SOFTWARE TEST REPORT

CAD TOOLS TEST REPORT

Submitted to:

SPAWAR Systems Center San Diego
53560 Hull Street, Code 2206
San Diego, California 92152-5001

Attention:
Mr. Mark Tukeman, Code 2712
(619) 553-1510

In fulfillment of the requirements for:

Contract No. N66001-02-D-0039
Development of Port Related Transportation Technologies
to Advance Military Responsiveness to National Needs

Task Order No. 0011
Computational Fluid Dynamic (CFD) Design Tool Development and Validation

CDRL Data Item No. A002BA

Security Classification: Unclassified

Prepared and Submitted by:

Center for the Commercial Deployment of Transportation Technologies
California State University, Long Beach Foundation
6300 State University Drive, Suite 220 • Long Beach, CA 90815 • 562.985.7394

Approved for public release: distribution is unlimited

September 30, 2003

Software Test Report (STR)

Computational Fluid Dynamics (CFD) Tool Development

Deliverable 3

CAD-Based CFD Tool development and Application

P/E Number: 2.20

Task Order Number: 11

POC

Hamid Hefazi, Professor & Chair

Mechanical and Aerospace Engineering Department

California State University, Long Beach

hefazi@csulb.edu

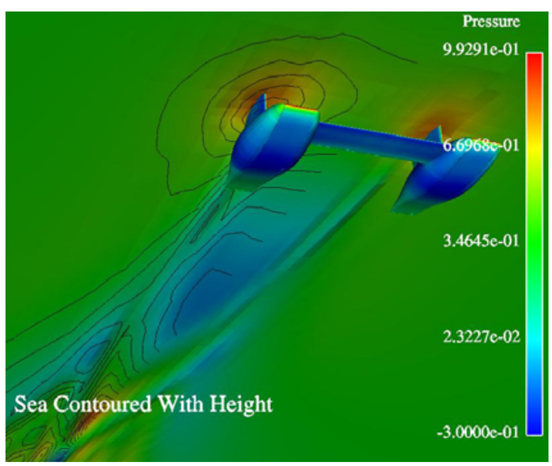

Table of Contents

This report presents the application of the optimization method based on Neural Networks described in the accompanying Computer Software Product End Items of Ref. [1]. Section 1 presents the design problem. Section 2 addresses the selection of design variables that will be optimized. Section 3 discusses the generation of the training and validation sets for the problem at hand as well as the outcome of the training. Optimization results are discussed in Sect. 4 where the outcome of the NN-based method is compared with that of the classical method.

1. Design problem description

The method is applied to the shape optimization of one of Pacific Marine/ Navatek, Ltd. lifting submerged configuration made of two displacement bodies, called *H-bodies*, linked with a thin foil, referred to as *cross-foil*, and attached to the ship by two struts (Fig. 1). The entire arrangement is referred to as the twin H-body configuration and is designed to be fitted to various mono-hull such as Deep-V, as well as multi-hull configurations such as catamaran or pentamaran . The displacement hulls are designed to provide good sea-keeping properties at lower speeds when the hulls of the catamaran or pentamaran are partially submerged, while the cross-foil is designed to provide additional lift at higher speed when the multiple hulls are lifted out of the water in order to reduce drag.

