ISSN (Print) : 2319-8613
ISSN (Online) : 0975-4024

Aerodynamic Prediction of Round-Nosed Bullet by Artificial Intelligence

Yongho Lee

Mechanical Engineering Department, Embry-Riddle Aeronautical University
1 Aerospace Blvd., Daytona Beach, FL 32114, USA

lee3a4@erau.edu

***Abstract*—Drag prediction and minimization of bullets with a relatively simple nose shape are accomplished by developing and applying a machine learning model that is effective and convenient in transforming a function into another for a correlated system. The artificial intelligence model learns from the aerodynamic data of two-dimensional (2-D) sections, and transforms the knowledge into the data for the corresponding three-dimensional (3-D) objects, by projecting the data of 2-D sections upon a few anchoring data points of 3-D bodies. Optimal length-to-diameter ratio of the bullet is determined by finding the minimum drag coefficient at the conditions investigated. The implication of the result is that the present method can be applied to bullets with shaper noses and also extended to other engineering systems with more complicated geometries such as airplane wings and wind turbine rotor blades, for which abundant 2-D data are available in the literature.**

Keyword - artificial intelligence, machine learning, aerodynamics, drag, bullet

I. INTRODUCTION

With the advent of the deep-learning program that finally won a victory in 2016 over world's best human player of the time in Go game, also known as Baduk and Weiqi, the artificial intelligence stimulated interest from various areas. In recent years, researchers have applied AI to engineering field, and [1] reviewed such applications in fluid mechanics. A noticeable breakthrough was made by [2] in airfoil shape optimization by way of DRL (Deep Reinforcement Learning).

In the community of artificial intelligence, if the performance of a computer program improves or has a potential to improve with experience, the program is said to learn from experience. In this paper, a new approach based on ML (Machine Learning) algorithm is developed and demonstrated for aerodynamic drag prediction and shape optimization of bullets with a simple geometry. The new methodology presented here is beneficial when a sufficient amount of experimental data is available for two-dimensional (2-D) objects, inasmuch as it uses 2-D data for the analysis and design of three-dimensional (3-D) objects. Another example of the application is the design of 3-D wings for airplanes with use of the existing data of airfoils (2-D wings). Since abundant data of various airfoils have been available from the early 20th century, this new technique is expected to aid in the design of airplane wings that are the most important aerodynamic components of airplanes. Wind turbine rotor blades can be also designed with a similar approach, although they are substantially more complicated than bullets due to more geometric parameters involved.

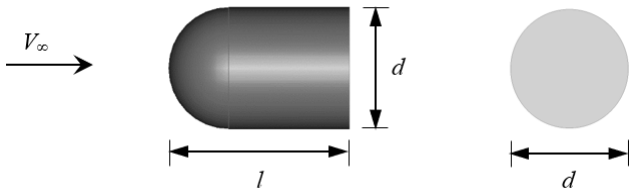


Fig. 1. Geometry and notations of a simple bullet considered in the present analysis; a side view and a cross-sectional view.

Figure 1 shows the geometry of the bullet with round nose, moving through air at the speed V_∞. The nose is a hemisphere with diameter d, and the total length of the bullet is denoted as l. In the present analysis, $l/d \geq 0.5$ is considered, with $l/d = 0.5$ equivalent to a hemisphere. According to Buckingham's Pi theorem (see, for example, [3]), the drag coefficient C_D of the bullet should depend on four nondimensional parameters, i.e., $C_D = f\,(l/d, Re, \varepsilon/l, Ma)$, where Re is Reynolds number, ε/l is the relative roughness of bullet surface, and Ma is Mach number:

$$C_D = \frac{D}{(1/2)\rho_\infty V_\infty^2 A} \qquad Re = \frac{\rho_\infty V_\infty d}{\mu_\infty} \qquad Ma = \frac{V_\infty}{a_\infty}.$$

Here, D is the dimensional drag force due to skin friction and pressure force acting on the bullet. The reference quantities, ρ_∞, μ_∞, and a_∞ are respectively air density, dynamic viscosity, and the speed of sound in the free stream far ahead of the object. The reference area A of the bullet may be chosen as the circular area in figure 1, $\pi d^2/4$. When the surface is very smooth as in many practical cases, the effect of ε/l can be neglected. For $Ma < 0.3$, the air flow around the object is essentially incompressible, the fluid density is nearly invariant, and as a result the effect of Ma becomes negligibly small, although Mach numbers of bullets readily exceed 1 during the motion. Therefore, we have $C_D = f(l/d, Re)$ for a smooth bullet at $Ma < 0.3$, and this paper considers such a low Mach number for demonstration purpose.

