A Pareto Frontier for Full Stern Submarines via Genetic Algorithm

by

Mark W. Thomas

B.S. Electrical Engineering, Oklahoma State University, 1984
S.M. Electrical Engineering and Computer Science,
Massachusetts Institute of Technology, 1996
Naval Engineer, Massachusetts Institute of Technology, 1996

Submitted to the Department of Ocean Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Hydrodynamics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1998

Author .....
Department of Ocean Engineering
May 12, 1998

Certified by .....
Justin E. Kerwin
Professor of Naval Architecture
Thesis Supervisor

Accepted by .....
Kim Vandiver
Chairman, Committee on Graduate Students

A Pareto Frontier for Full Stern Submarines via Genetic Algorithm

by

Mark W. Thomas

Submitted to the Department of Ocean Engineering
on May 12, 1998, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Hydrodynamics

Abstract

An exploratory design and analysis code for underwater vehicles with ducted, multi-stage propulsion systems is developed from an existing version. Boundary layer modeling is added and used to predict flow separation as well as to estimate the effect of wake fraction on propulsive efficiency. Force balance (i.e., convergence to a self-propelled condition) is achieved by automatic variation of advance coefficient. The force and boundary layer calculations of the revised code are validated using published experimental data.

A slightly modified version of the code is then used as the evaluator for an original Pareto genetic algorithm. The algorithm seeks the code's Pareto (non-dominated) frontier in terms of usable hull volume and propulsive efficiency, with the intention of investigating the feasibility of so-called "full stern" submarines. Three different methods of Pareto selection, drawn from the current literature, are installed in the algorithm and compared in terms of their ability to locate and define the frontier. A new concept in evolutionary computation—non-interbreeding competitive species—is introduced, allowing simultaneous optimization of four incompatible propulsor configurations. This produces a feasibility frontier with optimal propulsor configuration as a function of stern fullness. The ability of the algorithm to define a three-objective Pareto surface is demonstrated, by including minimal cavitation as a third objective.

The results provide evidence for the viability of full stern submarines, demonstrate the utility of genetic algorithms in obtaining Pareto design frontiers, and show that Pareto optimization is preferred to scalarized multi-objective optimization in general decision-making.

Thesis Supervisor: Justin E. Kerwin
Title: Professor of Naval Architecture

Acknowledgments

My thanks to those who have made my years at MIT, and in particular this research, possible and—more often than not—enjoyable:

Professor Jake Kerwin, for his friendly supervision and for lending his considerable respectability to this research,

Captain (retired) Al Brown, for five years worth of lively opinions and for his much-needed assistance in getting me started,

Commander (select) Mark Welsh, for resolving many of my self-imposed difficulties, both academic and administrative, and for putting up with frequent whining (much of which was warranted, by the way, especially where the MBTA was concerned),

Professor Jim Kirtley, who had the dubious distinction of approving both of my MIT theses, for time spent in providing me with good-natured and much-appreciated feedback over the past three years,

my committee members—Jake Kerwin, Al Brown, Jim Kirtley, Wallace Vander Velde, and George Apostolakis—for their guidance and support,

Todd, Gerard, Rich and Chris for technical and moral support, and for not complaining while MOGA evolved all over the lab computers,

Jennifer, Lyndon and Lacy, for being the best family anyone could hope to come home to and for reminding me that there are lots of things more important than work,

and my father and mother, Bill and Mary Thomas, for knowing I could do it even when I wasn't so sure.