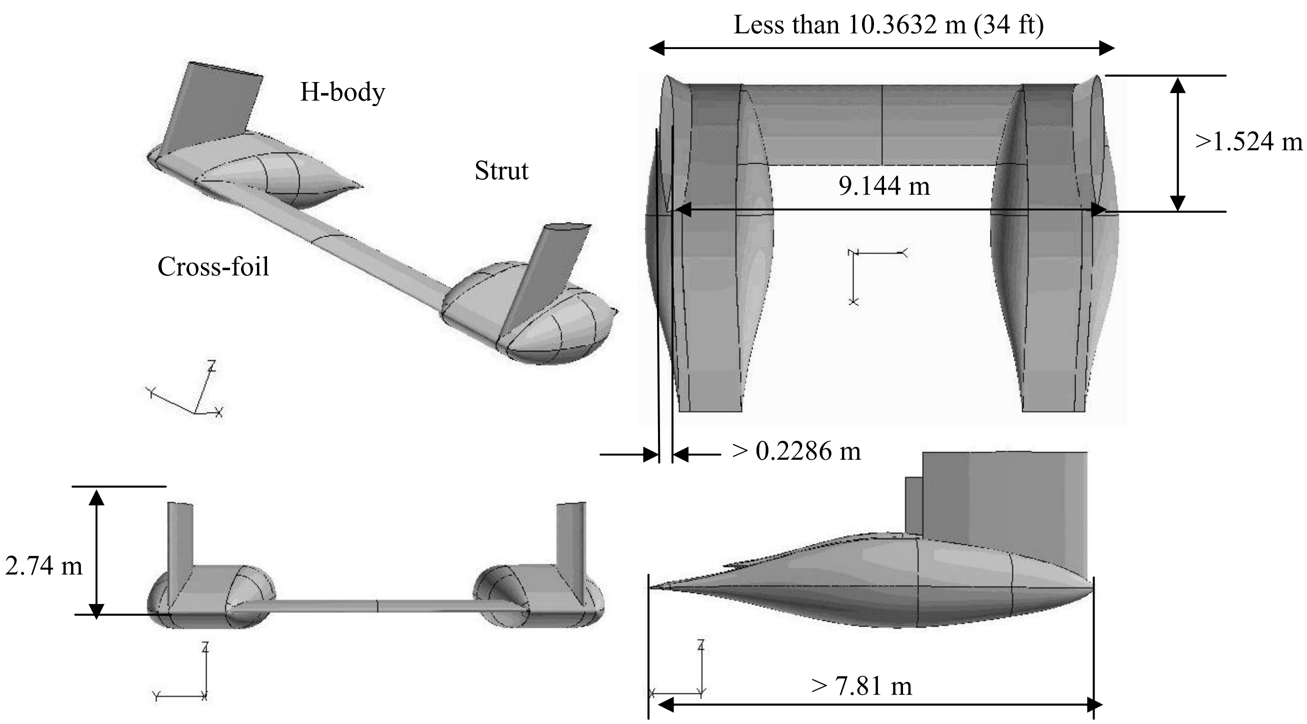

Fig. 1 Twin H-body configuration, Baseline

The objective of this optimization was to take a previously optimized configuration described in Reference [2] and to re-optimize it for a different speed and with additional constraints. The previous configuration (baseline design for this case) was optimized under CCDoTT work and is used in HDV-100 Technology Demonstrator. HDV-100 has a "Deep-V" hull form and is currently being constructed in Westerly Marine in Santa Ana, California. It is expected to go to sea in early 2004.

The optimization is performed for a boat speed of 47 knots (24.2 m/s). At this velocity, the boat is designed so that all hulls other than the twin H-body are out of the water. The whole configuration must be able to generate a total lift of 80 LT, including a minimum of 16 LT of displacement.

The objective is to maximize range, i.e. lift-to-drag ratio, which corresponds to minimizing drag at constant lift. In addition, the configuration is to be designed such that it can operate cavitation free at 52 knots. Additionally, the operating draft is to be 2.74 m (waterline to lowest point - 9 ft) and a structural constraint is to be imposed to prevent yield of the cross-foil. Also, the maximum width of configuration shall not exceed 10.36 m (34 ft) and the strut to strut distance is fixed by requirements for mating to upper hull.

From an optimization problem point of view, the objective function and design constraints can be summarized as:

- Objective: Maximize L/D
- Constraints:
  - Operational speed of 47 knots (i.e. Reynolds number is 211.63×10^6 based on a reference chord of 8.69 m in 75 F HI waters)
  - Cavitation free at 52 knots, i.e. $C_p > -0.269$ (HI water at 75 F)
  - Total lift equal to 80 LT
  - Displacement greater that 16 LT
  - 2.74 m (9 ft) operating draft
  - Strut centerline to strut centerline distance is fixed at 9.14 m (30 ft) for mating with upper hull
  - Overall beam length should not exceed 10.36 m (34 ft)
  - Minimum strut thickness 0.2286 m (0.75 ft)
  - Minimum strut chord of 1.524 m (5 ft)

- The cross foil should have the structural integrity not to yield using a material of yield strength of 344.7379 Mpa (50 ksi), assuming solid section.
- Displacement pod length should not exceed by more than 20% the parent body (defined as optimized configuration of Ref. 2), i.e. 7.81 m.

2. Hull parameterization and design variables

The hull shape parameterization is discussed first because it is required in the generation of the training and validation sets (TS & VS). It will also be used to perform the classical optimization which will be compared with the NN-based optimization below. The first step, as discussed in Ref. 1, is therefore to parametrically represent the configuration. These parameters, however, need not match the design variables, as is discussed below.

2.1. Hull parameterization

The hull is parameterized within Pro-Engineer using the same variables as those defined in Ref. 2 for the previous optimization. Table 1 presents the definition of these parameters.

Table 1. Parameters defining each component of Fig. 1

PARAMETER ***File*** *& Name*	***DESCRIPTION***
Hbody.ptr	
Hchord	Chord length of H-body (in meters)
Hwidth	Width of H-body (in meters)
Hthick	Thickness ratio (thickness/chord)
T0-T6	a_0 to a_6 for H-body (as defined in method description [1])
Camber_m	Maximum camber ratio (max_camber/chord)
Camber_p	Relative location of maximum camber (max_camber_loc/chord)
Hstrutpcspan	Relative spanwise location of strut on H-body (spanwise_strut_loc/chord)
Hstrutpcchord	Relative chordwise location of strut ¼ chord (chordwise_strut_loc/chord)
Hbodyspan	H-body spanwise location measured from strut centerline at ¼ chord from mid plane of model (in meters)
Hbodyangle	Angle of H-body. Angle can be positive or negative. A positive angle gives a nose down attitude to the H-body (in degrees)
Hbodydepth	Clearance of the H-body relative to the undisturbed sea surface (in meters)
Hcrosspcchord	Chord location of the ¼ chord of the cross-foil relative to the H-body chord
Hextra	Internal parameter related to the wakes (not changed)
Hbi	Ratio of inner ellipse side width relative to height
Hbo	Ratio of outboard ellipse side width relative to height