II. Numerical Simulations to Complement Training Data

Regarding the object in figure 1, [4] presented several C_D values for 2-D sections, but only one value for 3-D object with $l/d = 0.5$, i.e., a hemisphere. In order to obtain the minimum number of data points required by the present AI algorithm, Ansys® FLUENT 19.1 was used to simulate the air flows around the bullet by numerically solving the integral form of conservation of mass and momentum:

$$0 = \frac{\partial}{\partial t}\iiint_{\forall} \rho \, d\forall + \iint_S \rho \mathbf{v}_r \cdot d\mathbf{S} \tag{1}$$

$$\sum \mathbf{F} = \frac{\partial}{\partial t}\iiint_{\forall} \rho \mathbf{v} \, d\forall + \iint_S \rho \mathbf{v} \, \mathbf{v}_r \cdot d\mathbf{S}, \tag{2}$$

where $\mathbf{F}$ is the external force vector (including pressure, friction, and body forces in general), and $\mathbf{v}$ is the fluid velocity vector. Equations (1) and (2) are discretized in the flow domain with numerous small control volumes (i.e., mesh elements) approximating $d\forall$ and the differential surface area vector, $d\mathbf{S}$, normal to the control surface and pointing outward. The vector, $\mathbf{v}_r$, is the velocity of fluid relative to the control surface, merely equal to $\mathbf{v}$ if the control surface is stationary.

For turbulent flows occurring at high Reynolds numbers, additional equations for turbulence modeling are necessary. Although not shown here in detail, Spalart-Allmaras turbulence model was adopted, which has been extensively validated for aerodynamics problems. For sufficient accuracy in the simulation results, Spalart-Allmaras model in Ansys® FLUENT requires that the wall y^+ should be such that $y_w^+ \approx 1$ or $30 \le y_w^+ \le 200$. This y_w^+ is the inner-law variable defined near solid wall (the bullet surface in this paper),

$$y_w^+ = \frac{\rho y_1 u^*}{\mu}, \tag{3}$$

where y_1 is the height of the first mesh cell centroid adjacent to the solid surface and u^* is the wall-friction velocity, viz.,

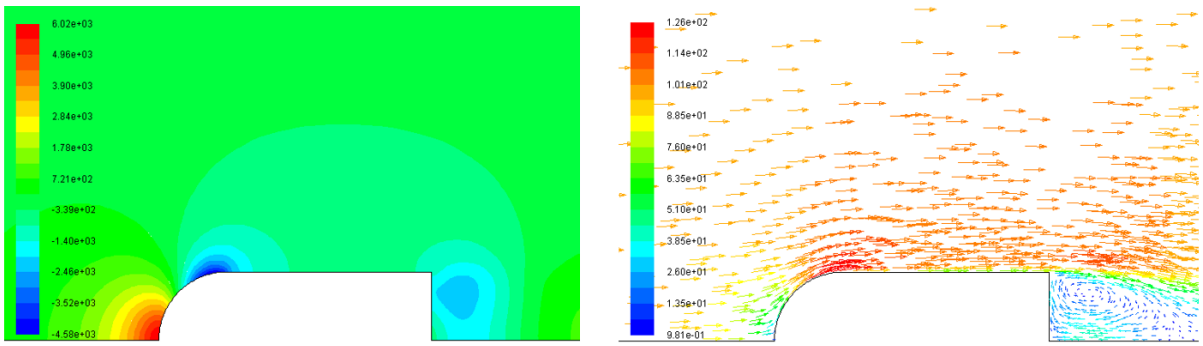
$$u^* = \sqrt{\frac{\tau_w}{\rho}} \tag{4}$$

with τ_w representing wall shear stress. Ansys® FLUENT uses an accurate wall function approach when $30 \le y_w^+ \le 200$, and hence in all the numerical simulations performed it was ensured that the area-averaged values of y_w^+ are in the appropriate range.