Mark W. Thomas
May 12, 1998

Contents

List of Figures

List of Tables

Nomenclature

A		empirical constant for cavitation correlation
A_B	πR_B^2	area of maximum vehicle cross-section
c		duct chord length
C_D	$\frac{D}{\rho u_c^3}$	dissipation coefficient
C_{L_0}	$\frac{\text{Lift}}{\frac{1}{2}\rho \text{W}_1^2 \text{A}}$	blade tip lift coefficient at zero clearance condition
C_p	$\frac{p-p_\infty}{\frac{1}{2}\rho V_s^2}$	pressure coefficient
C_P	$\frac{Q\omega}{\frac{1}{2}\rho V_s^3 A_B}$	power coefficient
C_Q	$\frac{Q}{\frac{1}{2}\rho V_s^2 A_B D_B}$	alternative torque coefficient (see K_Q)
C_T	$\frac{T}{\frac{1}{2}\rho V_s^2 A_B}$	alternative thrust coefficient (see K_T)
C_V	$\frac{\nabla}{\pi R_B^2 L}$	volume coefficient
C_τ	$\frac{\tau}{\frac{1}{2}\rho V_s^2}$	shear coefficient
D		propeller diameter
D		shear dissipation ($\frac{\text{M}}{\text{T}^3}$)
D_B		maximum vehicle diameter
E		function mapping design parameters to objective space
f		function mapping schemata to string fitness
F_i		fitness of i^{th} member of population
$\bar{\mathcal{F}}$		cumulative fitness vector
G	$\frac{\Gamma}{2\pi R_B V_s}$	non-dimensional circulation
h		a schema, or similarity template
H	$\frac{2\frac{u_t^*}{V_s}\frac{r}{R_B}}{Z}$	non-dimensional local circulation
H	$\frac{\delta^*}{\theta}$	boundary layer shape parameter
H^*	$\frac{\theta^*}{\theta}$	boundary layer kinetic energy shape parameter
J	$\frac{V_s}{nD}$	traditional advance coefficient
J_B	$\frac{V_s}{nD_B}$	alternative advance coefficient
k_s		empirical constant for cavitation correlation
K_T	$\frac{T}{\rho n^2 D^4}$	traditional thrust coefficient
K_Q	$\frac{Q}{\rho n^2 D^5}$	traditional torque coefficient
ℓ		length of a bit string in number of bits
L		vehicle length

m		number of times a given schema appears in a population
n		propeller rotation rate ($\frac{\text{rev}}{\text{sec}}$)
O		order of a schema (number of non-"#" characters)
p		pressure
p_c		probability of crossover
p_m		probability of mutation
p_∞		total far-field fluid pressure
P		population size
P		propeller pitch[1]
P/D		propeller pitch-to-diameter ratio
Q		shaft torque
r		radial coordinate
R_B		maximum vehicle radius
Re	$\frac{V_s L}{\nu}$	Reynolds number
Re_c	$\frac{W_\infty c}{\nu}$	blade tip chord relative Reynolds number
T		thrust, in general
U		tip speed (used in cavitation correlation)
$\vec{U}$		velocity vector
u_e		magnitude of boundary layer edge velocity
u_i		i^{th} component of velocity
$\bar{u}_t$		circumferential mean self-induced tangential velocity
u_t^*		local self-induced tangential velocity
$\hat{u}_a$		total local induced axial velocity
$\hat{u}_t$		total local induced tangential velocity
V_s		reference velocity (vehicle speed)
W_1		relative fluid velocity at blade tip
W_∞		edge velocity of end-wall boundary layer
x		longitudinal coordinate (Huang experiments only)
$\hat{x}$		boundary layer curvilinear coordinate parallel to local surface
$\hat{y}$		boundary layer curvilinear coordinate normal to local surface
z		longitudinal coordinate
Z		number of propeller blades
γ		circulation per unit length ($\frac{\text{L}}{\text{T}}$)
Γ	$\frac{4\pi r \cdot \bar{u}_t}{Z}$	circulation on a lifting line ($\frac{\text{L}^2}{\text{T}}$)

[1] Propeller pitch is the distance the propeller would travel through its medium during the course of one revolution with no slip. It is generally a function of radius, and may also be defined as $2\pi r \tan\phi$, where ϕ, the pitch angle, is the angle between the intersection of the chord line of the section and a plane normal to the propeller axis.