Crosswing.ptr	
Crossthick	Thickness–to-chord ratio of the foil
Crosschordi	Chord of inner (center line) foil (in meters)
Crosschordo	Chord of outboard foil (in meters)
Crosstwisti	Twist of foil at inboard position (in degrees). Angle can be positive or negative. Positive angle causes a nose down rotation of the airfoil.
Crosstwisto	Twist of foil at outboard position (in degrees). Angle can be positive or negative. Positive angle causes a nose down rotation of the foil.
U0-U6	a_0 to a_6 for cross-foil (as defined in method description [1])
Camber_m	Maximum camber ratio (max_camber/chord)
Camber_p	Relative location of maximum camber (max_camber_loc/chord)

Strut.ptr	
Strutchord	Chord length of strut (in meters)
s0-s6	a_0 to a_6 for strut (as defined in method description [1])
Camber_m0	Maximum camber ratio near H-body (max_camber/chord)
Camber_m1	Maximum camber ratio near sea surface(max_camber/chord)
Camber_p	Relative location of maximum camber (max_camber_loc/chord)
Strutthick	Thickness–to-chord ratio of the foil

2.2. Design variables

As described in Refs. 1 and 2, this hull parameterization is not necessarily the most appropriate when performing a particular optimization. For example, using parameters which have a direct relationship with the shape of the hydrofoil cross-section, such as thickness, is superior to the use of polynomial coefficients as design variables. One of the issues associated with the use of polynomials for thickness representation is the fact that a rather small disturbance in one or more of the coefficients may result in an unsuitable wavy profile. To avoid this problem during optimization, the foil cross-section is represented by “geometrical” parameters, which define, a priori, a suitable foil, and “compatibility relations” between these parameters and the polynomial coefficients is derived. The following parameters are the resulting design variables:

1. r_0: leading edge radius, related to a_0 by $a_0 = \sqrt{2r_0}$
2. a_1, which is a parameter related to the airfoil shape near the nose since
   $$a_1 = \left.\frac{d\left(y/\sqrt{x}\right)}{d\left(\sqrt{x}\right)}\right|_{x\to 0}$$
3. γ: slope at trailing edge with respect to horizontal, in deg.
4. x_m: location of maximum thickness, $0 < x_m < 1$
5. α: value of the term in $(1-x)^2$ at the trailing edge ($x = 1-\varepsilon$, where ε is small). This term represents the deviation from a linear thickness variation at the trailing edge and is given by

$$\alpha = \sum_{i=2}^{6} \frac{i(i-1)}{2} a_i - \frac{a_0}{8}$$

Two additional relations close the problem:

- $y_{th} = 0$ in $x = 1$
- $y_{th} = th/2$ at x_m. Here, the thickness ratio is controlled in the Pro-Engineer parameter file so that $th/2$ is set to 1.

Also, certain geometrical constraints may link two or more parameters and thus, are more effectively implemented by linking these parameters.

Table 2 presents the design variables used for the present problem and how they are related to the parameters of Table 1.

Table 2. Design variables and baseline values

Design variables	***Description** **(from Table 1 unless noted otherwise)***	***Min***	***Max***
Cw1	a_1 for cross-foil	1.905	2.390
Cwtwo	Crosstwisto	-1.085	-0.621
Cwthi	Crossthick	0.026	0.030
Cwordi	Crosschordi	1.435	1.627
Cwtwi	Crosstwisti	-0.423	-0.353
Hbang	Hbodyangle	-0.655	0.0784
Hbbi	Hbi	0.809	1.0238
Hbwid	Hwidth	0.820	1.426
Hbcwpcc	Hcrosspchord	0.076	0.0964
Strutchord	Strutchord	2.189	2.829
A	Used to calculate Hstrutpcspan and Hbo: Hbo=0.6096×(1-A)/(hbch×hbthi) Hstrutpcspan=A×0.6096/hbwid	0.459	0.572
Cwalpha	α for cross-foil cross-section	-5.073	-3.143
Hbalpha	α for H-body cross-section	-4.647	3.165
Cwcambm	Camber_m for cross-foil	0.0136	0.0142
Cwcambp	Camber_p for cross-foil	0.4456	0.4485
Hbcambm	Camber_m for H-body	0.6372	0.6645
Hbcambp	Camber_p for H-body	0	0.00175
Hbxm	x_m for H-body cross-section	0.4175	0.4512
Hbgamma	γ for H-body cross-section	46.48	49.90
Hbr0	r_0 for H-body cross-section	0.3455	0.3509
Cwxm	x_m for cross-foil cross-section	0.441	0.543
Cwgamma	γ for cross-foil cross-section	57.81	83.24
Cwr0	r_0 for cross-foil cross-section	0.510	0.872
Cwordo	Crosschordo	1.436	1.649
Hbthi	Hthick	0.0624	0.0738

Hbstpcc	Hstrutpcchord	0.0081	0.0109
Hbch	Hchord	6.314	7.810
Hb1	a_1 for H-body	4.090	5.173

3. Training/Validation Sets (TS/VS) and Neural Network training

3.1. Training and validation sets

For the present analysis, a validation set (VS) of 300 points was generated with the iSIGHT setup described in Ref. 1 and using the variables presented in Sect. 2, using a Latin Hypercube instead of an optimization algorithm. Three training sets (TS) were also generated using the same process, one with 1000 points, the second with 2000 points, and the third with 5000 points. Each data point takes approximately 6 to 7 minutes of computer time on an SGI Origin 3200 server.