In the current analysis, V_∞ is set at 100 m/sec equivalent to $Ma = 0.294$ and $Re = 68{,}459$ so that the results are compatible with the data of [4] presented for incompressible flows ($Ma < 0.3$) at $Re > 10^4$, while the geometric parameter l/d is varied. The present analysis may be extended to $Ma > 0.3$ by including the dependence of C_D on both Ma and Re. However, since V_∞ is fixed, both Ma and Re are fixed herein, and the drag coefficient for smooth bullets is regarded as a function of l/d only.

Figure 2 shows a contour plot and a vector plot in the vicinity of the bullet with $l/d = 2.0$, moving at 100 m/sec. Owing to the symmetry about the centerline of the 3-D bullet in figure 1, only a portion of the object appearing in figure 2 is sufficient in the numerical simulations. Rotational motions of bullets are not considered herein, and thus the application of axis boundary condition along the centerline leads to the solution for the entire flow field around the 3-D bullet. No-slip condition was applied on the bullet surface, and velocity inlet

condition was used along the boundary of the computational domain far ahead of the bullet, albeit not shown in figure 2. Along the other boundaries of the domain, pressure outlet condition was applied with zero gage pressure. For each value of *l/d* simulated, the computational domain and the corresponding mesh were created. To ensure that the computational domain is sufficiently larger than the bullet, the dimension of the domain in each simulation is 20 times greater than the nose radius both in front of and in the side way of the bullet. Downstream of the bullet, the domain size is more than 20 times the bullet length.



(a) (b)

Fig. 2. Computational results of Ansys® FLUENT simulations for l/d = 2.0 at V_∞ = 100 m/sec: (a) static gage pressure in Pascal, (b) relative velocity vectors of air flow in m/sec.

In figure 2(a), the static gage pressure is illustrated in Pascal, where the stagnation point at the center of the nose has the highest value and the lowest pressure is observed near the conjunction between the nose and the cylinder because of flow acceleration. Figure 2(b) exhibits velocity vectors of air in the same portion of the computational domain with the color map legend presented in meters per second. The flow velocity in figure 2(b) is the air velocity relative to the moving bullet, and it is manifested that the flow accelerates around the nose due to the curvature and circulates behind the bullet due to the flow separation occurring at the base. The recirculation zone is associated with the low pressure region in figure 2(a).

The calculated data of C_D for 3-D bullets are plotted in figure 3, together with the data found in the literature. Five of 2-D data and one 3-D value at l/d =0.5 are from [4]. The other data for 2-D sections were taken from an unpublished reference cited by [4]. The experimental value of [4] for 3-D is C_D = 0.42 at l/d =0.5, and the computed value, C_D = 0.489, confirms the accuracy of the present numerical simulations, in comparison with the dispersion of the 2-D data. The 2-D values in figure 3 involve the reference area, $A = d \times$ width, whereas $A = \pi d^2/4$ for the 3-D drag coefficients. All the 2-D data in the figure are used as training data in the present analysis, but only the three simulation data for 3-D at l/d = 0.5, 3.0, and 6.0 are taken as part of the training data. The calculation of the drag force, D, exerted on the bullet involves a surface integral of F_x, which is the component of the forces in the direction parallel to V_∞:

$$D = \frac{1}{S}\iint_S F_x \, dS \,, \tag{5}$$

where $S = \iint_S dS$, equivalent to the summation of the areas of the facets (i.e., triangular and/or quadrilateral mesh elements) defining the bullet surface. Blunt bodies at small values of l/d have large drag caused by pressure difference between the nose and the base. As the body length increases, the pressure drag decreases by virtue of the elongation that enables the streamlining of the body, but the skin friction drag increases due to the enlarged surface area in contact with air. Therefore, the drag coefficient has a minimum at an intermediate value of l/d, although this trend is not apparent in the 2-D data because of the difference in geometry.

It is possible to produce more data for 3-D C_D by creating additional computational domains and meshes and by conducting more numerical simulations, or by manufacturing 3-D physical models to be used in multiple wind tunnel tests. However, the main objective of this paper is to develop and apply a new AI algorithm to save time and effort. Although a numerical simulation for a 3-D bullet does not require much computational resource owing to the simple axisymmetric geometry, fully converged numerical solutions of equations (1) and (2) covering a wide range of *Ma* and *Re* are still time-consuming, even if parallel processing is pursued.

