (2003).

[13] Hrvoje Jasak, Aleksandar Jemcov, Zeljko Tukovic, et al. 2007. OpenFOAM: A C++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, Vol. 1000. IUC Dubrovnik Croatia, 1–20.

[14] Donald R Jones, Matthias Schonlau, and William J Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13, 4 (1998), 455–492.

[15] Miklos Maroti, Will Hedgecock, and Peter Volgyesi. [n. d.]. Rapid Design Space Exploration with Constraint Programming. ([n. d.]).

[16] Jonas Močkus. 1975. On Bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*. Springer, 400–404.

[17] DF Myring. 1976. A theoretical study of body drag in subcritical axisymmetric flow. *Aeronautical quarterly* 27, 3 (1976), 186–194.

[18] Carl Edward Rasmussen. 2003. Gaussian processes in machine learning. In *Summer school on machine learning*. Springer, 63–71.

[19] Juergen Riegel, Werner Mayer, and Yorik van Havre. 2016. FreeCAD.

[20] H Schweyher, Th Lutz, and S Wagner. 1996. An optimization tool for axisymmetric bodies of minimum drag. In *2nd international airship conference, Stuttgart/Friedrichshafen*. 3–4.

[21] Baowei Song, Qifeng Zhu, and Zhanyi Liu. 2010. Research on multi-objective optimization design of the uuv shape based on numerical simulation. In *International Conference in Swarm Intelligence*. Springer, 628–635.

[22] William Stewart, Warren Weisler, Marc MacLeod, Thomas Powers, Aaron Defreitas, Richard Gritter, Mark Anderson, Kara Peters, Ashok Gopalarathnam, and Matthew Bryant. 2018. Design and demonstration of a seabird-inspired fixed-wing hybrid UAV-UUV system. *Bioinspiration & biomimetics* 13, 5 (2018), 056013.

[23] Harsh Vardhan and Janos Sztipanovits. 2021. Rare event failure test case generation in Learning-Enabled-Controllers. In *2021 6th International Conference on Machine Learning Technologies*. 34–40.

[24] Harsh Vardhan and Janos Sztipanovits. 2022. Deep Learning based FEA Surrogate for Sub-Sea Pressure Vessel. In *2022 6th International Conference on Computer, Software and Modeling (ICCSM)*. IEEE, 36–39.

[25] Harsh Vardhan and Janos Sztipanovits. 2022. Reduced Robust Random Cut Forest for Out-Of-Distribution detection in machine learning models. *arXiv preprint arXiv:2206.09247* (2022).

[26] Harsh Vardhan and Janos Sztipanovits. 2023. Search for universal minimum drag resistance underwater vehicle hull using CFD. *arXiv preprint arXiv:2302.09441* (2023).

[27] Harsh Vardhan, Umesh Timalsina, Peter Volgyesi, and Janos Sztipanovits. 2022. Data efficient surrogate modeling for engineering design: Ensemble-free batch mode deep active learning for regression. *arXiv preprint arXiv:2211.10360* (2022).

[28] Harsh Vardhan, Peter Volgyesi, and Janos Sztipanovits. 2021. Machine learning assisted propeller design. In *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*. 227–228.

[29] Harsh Vardhan, Peter Volgyesi, and Janos Sztipanovits. 2023. Fusion of ML with numerical simulation for optimized propeller design. *arXiv preprint arXiv:2302.14740* (2023).

[30] AG Zhilinskas. 1975. Single-step Bayesian search method for an extremum of functions of a single variable. *Cybernetics* 11, 1 (1975), 160–166.