For each size of Latin Hypercube sampling in the design space of Table 2, approximately 20 to 25% of the points were geometrically unfeasible (i.e. the model could not be successfully reconstructed) and had to be removed from the training sets. Nevertheless, the VS and TS will be referred to as 300-VS, 1000-TS, 2000-TS and 5000-TS subsequently.

3.2. Neural network training

Each neural network (NN) was trained with 28 inputs (or design variables) and one output, i.e. one for each function: Buoyant Lift (BL), Cpmin, Dynamic Lift (DL), L/D (LOD), Structural constraint (Struc), Total Lift (TL). In all cases, 7 candidate units were used along with the stopping criterion UP3. The UP3 criterion [3] consists in stopping training when the error on the VS has increased in three consecutive strips of length *k,* where $k = 5$. After every *k* hidden units (HU) have been added to the network, the error on the VS is compared with the previous one and training is stopped if this error has increased three consecutive times. Then, the “best” number of HU (less than when the training stopped) is chosen for the NN which leads to the minimum validation error. This stopping criterion was determined in Ref. 4 to be very accurate as it was shown in the study that the absolute minimum validation error was found each time on the 2D model problem. For each function, 10 NNs were built and the one that had the best validation error was chosen. Typically, training each network takes from about an hour (for 1000-TS) to a day for the more complete data sets (5000 TS). Unlike CFD methods which are demanding in computing power (I/O, memory), training demands relatively little other than CPU time and thus, all training can be done in parallel on the same server without interfering with other on-going computations. This feature is yet another advantage compared to training a single multiple-output network.

Table 3 presents the results of the training. Total lift (TL) has been included although it is not used. Instead, in the optimization method, it is calculated by adding buoyant and dynamics lifts as determined by their respective network. When the number of TS points increases, the squared Error, E/Pmax, decreases on both VS and TS, as would be expected. Also when the size of the TS increases, the number of HU found usually increases. This also makes sense since the number of points is relatively small compared to the complexity of the function over the design space. At some point, when there are enough points to describe accurately the function, one should expect the number of HUs to stop increasing. It should be noted, however, that the purpose of the current approach is not to reach this "best" number of HUs since this would require a very large number of points in the TS thus rendering useless the use of NN. Instead, the objective is to assess whether with a limited number of points in the TS, one is able to generate an improved design when compared to the classical CFD approach.

As for the maximum error, Emax, it is the maximum deviation between the targets and the outputs from the network over the entire TS or VS. Since the targets and outputs have been non-dimensionalized by the average target over the TS, this value gives an idea of the percentage of *maximum* error which can be expected with the NN replacing the CFD. This maximum error, although quite large for certain variables such as Cpmin, will be in general much smaller than that, and in particular, is *likely* to be much better at or near the optimum point.

The largest max errors are found on Cpmin (up to 34% for TS 1000) and the structural constraint (15%). This is partly due to the fact that Struc and Cpmin are both C^0 functions and that the network is built to create C^∞ functions. One can also notice that in general there is a significant improvement in maximum error on the TS and VS between 1000 and 2000, but that for 5000, there is no improvement especially on the TS. The reason for this behavior is unclear and is being investigated.

Table 3. Output from NN training. Note that errors are normalized

			Error on TS		Error on VS	
Function	TS	#HU	(E/Pmax)tr	(EMax)tr	(E/p)val	(EMax)val
BL	1000	6	1.044E-05	1.876E-02	2.025E-05	2.831E-02
	2000	15	5.959E-06	2.290E-02	1.351E-05	1.590E-02
	5000	17	4.011E-06	2.538E-02	4.548E-06	1.089E-02
Cpmin	1000	3	5.334E-04	3.369E-01	6.584E-04	2.023E-01
	2000	26	1.138E-04	8.772E-02	5.123E-04	1.604E-01
	5000	32	1.663E-04	2.754E-01	4.230E-04	1.497E-01
DL	1000	3	3.174E-05	9.733E-02	2.611E-05	2.215E-02
	2000	2	2.155E-05	7.087E-02	2.082E-05	2.059E-02
	5000	7	2.185E-05	1.115E-01	1.933E-05	2.306E-02
LOD	1000	2	2.758E-04	9.457E-02	3.672E-04	8.725E-02
	2000	5	2.154E-04	7.931E-02	2.957E-04	6.631E-02
	5000	13	1.605E-04	1.053E-01	2.340E-04	6.232E-02
Struc	1000	5	2.415E-04	1.398E-01	4.370E-04	1.521E-01
	2000	10	2.124E-04	1.369E-01	3.579E-04	9.857E-02
	5000	54	1.853E-05	2.446E-02	1.350E-04	1.234E-01
TL	1000	2	2.471E-05	8.064E-02	1.879E-05	2.349E-02
(not used)	2000	2	1.566E-05	5.904E-02	1.574E-05	2.126E-02
	5000	9	1.421E-05	9.617E-02	1.403E-05	1.939E-02

4. Optimization results: classical vs. NN-based

This section discusses the results of the application of the optimization methods to the problem introduced thus far. Since the generation of TS and VS (see Sect. 3.1) requires setting up the automatic analysis controlled via iSIGHT, it was possible to easily run the classical optimization approach. The latter is discussed in Sect. 4.1. Section 4.2 presents the results obtained with the NN-based optimization method. The two are compared in Sect. 4.3.