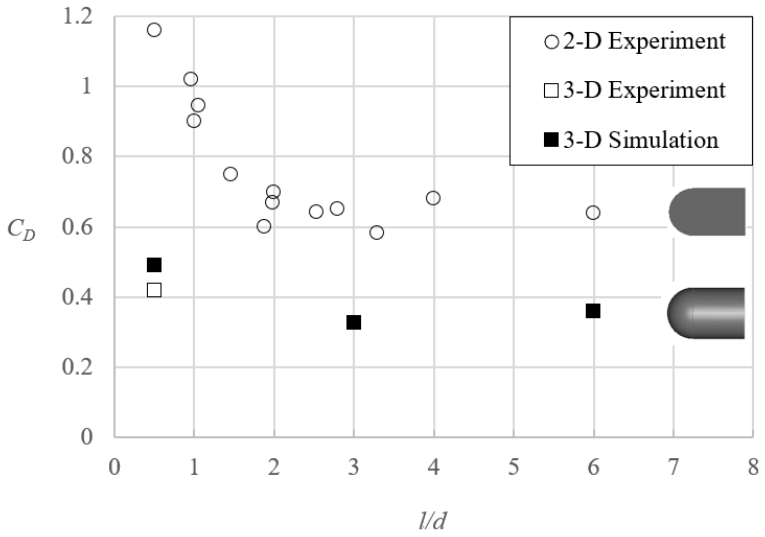


Fig. 3. Training data for drag coefficient in terms of $0.5 \leq l/d \leq 6.0$: experimental data from [4] and its unpublished reference, plotted together with Ansys® FLUENT simulation results. All data points are for incompressible flows ($Ma < 0.3$) at $Re > 10^4$.

III. ML-BASED AI ALGORITHM FOR AERODYNAMIC OPTIMIZATION

Conventional engineering practice in aerodynamic design is to work with a limited number of shapes or restrict the shapes with only a few degrees of freedom. The unique feature introduced in the present method is the use of 2-D data as a baseline in predicting 3-D data, which is more accurate and efficient than using a limited number of 3-D data, when the acquisition of additional data is costly and/or time-consuming. Through machine learning, the task can be achieved without explicit programming of aerodynamics analysis. This technique can save time and effort as well as increase the accuracy of prediction, when extended to the design of complex 3-D objects by using the data available for 2-D geometry. The flow chart of the algorithm developed in this paper is shown in figure 4, and Python™ was used to implement it.

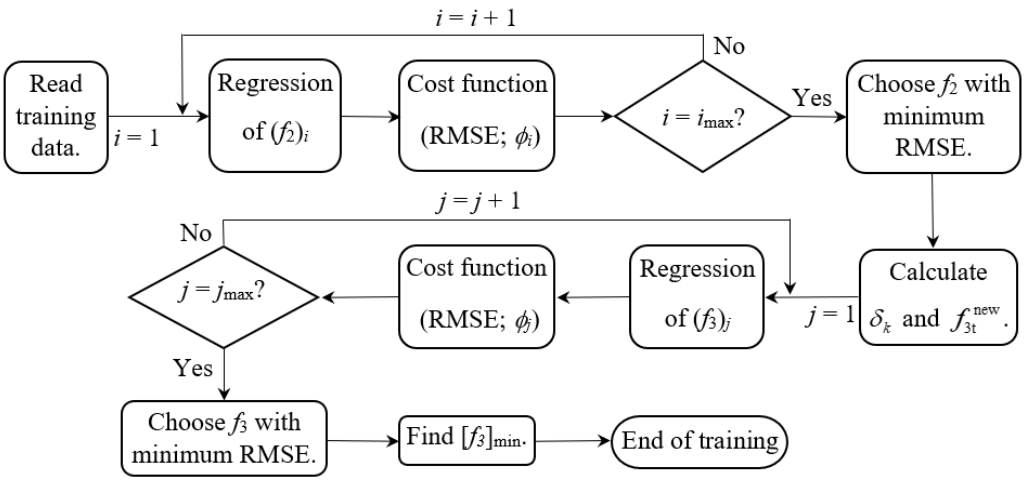


Fig. 4. Flow chart of the machine learning algorithm to construct a trained model for 3-D object from 2-D section training data.