Γ	$\oint u_i \, dx_i$	general circulation ($\frac{L^2}{T}$)
δ		length of a schema in number of bits
δ^*	$\frac{1}{u_e} \int_0^\infty (u_e - u_{\hat{x}}) \, d\hat{y}$	boundary layer displacement thickness
δ_{99}		value of $\hat{y}$ at which $\lvert u_{\hat{x}}(\hat{y}) \rvert = 0.99 u_e$
ϵ		tip gap to blade chord length ratio
θ		tangential coordinate
θ	$\frac{1}{u_e^2} \int_0^\infty (u_e - u_{\hat{x}}) u_{\hat{x}} \, d\hat{y}$	boundary layer momentum thickness
θ^*	$\frac{1}{u_e^3} \int_0^\infty (u_e^2 - u_{\hat{x}}^2) u_{\hat{x}} \, d\hat{y}$	boundary layer kinetic energy thickness
$\vec{\Theta}$		vector of objective function values
κ	$\frac{\bar{u}_t}{u_t^*}$	Goldstein reduction factor
κ		air content correction factor
λ		ratio of tip gap to maximum blade thickness at tip
ν		kinematic viscosity ($\frac{L^2}{T}$)
ρ		fluid density
σ	$-C_{p_{\min}}$	cavitation index
τ		viscous shear
τ_w		viscous shear at hull surface (wall)
$\vec{\psi}$		vector of design parameters
ω		angular velocity ($\frac{\text{rad}}{\text{sec}}$)
∇		hull volume, less gear and shafting estimates (see Appendix B)
#		wild card character in a schema

Chapter 1

Introduction

This research is undertaken to support two independent theses:

1. The volume and arrangement advantages offered by full stern submarines may be realized, without efficiency penalties, by properly designed ducted propulsors.

2. Valid design optimization in the presence of multiple objectives must be based on knowledge of the realizable solution space. This requires definition of the Pareto frontier, which may be obtained by genetic algorithm.

An existing exploratory design and analysis code for submarines with ducted propulsors is revised and upgraded to perform the calculations needed for optimization. This code is then used as the evaluator in an original Pareto genetic algorithm, with dual objectives of propulsive efficiency and usable hull volume. The algorithm seeks the Pareto frontier by constantly pressuring a random initial population of variants toward improvement in terms of both objectives simultaneously.

1.1 Ducted Propeller Systems

The potential benefits of enclosing a marine propeller in a duct were proposed as early as the 1930's [58]. Depending on the duct's shape and construction, these benefits can include greater efficiency, mechanical protection or reduced noise. In some cases, however, the additional design, production and maintenance costs of early applications offset any benefit obtained. This concern remains relevant to some extent today, and is why the majority of marine vessels continue to use open propellers.

One rather obvious benefit of a duct is that it can prevent propeller damage and fouling when the vehicle is operated near obstacles, in shallow water or under ice. When protection is the primary function, a duct is perhaps more accurately called a shroud, although terminology varies in the literature. Protective shrouds are usually

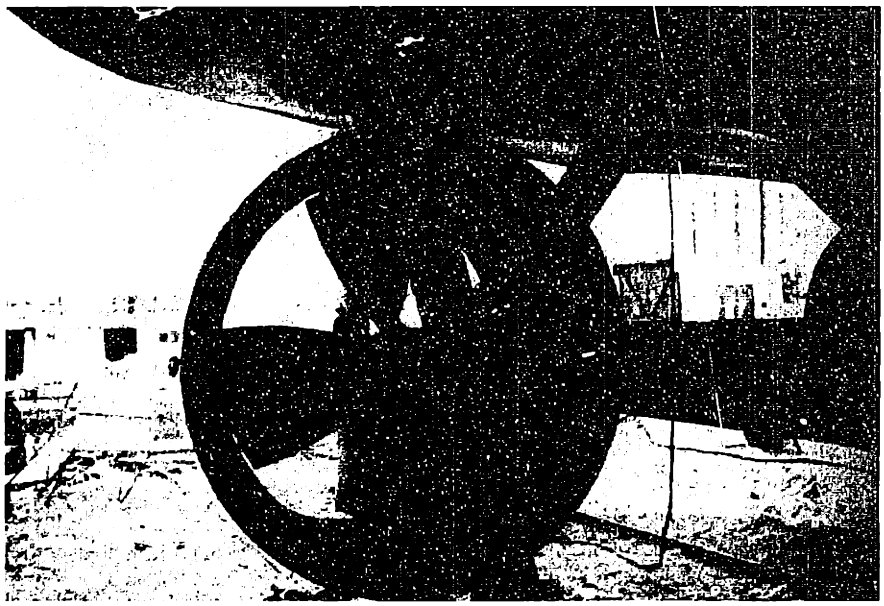

Figure 1-1: A ducted propeller on a surface vessel. Taken from Gillmer and Johnson, *Introduction to Naval Architecture*, Naval Institute Press, 1982.

thin and straight in cross-section and are not intended to have significant hydrodynamic properties. They are aligned more or less with the direction of flow—so as to generate no force—and may be slotted or perforated to allow through-flow. These are fairly common on small submersibles such as autonomous underwater vehicles (AUVs) and torpedos.