4.1. Classical approach

The optimization process is that of Ref. 2 using the design variables of Sect. 2. *iSIGHT* controls the process and calls the scripts, which, ***from the design variables***,

1. Defines the geometry (Pro-Engineer),
2. Check for structural constraint violation (from out of Pro-Engineer)
3. Generates a suitable mesh (ICEM-CFD),
4. Executes the CFD method (IBL-Neg.), and
5. Extracts the data needed by *iSIGHT* (***objective function and constraint values***) for the next iteration.

This process includes modifications to take into account the structural constraint as well as define the constrained objective function, f_c, which is being minimized by iSIGHT using Genetic Algorithm. This process is illustrated in Fig. 2 and is the application of the description of the method presented in Ref. 1 to the problem of Sect. 1.

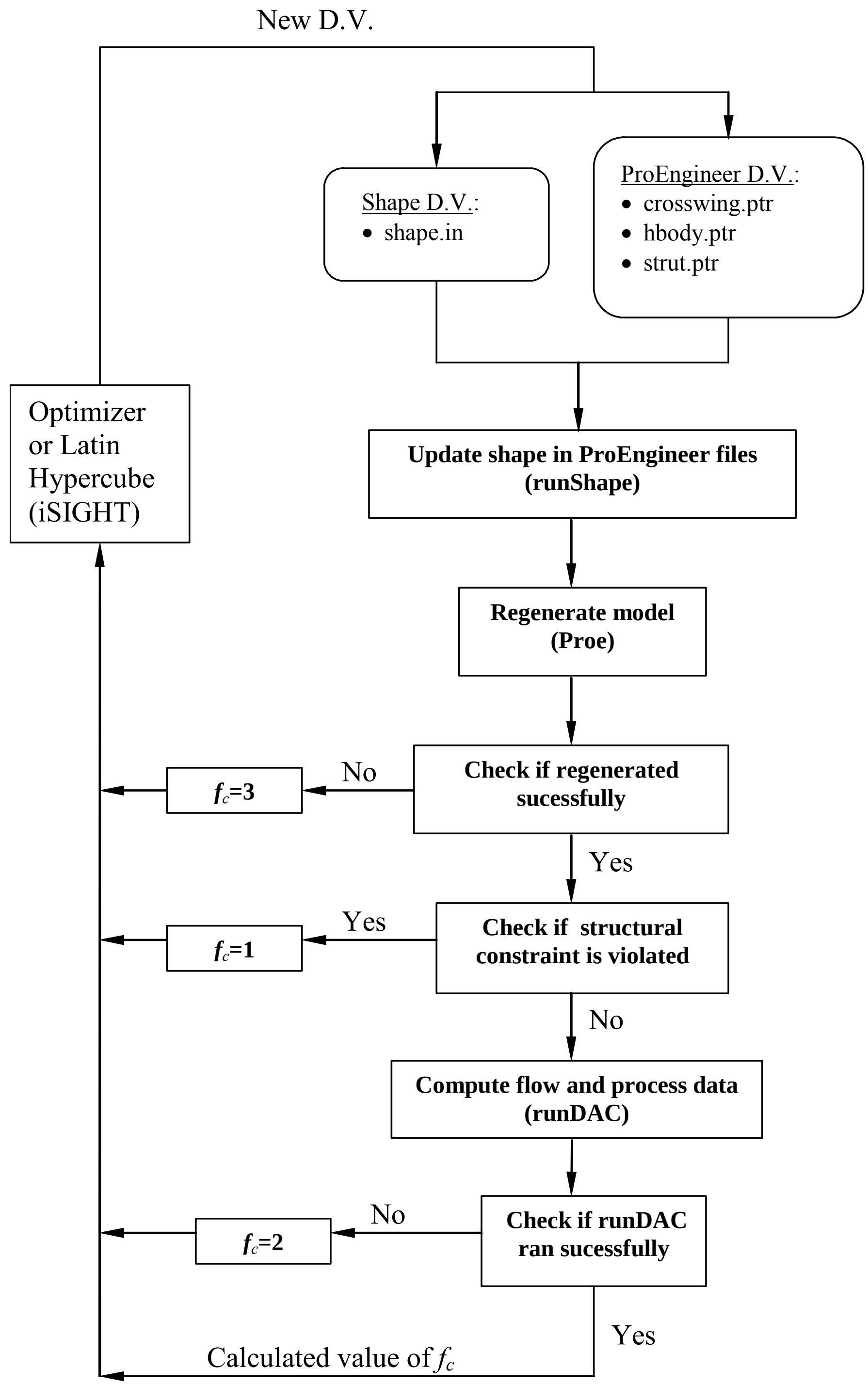

Fig. 2 Optimization with classical approach or training/validation set generation process