The first step in the proposed algorithm is to search for a good regression model of the experimental training data, f_{2t}, for 2-D C_D shown by circles in figure 3. The regressions for these 2-D data are denoted as $(f_2)_i$ in the flow chart of figure 4, where the subscript i is the index for the iteration over all different regression types available. Since polynomial regression is sufficient for the current problem, i changes from 1 to i_{max} equal to the highest degree for polynomials that can be implemented. For typical machine learning algorithms, one may attempt to apply the regression directly to a few data for 3-D C_D, but with the limited number of data, the regression model does not represent the problem reasonably well. During the iteration, a cost function ϕ_i is calculated by using RMSE (Root Mean Squared Error):

$$\phi_i = \sqrt{\frac{1}{M}\sum_{k=1}^{M}(f_{2t,k} - F_{i,k})^2}\,, \tag{6}$$

where M is the number of data points available for 2-D C_D, $F_{i,k}$ is the value of $(f_2)_i$ evaluated at each value of $(l/d)_k$ in the training data set, and k is the index for each data point. The regression model $(f_2)_i$ with the minimum RMSE is selected as f_2. To allow an appropriate representation of the data somewhat sparse as in figure 3, it is necessary to impose $[(f_2)_i]_{max} \le [f_{2t}]_{max}$ and $[(f_2)_i]_{min} \ge [f_{2t}]_{min}$ in $(l/d)_1 \le l/d \le (l/d)_M$.

For the next step in the machine learning algorithm, another function, $\tilde{f}_2$, is defined as a linear function connecting two points, $(x_k, f_2(x_k))$ and $(x_{k+1}, f_2(x_{k+1}))$, with $x_k \equiv (l/d)_k$ introduced for simplicity in notation. The distance between x_k and x_{k+1} is now divided into three segments by two equispaced intermediate points, ζ_k and ξ_k,

$$\zeta_k = x_k + \frac{1}{3}(x_{k+1} - x_k) \qquad \xi_k = x_k + \frac{2}{3}(x_{k+1} - x_k)\,. \tag{7}$$

Let us define a normalized deflection, $\delta_k(X_k)$, which involves the difference between $\tilde{f}_2$ and f_2 at an intermediate point X_k (either ζ_k or ξ_k),

$$\delta_k(X_k) = \frac{\tilde{f}_2(X_k) - f_2(X_k)}{\left| f_2(x_k) - f_2(x_{k+1}) \right| + \left| \tilde{f}_2(X_k) - f_2(X_k) \right|}\,. \tag{8}$$

This is a reasonable approximation for the deflection at X_k, if the sign of $\tilde{f}_2(X_k) - f_2(X_k)$ in the numerator is fixed in $x_k < X_k < x_{k+1}$. Since $\delta_k(X_k)$ becomes indeterminate when the denominator is zero, equation (8) is replaced with $\delta_k(X_k) = 0$ in that case. In the instance where the sign of $\tilde{f}_2(X_k) - f_2(X_k)$ changes in $x_k < X_k < x_{k+1}$, it is necessary to modify $\delta_k(X_k)$ to

$$\delta_k(X_k) = \frac{\dfrac{1}{x_{k+1} - x_k}\int_{x_k}^{x_{k+1}} (\tilde{f}_2 - f_2)\,dx}{\left| f_2(x_k) - f_2(x_{k+1}) \right| + \left| \tilde{f}_2(X_k) - f_2(X_k) \right|}\,, \tag{9}$$

which is replaced by $\delta_k(X_k) = 0$ if the denominator is zero and thus the numerator is nearly zero as well. The integral in the numerator has the effect of reducing exaggerated oscillation of the curvature in f_2.