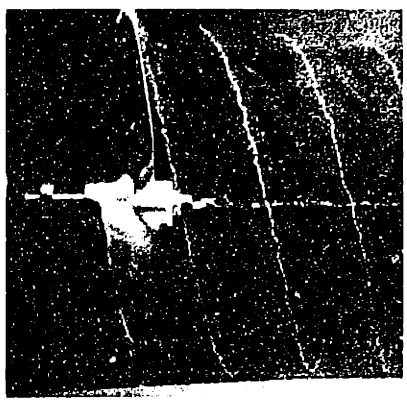

Figure 1-2: Tip vortex cavitation on an open propeller.

A duct may also be used to decrease the minimum propeller diameter for a given thrust; that is, to increase the maximum load that the propeller blades can carry. This was the primary motivation for early applications on tugboats. An open propeller is normally designed to be lightly loaded at the blade tips; any significant pressure difference across the blade at the tip will cause the flow to spill over, resulting in a tip vortex as shown in Figure 1-2. The production of a tip vortex requires energy and therefore causes a loss of thrust as well as possible cavitation, noise, vibration and erosion [10]. However, if a duct surrounds the propeller and the clearance between the blade tips and the duct inner surface is small, a pressure differential can be maintained across the blade tip. The result is an increase in the load-carrying capacity of the blade, a reduction in the required

propeller diameter, or some combination of the two.

Both propeller protection and increased load capacity are possible with simple shroud-like ducts. Further benefits are possible if the duct is given a thickness and camber distribution and/or is operated at a non-zero angle of attack, in which case it becomes an annular hydrofoil (see Figure 1-3). The duct is then capable of altering

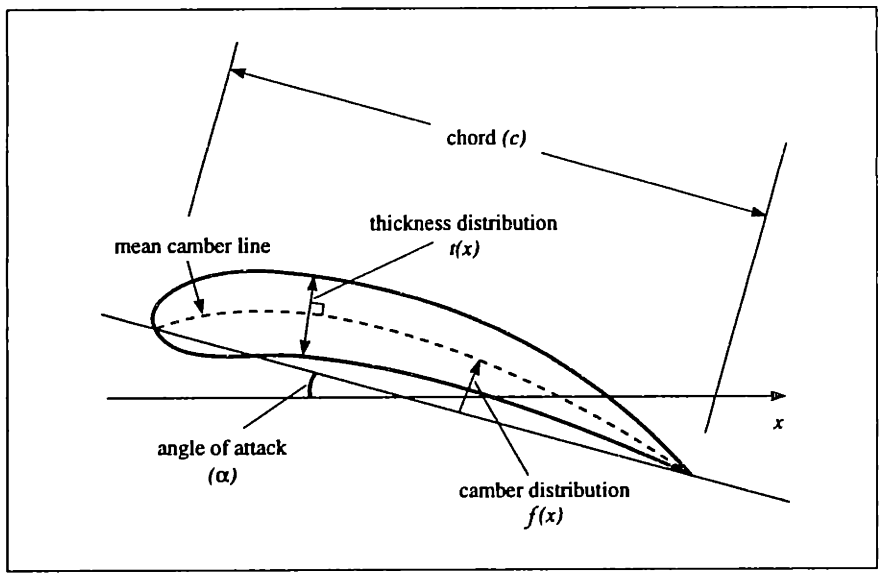

Figure 1-3: Typical duct cross-section with parameter definitions.

the propeller inflow velocity as well as generating lift.[1] This is illustrated in Figure 1-4a, which shows the stern of an axisymmetric vehicle operating at constant speed. Thrust provided by the rotor exactly balances the viscous and pressure drags acting on the vehicle. Streamlines generally follow the contour of the hull, perhaps with some small contraction imposed by the suction of the rotor.[2] Figure 1-4b shows the same configuration with the propeller now surrounded by a thin duct, represented by its two-dimensional cross-section. If the duct's mean camber line lies along a former streamline as shown, the duct generates no lift; its only effect on propulsive efficiency is some small axial force due to its viscous drag. Now let the duct rotate slightly about the propeller tip such that it attains a small positive angle of attack relative to the inflow, as in Figure 1-4c. The Kutta condition dictates negative (clockwise) rotation in the surrounding velocity field to prevent infinite velocity around the sharp trailing edge. This rotation may be quantified by a contour integration of velocity around the duct; it has dimension of length squared per unit time and is known as