The optimized configuration of Ref. 2 was chosen as initial condition. As noted above, however, the operating conditions and constraints are quite different for the present application. Therefore, the penalized objective function would have been very close to f_{max}. In addition, the optimization algorithm used here is based on Genetic Algorithm, and as such, has difficulties with highly constrained problems. In order to speed up the optimization and start near a point where higher quality populations could be rapidly found, the configuration was modified "by hand", i.e. by varying parameters one at a time and seeing how the constraints behave, to obtain a suitable geometry. It should be noted that, with the current implementation within iSIGHT, this task is simple since it only entails modifying parameters in *ptr* files. Also, the arrangement of the twin H-body configuration lends itself to this "manual" iterative process relatively easily because most variables, other than actual shape of the body cross-sections, play a relatively independent role on the functions of interest. For example, increasing the size (chord and width) of the H-body increases displacement but only has a secondary effect on dynamic lift. The reverse is true when the chord of the cross-foil is increased: the dynamic lift is increased but the displacement changes very little. Finally, the minimum pressure constraint (related to cavitation at higher speed) was met by altering the thickness of the profiles. The resulting configuration was termed "baseline" twin H-body configuration for this optimization. It has a lift-to-drag ratio (L/D) of 10.23 as computed by the CFD method.

A Genetic Algorithm optimization was run over the design space of Table 2 for 5000 iterations. The outcome was a configuration with an L/D of 12.92, which represents a 26% improvement in L/D. Table 4 presents the parameters defining the baseline and optimized configurations. Also, a comparison of the performance of the baseline and optimum configuration is shown in Table 5. All other constraints are met automatically in the model implementation.

Table 4. Optimized vs. baseline configuration definition

PARAMETER ***File*** *& Name*	***Baseline***	***Optimized***
Hbody.ptr		
Hchord	7.810000	7.5380230380
Hwidth	1.108128	1.0671455790
Hthick	0.063	0.0653296870
T0-T6	T0= 0.86788 T1=4.60988200 T2= -25.24839 T3= 84.14509 T4=-151.06258 T5= 124.43371 T6= -37.74559	T0= 0.83555 T1=4.87769328 T2= -28.93189 T3= 102.66134 T4=-190.11619 T5= 160.07984 T6= -49.40634
Camber_m	0.006589	0.0004957190
Camber_p	0.498800	0.6462950000
Hstrutpcspan	0.275058	0.2858145796
Hstrutpcchord	0.010000	0.0099889960
Hbodyspan	3.657600	4.57200
Hbodyangle	-0.054293	-0.3934063250
Hbodydepth	2.192044	2.1920440000
Hcrosspcchord	0.07340240	0.0836762470
Hextra	0.001	0.001
Hbi	1.000000	0.8861223030
Hbo	0.619474	0.6185198493
Crosswing.ptr		
Crossthick	0.030	0.0273352010
Crosschordi	1.6	1.5631906610
Crosschordo	1.6	1.5069782420
Crosstwisti	-0.00179	-0.3999915520
Crosstwisto	-1.316	-0.7631733630
U0-U6	U0= 1.26023 U1=2.28394000 U2= -7.10073 U3= 12.73618 U4= -21.85686 U5= 18.93208 U6= -6.25484	U0= 1.08185 U1=2.06661627 U2= -16.02129 U3= 74.44746 U4=-155.40889 U5= 137.25177 U6= -43.41752
Camber_m	0.020	0.0140000480
Camber_p	0.447090	0.4466030030
Strut.ptr		
Strutchord	2.8	2.3999711610

s0-s6	s0= 2.25919 s1=0.18633100 s2= -9.53275 s3= 29.44936 s4= -43.21236 s5= 24.96507 s6= -4.11484	s0= 2.25919 s1=0.18633100 s2= -9.53271 s3= 29.44913 s4= -43.21194 s5= 24.96473 s6= -4.11474
Camber_m0	0.0	0.0
Camber_m1	0.0	0.0
Camber_p	0.3	0.3
Strutthick	0.041	0.0476255723

Table 5. Performance of optimized vs. baseline configuration

	Baseline	**Optimum**	Objective
Displacement	18.4 LT	**16.1 LT**	> 16 LT
Total lift	80.2 LT	**81.6 LT**	= 80 LT
$C_{p,\min}$	-0.263	**-0.250**	> - 0.269
L/D	10.23	**12.92**	Maximum

4.2. NN approach

As discussed in Section 3.2, the approach specific to the twin H-body optimization problem of Sect. 1 calls for the use of 5 single output neural networks as shown in Fig. 3, one for the objective function and the others for the constraints: lift-to-drag ratio (LOD), minimum pressure (Cpmin), dynamic lift (DL), buoyant lift (BL) and maximum stress value (struc). Alternatively, a single NN with 5 outputs could have been used, but typically, the resulting network is much more complex than 5 individual networks and, hence, takes more time to generate, i.e. train, than 5 networks.

Also, although the optimization is performed on the constrained objective function, f_c, it was not represented directly by a single NN because it would have required a very good training set definition in the –small– regions of the design space where the design was feasible. Instead, using 5 networks allows for accurate representation of each function over the design space and thus improved predictions for f_c.