In order to replicate the curvature of f_2 approximately and reflect it in the model for 3-D C_D, let us use f_{3t} for 3-D C_D training data (the three solid squares in figure 3) and introduce a new data set, f_{3t}^{new}, of which elements are evaluated at all values of X_k:

$$f_{3t}^{new}(X_k) = \tilde{f}_{3t}(X_k) - \delta_k(X_k)\cdot\left\{ \left| f_{3t}(x_k) - f_{3t}(x_{k+1}) \right| + \left| \tilde{f}_{3t}(X_k) - f_{3t}^{temp}(X_k) \right| \right\}, \tag{10}$$

where $\tilde{f}_{3t}$ is a linear function passing through two points, $(x_k, f_{3t}(x_k))$ and $(x_{k+1}, f_{3t}(x_{k+1}))$, and f_{3t}^{temp} is a temporary regression of f_{3t} approximated by a polynomial function of degree N–1, with N denoting the number of 3-D training data in f_{3t}. It is not meaningful to take a polynomial of degree higher than N–1, particularly when only a few training data are used in the analysis. The index k in equations (7) through (10) is 1, 2, ⋯, N–1, namely, $x_1 = 0.5$ and $x_N = 6.0$ with $N = 3$ from figure 3. In the bullet problem investigated, δ_k is evaluated at four points, for the number of elements in f_{3t}^{temp} is $2\cdot(N–1) = 4$. To establish reasonable accuracy in the result, it is required that $N \ge 3$ spanned over most part of the independent variable domain. The training data in f_{3t} are used as anchoring points in the process.

The next part of the algorithm in figure 4 is analogous to that of the previous iteration loop, with $(f_2)_i$ and i replaced by $(f_3)_j$ and j. For the purpose of selecting the best regression from $(f_3)_j$, the cost function ϕ_j in this loop is defined as

$$\phi_j = \sqrt{\frac{1}{3N-2}\sum_{k=1}^{3N-2}(f_{3t,k}^{all} - G_{j,k})^2}\,, \tag{11}$$

where $f_{3t}^{all} = f_{3t} \cup f_{3t}^{new}$, and $G_{j,k}$ is the value of $(f_3)_j$ evaluated at each $x_k = (l/d)_k$. The number of elements in f_{3t}^{all} is $3N-2$, and the index k in equation (11) is for all x_k and X_k. After $(f_3)_j$ with the minimum ϕ_j is obtained and chosen as f_3, the model for 3-D C_D is approximately determined by

$$[C_D]_{3\text{-D}} = f_3 . \tag{12}$$

Since f_{3t}^{all} has only 7 elements in this demonstration, the polynomial of degree 6 is the best regression for f_3, insofar as the 7 points represent the 3-D C_D with reasonable accuracy. Thus not required in this problem, but generally it is needed to impose $[(f_3)_i]_{max} \le [\, f_{3t}^{all} \,]_{max}$ and $[(f_3)_i]_{min} \ge [\, f_{3t}^{all} \,]_{min}$ in the entire range of l/d.

TABLE I
TEST DATA FOR 3-D C_D FROM NUMERICAL SIMULATIONS AND THE PREDICTION OF MACHINE LEARNING ALGORITHM, COMPARED WITH THE VALUES THAT DO NOT UTILIZE 2-D DATA. THE VALUES IN THE PARENTHESES ARE THE ERRORS CALCULATED RELATIVE TO THE TEST DATA.

l/d	Test Data (3-D C_D)	Prediction of Eq. (12)	Direct regression of f_{3t} without using 2-D data
2.0	0.317	0.347 (9.46%)	0.372 (17.4%)
4.0	0.340	0.341 (0.29%)	0.311 (8.53%)

For verification, the model in equation (12) is now applied to the test data listed in table I. The two test data sets were obtained by additional computer simulations. The values in the fourth column of table I correspond to those of f_{3t}^{temp}, and the values in the parentheses are the errors calculated relative to the test data. It is evident that the prediction of equation (12) for 3-D C_D is significantly more accurate than that of f_{3t}^{temp} at both values of l/d tested. The model in equation (12) is highly accurate at $l/d = 4.0$.