[1] In very general terms, ducted propellers which decelerate the inflow are quieter and less efficient than equivalent open systems and are referred to as pumpjets. Ducted systems which accelerate the inflow are noisier but more efficient and are referred to as nozzles, or sometimes as Kort nozzles after their inventor.

[2] Streamlines aft of the rotor actually follow a helical path; those in the figure may be thought of as a centerline slice through the wake.

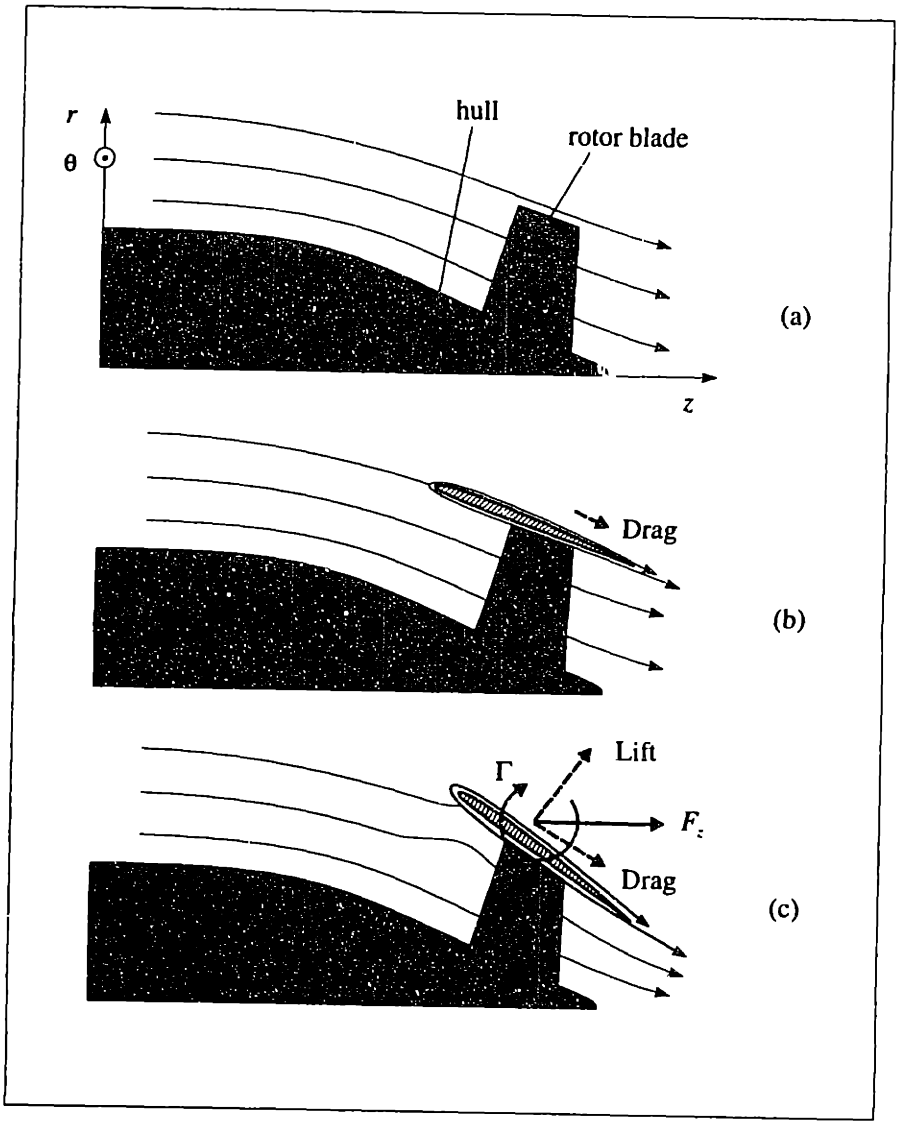

Figure 1-4: Effect of duct on propulsion system. (a) Non-ducted system. (b) Thin duct aligned with former streamline. (c) Duct loaded by placing it at a non-zero angle of attack.