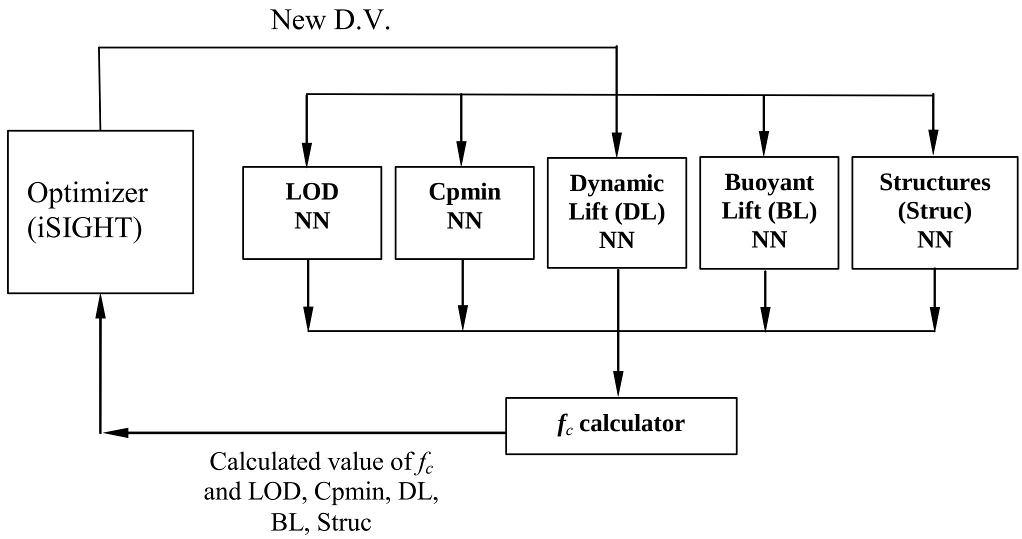

Fig. 3 Optimization process with NN approach

The neural networks generated in the form of 5 executables for Cpmin, LOD, DL, BL and Struc using the different sizes of training sets (1000, 2000 and 5000) were integrated in iSHIGHT following the standard iSIGHT process. A Genetic Algorithm (GA) optimization was used with an initial population of 50 and 35000 iterations were run based on the constrained objective function.

The best 100 runs resulting from the optimization with the five NNs (best f_c) were then run using the Pro-Engineer model and the CFD code to compute the objective function and constraints and compare them with those of the NN near the optimum. Also, a few results from the optimization (i.e. as determined by the NN) with a slightly higher LOD than that of the best f_c but with constraints closer to their limit (therefore resulting in a lower f_c) were chosen and also run through the CFD package. A summary of the results is shown in Table 6. For each size of training set, the table shows:

- The best f_c as determined by the optimizer (i.e. after 35000 GA iterations using the NN)
- The best LOD based on the NN with minimal constraint violations: corresponds to some hand-picked results from the optimization showing a slightly better LOD but with constraints close to the acceptable limits resulting in a higher f_c
- The best f_c based on CFD results from the 100 best NN points (as determined by the GA)
- The best LOD based on CFD results from the 100 best NN points (as determined by the GA)

In each case, buoyant lift (BL), total lift (TL), lift-to-drag ratio (LOD) and minimum pressure (Cpmin) values computed by the NN and the CFD method are shown. The stress value (Struc) is not shown because it exhibited little variations between points and because the constraint was not violated. Also, the dynamic lift (DL) can be directly calculated from total and buoyant lift values (the optimization actually uses DL and BL to determine TL, but the latter is presented because it corresponds to a primitive twin H-body requirement). It should also be noted that the constraints are not implemented as step functions but rather very steep functions which do vary near the constraint border. For example, the primitive requirement calls for a displacement or buoyant lift greater than 16, but as implemented here, a value close to 15 is acceptable. For this reason, f_c may vary in a counter-intuitive fashion rendering the analysis of its values difficult. It is therefore not shown.

Table 6. Results from NN optimization and comparison with CFD

		Output from NN				Same DV but with CFD			
TS	Results	BL	TL	LOD	Cpmin	BL	TL	LOD	CpMin
1000	Best f_c optimized from NN	15.23	80.59	14.39	-0.265	15.24	81.59	13.76	-0.266
	Best LOD optimized from NN	14.75	79.94	14.44	-0.267	14.78	81.04	13.85	-0.268
	Best f_c (using CFD) out of 100 best runs	15.49	80.81	14.30	-0.264	15.51	81.67	13.70	-0.266
	Best LOD (using CFD) out of 100 best	15.25	80.60	14.38	-0.265	15.25	81.64	13.78	-0.265
2000	Best f_c optimized from NN	15.20	80.67	14.49	-0.265	15.13	80.86	13.70	-0.271
	Best LOD Optimized from NN	15.07	80.66	14.51	-0.265	15.01	80.78	13.81	-0.269
	Best f_c (using CFD) out of 100 best runs	15.30	80.74	14.42	-0.263	15.23	80.96	13.71	-0.265
	Best LOD (using CFD) out of 100 best	15.17	80.59	14.47	-0.265	15.11	80.75	13.79	-0.270
5000	Best f_c optimized from NN	15.17	80.64	14.43	-0.265	15.08	80.71	13.65	-0.275
	Best LOD Optimized from NN	15.05	80.90	14.48	-0.264	14.98	80.97	13.69	-0.274
	Best f_c (using CFD) out of 100 best runs	15.18	80.19	14.09	-0.265	15.18	80.14	13.59	-0.274
	Best LOD (using CFD) out of 100 best	15.17	80.64	14.43	-0.266	15.08	80.71	13.68	-0.276