Calculation of l/d leading to the minimum 3-D C_D, $[f_3]_{min}$, is straightforward once f_3 is obtained, and the process can be automated using conventional numerical analysis procedures. Figure 5 compares f_2 (long-dashed curve), f_{3t}^{temp} (short-dashed curve), f_3 (solid curve), and all the values from numerical simulations (rhombus symbols; both training data and test data). The minimum 3-D C_D is also marked for f_{3t}^{temp}, f_3, and all the data from simulations. The arrow marked with A indicates the approximate minimum C_D obtained by inspection of all five simulation data. The minimum C_D from equation (12) and f_{3t}^{temp} are marked respectively with B and C. The three values of minimum C_D for A, B, and C are 0.317, 0.328, and 0.310 at $l/d = 2.0$, 2.90, and 4.13. The errors in the minimum C_D are small for both B and C, but C shows a large amount of discrepancy in l/d for minimum drag coefficient, whereas a substantial improvement is made by equation (12). It appears that, with more 2-D experimental data at low values of l/d where the gradient of C_D is large, the long-dashed curve for f_2 may be deflected further downward so that the projected f_3 can be somewhat closer to the rhombus symbol at $l/d = 2.0$, which will further improve the accuracy of the predicted 3-D C_D.

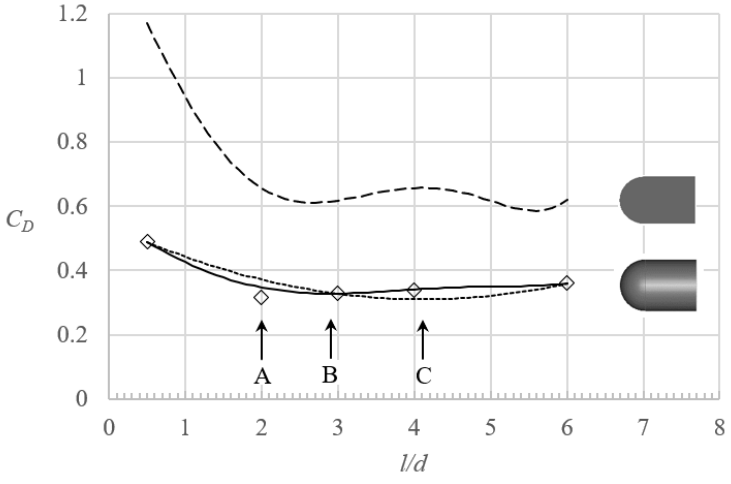


Fig. 5. Drag coefficients of 2-D and 3-D bullets for $Ma < 0.3$ and $Re > 10^4$: long-dashed curve is f_2, short-dashed curve is the polynomial function for f_{3t}^{temp}, and solid curve is f_3 in equation (12). Rhombus symbols are the collection of all the training and test data for 3-D case.

IV. CONCLUSION

The proposed machine-learning algorithm using available 2-D data as a basis to predict the information for 3-D objects significantly enhances the accuracy of predicted drag coefficients for bullets with round nose, when compared with the case where the algorithm is not adopted. The more data available for 2-D sections, the more accurate result can be accomplished for 3-D objects and their shape optimization. However, more work remains to cover a wide range of Mach numbers and Reynolds numbers, since the calculation was performed only at one Mach number with a fixed Reynolds number.

Bullets with sharper noses that are not investigated in this paper will show different results for the optimal ratio of length to diameter, but the same procedure can be applied for various nose shapes, provided that the drag coefficient data for 2-D objects are available. If the experimental data are not available, many 2-D numerical simulations and a few 3-D simulations can be performed to create such training data without making models and conducting many wind tunnel tests. More importantly, the AI-implemented technique appears to be applicable to many other problems in other fields as well as engineering.

REFERENCES

[1] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *Ann. Rev. of Fluid Mech*., vol. 52, pp. 477-508, 2020.
[2] J. Viquerat, J. Rabault, A. Kuhnle, H. Ghraieb, A. Larcher, and E. Hachem, "Direct shape optimization through deep reinforcement learning," *J. Comp. Phys*., vol. 428, 110080.
[3] B. R. Munson, D. F. Young, T. H. Okiishi, W. W. Huebsch, *Fundamentals of fluid mechanics*, 6th ed., John Wiley & Sons, Inc., 2009.
[4] R. D. Blevins, *Applied fluid dynamics handbook*, Van Nostrand Reinhold Co., Inc. 1984.

AUTHOR PROFILE

Yongho Lee is a professor in Mechanical Engineering Department at Embry-Riddle Aeronautical University, where he started his teaching career in 2002. He received his doctoral degree in Aeronautical & Astronautical Engineering from the University of Illinois at Urbana-Champaign. His areas of interest include CFD and multi-physics simulation, flow stability and turbulence, wind/water energy conversion, magnetic fluids application, and rocket propulsion.