the *circulation* (Γ) of the duct. Application of the Kutta-Joukowski law

$$\text{Lift} = \rho \vec{U} \times \vec{\Gamma} \tag{1.1}$$

reveals that the duct is now subject to a lift force normal to the direction of the local flow. Viscous drag is still present and acts parallel to the camber line, along with additional induced drag due to the lift [54]. The radial component of the resultant force has no effect on the vehicle's propulsion system, since the duct is axisymmetric, but the axial component F_z must be compensated by rotor thrust in order for forces on the vehicle to balance (note that if the angle of attack were negative, the lift would be in the opposite direction shown and the duct would provide thrust).

The effect of the loaded duct on propeller inflow is indicated by the altered path of the middle streamline; this alteration will affect propeller forces and thus overall efficiency of the vehicle. The blades, like the duct, are lifting surfaces and are subject to the Kutta-Joukowski force. The difference is that the vorticity vector of the rotor blade shown in the figure lies in a z–r plane, while that of the duct lies in a z–θ plane (the blade Γ, if shown, would be edge-on in the figure). According to Equation (1.1), any variation in the axial (z) component of velocity at the rotor alters the torque required to turn the shaft, while variation in tangential (θ) velocity affects the thrust produced.[3] Both of these quantities affect propulsive efficiency. A loaded duct, therefore, affects a vehicle's overall propulsion characteristics in two primary ways: indirectly, through alteration of the propeller inflow, and directly, by generating lift..

Whether and how the combination of these effects can be made beneficial to the vehicle is not obvious, because the mechanisms described above are coupled with each other and with body forces as well. The increased velocity due to the suction effect of the propeller will lower the fluid pressure at the stern and increase the pressure drag on the body; this added drag may be augmented or diminished by the duct circulation. The propeller and duct loads, however, are dependent on the inflow velocities presented to them, and inflow direction is very much a function of the body shape at the stern. Stern fullness—the abruptness of transition from maximum body radius to the rotor hub radius—thus affects the velocities induced by the propeller and duct which in turn affect drag on the hull. It may be possible to manipulate this interaction among the hull, duct and propeller by adjusting design parameters. If so, it would be of interest to know which combination of parameters—that is, which propulsor configuration—requires the least power to propel a given hull form at a given steady speed. Such information would be particularly useful in investigating the feasibility of full stern submarines, which are discussed below.[4]

1.2 Full Stern Submarines

The full stern submarine is a conceptual departure from traditional submarine design, involving a relatively abrupt transition from maximum hull radius to propeller hub radius at the stern. Figure 1-5 compares a notional full stern profile with that of an ideal, minimal drag profile and the common parallel mid-body/tapered stern profile. The ideal profile represents a trade-off between friction drag and pressure, or form,

[3] Assuming that the propeller continues to rotate at the same speed, of course.

[4] Capt. Harry Jackson provided the author's first exposure to the concept of full stern submarines at an MIT Professional Summer course in 1995 [50].