The differences between the NN and the CFD are within acceptable limits; they do not exceed the Emax shown in the previous section and are in general much less. In particular, they are very close in many instances. Also, differences between the values resulting from different selection processes for a particular TS are rather small. Finally, while one observes that the LOD is over-predicted by the NN, the differences between points are about the same for a given TS.

Unfortunately for 5000 TS, the Cpmin constraint is violated (<-0.269) for the 100 "best" runs, resulting in only near-feasible designs. Those would then require to be "tweaked" by hand to make sure they respect all constraints. Also, the optimum LOD determined by this TS (around 13.65 based on CFD) is lower that found with the 1000 and 2000 TS (higher than 13.7 for both). Regardless of which TS is used, however, LOD is greatly improved from the 12.9 result obtained with the direct CFD method.

4.3. Comparison between classical and NN-based method

All results for all TS show LOD ranging from 13.59 to 13.85, which is a definite improvement from the classical optimization. This improvement is due to the ability to increase the exploration of the design space within the time available in a given design project. Because CPU time when using direct CFD is going to limit the design space exploration (5000 GA iterations require approximately one month of constant calculations here), a true optimum cannot be reached. **With as low as 1000 points generated to approximate the various functions over a design space with 28 design variables, improvements of about 34% in LOD compared with the original baseline twin H-body can be reached at one fifth the cost needed to get a 26% improvement!**

These results also point to a few improvements which would need to be implemented related to function differentiability (to improve Cpmin predictions, for example), to which constrained objective function to use, to the "best" size of the training set, and to the choice of optimization method. The latter comment is particularly pertinent for problems in which, like it might be the case for the results obtained with the 5000 TS, the optimizer (GA here) has focused its attention in a region of the design space where one (or more) function (objective or constraint) is not well approximated (Cpmin in this case). One could benefit from having an optimization method with explores several regions of the design space, as illustrated in Ref. 5.

Finally, more sample test cases would need to be investigated to confirm the findings presented here. Results do show, however, that the method could provide the design engineer with a valuable tool for improving designs within a limited time frame and possibly at a lower cost than using conventional analysis tools.

5. Summary/Conclusion

The CAD-Based CFD and Design Optimization Tool developed and documented in the accompanying Computer Software Product End Item report has been validated by applying it to an under water lifting body hull form. This novel method replaces classic optimization approach with a Neural Network based optimization. The application shows that better optimization can be achieved with substantially less computer resources.

6. Glossary

Symbols

$C_p = \frac{p - p_{atm}}{q_\infty} = 1 - \left(\frac{V}{V_\infty}\right)^2 - \frac{2}{F_r^{\;2}}\left(\frac{z}{L}\right)$: Pressure coefficient

p_{atm}	Atmospheric pressure
q_∞	Free stream dynamic pressure.
Cpmin	Minimum pressure Coefficient

Acronyms

LT	Long Tons
L/D	Lift to Drag Ratio
Mpa	Mega Pascal
TS	Training Set
VS	Validation Set
NN	Neural Network
BL	Buoyant Lift
DL	Dynamic Lift
TL	Total Lift
HU	Hidden Units
IBL	Interactive Boundary Layer
GA	Genetic Algorithm
LOD	Lift over Drag

7. Acknowledgements

The CCDoTT project is a collaborative effort involving several faculty and students at California State University, Long Beach. Staffs primarily responsible for this portion of the work are Dr. Eric Besnard, Associate Professor and Adeline Schmitz, Research Associate and Stanly Baski graduate student in the Mechanical and Aerospace Engineering Department.

8. References

1. Hamid Hefazi *et al.*, " Computer Software Product End Items, Deliverable 3 CAD-Based CFD Tool development and Application DI-MCCR-80700" Report, Center for the Commercial Deployment of Transportation Technologies, Long Beach, CA, Sept. 2003.
2. Hamid Hefazi et al, "CFD Design Tool Development and Validation, CCDoTT FY00 Task 2.8", Center for the Commercial Deployment of Transportation Technologies, Long Beach, CA, Feb. 2002.
3. Hamid Hefazi *et al.*, "Computer Software Product End Items, Deliverable 2 Optimization Tool Development, DI-MCCR-80700", Center for the Commercial Deployment of Transportation Technologies, Long Beach, CA, Sept. 2003.
4. Hamid Hefazi *et al.*, "Software Test report, Deliverable 2 Optimization Tool Development, DI-IPSC-81440A" , Center for the Commercial Deployment of Transportation Technologies, Long Beach, CA, Sept. 2003.
5. C.Y. Lin and W.H. Wu, "Niche Identification Techniques In Multimodal Genetic Search With Sharing Scheme," *Advances in Engineering Software 22*, pp. 779-791, 2002.