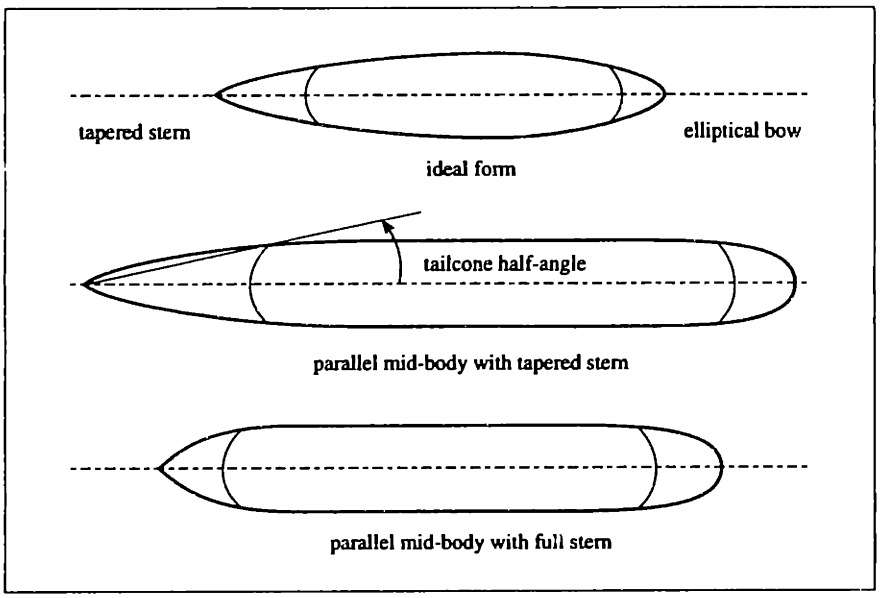

Figure 1-5: Notional full stern submarine profile, as compared to the ideal (in terms of drag) profile and the common parallel mid-body/tapered stern profile. The internal walls represent possible pressure hull boundaries.

drag on the hull. Friction drag is directly related to surface area and fluid velocity at the surface; pressure drag is a more complex function of viscosity and momentum recovery. It is sufficient to note here that pressure drag is minimal on a long, slender body with a slowly varying cross-section. Such a body will experience relatively high friction drag, however, as it requires more surface area than a shorter body of the same volume. An optimum length-to-diameter ratio should exist, then, which minimizes the sum of these drags. This ratio has been estimated at 6:1, although the total drag curve in the region of the minimum is quite flat [10].

There are several reasons why the ideal profile and optimal length-to-diameter ratio are not common among operational submarines. Given some required volume, these characteristics it may require an unacceptably large diameter or allow insufficient length to contain internal machinery and systems. The non-constant radius of the ideal hull increases production costs and presents difficulties in both construction and maintenance (e.g., dry-docking). It also makes the shape of the inner (pressure) hull problematic. If the pressure hull conforms to the outer hull, as in the upper profile of Figure 1-5, internal arrangement and deck layout are difficult and some pressurized volume may be unusable. If pressure hull walls are made parallel regardless of the outer hull curvature, extra volume is introduced between the two hulls. This increases the total volume to be propelled, possibly resulting in an overall efficiency decrease. Such considerations have driven the majority of submarine designs toward a parallel

mid-body profile, such as the middle profile of Figure 1-5. The advantages realized in cost and arrangeability are assumed to offset the increase in drag, which in any event is slight due to the flatness of the total drag curve.

Parallel mid-body designs generally retain the long, tapering stern shape of the minimal drag profile. Tailcone half-angles do not normally exceed 20 degrees [88]. Construction complications associated with non-constant radius are therefore not eliminated entirely by the parallel mid-body plan; they are simply avoided along most of the length. Also, the attachment of the long stern section to the pressure hull remains a structural challenge, increasing in difficulty as the tailcone angle decreases.

A short, full stern section would reduce or eliminate these concerns. Also, since propulsion systems and other high-volume machinery are usually located aft, a full stern would allow greater flexibility in arrangement. The additional volume might also be used to increase accessibility for maintenance and/or to reduce the overall length of the vehicle. Reduced length would bring most parallel mid-body designs closer to the optimal length-to-diameter ratio. Such shortening has been investigated by Warren [88] and shown to be feasible, possibly even resulting in greater maximum speed or propulsive coefficient.[5] Full sterns also produce relatively large wake fractions, or "viscous shadows." Propellers operating in these regions of reduced velocity may realize an efficiency advantage over an equivalent open-water system. This potential benefit is taken up in greater detail in Section 2.3.1.

Despite the possible advantages of full sterns, they remain notably absent among current designs. One reason for this—the desire to limit form drag—has already been mentioned above. Another reason, more compelling, is to avoid flow separation. As the fluid near the body surface flows over the stern, it decelerates due to the increasing pressure.[6] If this deceleration reaches a critical value, the flow will detach from the hull and proceed more or less directly downstream; this condition is known as separation (see Figure 1-6). The likelihood of separation at a given speed depends primarily on body shape. Bluff, or blunt, sterns with rapidly changing cross-sections tend to cause separation where slender, tapered sterns do not. The consequences of separation are quite undesirable, and body profiles with the potential to cause separation are avoided. Separation causes a significant increase in drag due to the low pressure region it creates behind the vehicle. Fluid in a separated wake is relatively stagnant on a large scale, but is at reduced pressure due to the requirement of continuity with the

[5] Propulsive coefficient is a measure of marine propulsive efficiency, but is not referred to as such because it has no absolute upper limit. See Gillmer and Johnson [30] or Burcher and Rydill [10] for an overview. A more detailed analysis, relating efficiency to the various physical phenomena involved, is given by Dyne [23].

[6] Of course, it is actually the *vehicle* which is moving, but a stationary vehicle with fluid flowing past it is an equivalent situation and is consistent with most models in the literature.

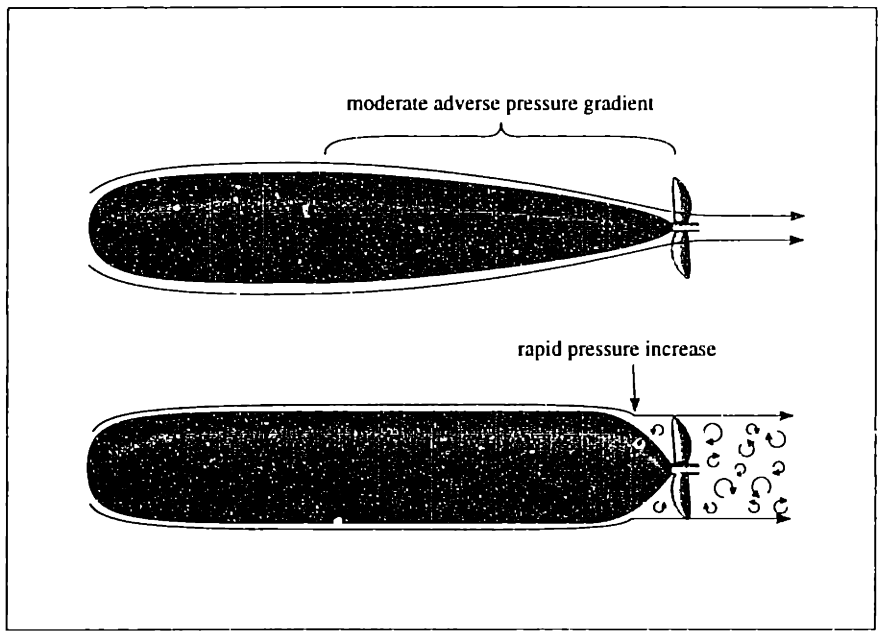

Figure 1-6: Graphical comparison of attached and separated flows, showing how streamlines detach from the hull at the point of separation. Fluid in the separated region is at low pressure, causing an increase in drag.

free-stream pressure outside the wake [67]. The propeller is forced to operate in this region of erratic local velocities, making efficiency unpredictable at best. Additionally, separation is generally accompanied by an increase in noise level. This, of course, is a particularly unwelcome effect for an operational submarine.

Fortunately for proponents of fuller sterns, the stern shape is not the only factor affecting the likelihood of separation. As previously mentioned, the suction effect of the propeller is felt upstream for a distance of a few propeller radii and a duct, if present, may also alter the flow field locally. It is conceivable, given the complex interaction among propulsor components, that ducted propulsors can be designed to delay or prevent separation which would otherwise occur on a full stern without degrading overall efficiency.

This presents several interesting questions for submarine designers. There are benefits to be had by increasing the taper of the stern, but separation must be strictly avoided. Propeller and duct effects, which are interdependent and coupled with the shape of the stern, may delay or prevent separation. How abrupt can the stern transition be, and what is the best propulsor configuration for that profile? Propulsive efficiency is critical, particularly in non-nuclear submarines, where a twenty to thirty year life cycle means that operating costs will exceed those of design and construction. Does the maximum attainable efficiency decrease as the stern becomes fuller, and if

so, are the volume benefits worth the efficiency penalties? Ducts are known to affect propulsive efficiency and cavitation noise. Should a duct be used, and if so, should it accelerate or decelerate the propeller inflow? Guide vanes, or stators, can be placed forward of the rotor to provide efficiency-enhancing pre-swirl or aft of the rotor to recover wake energy in the form of tangential velocities. Any benefit so obtained, however, must be balanced against increased system complexity and friction drag. Should stators be used, and if so, should they be forward or aft of the rotor (or both)?

The answers to these questions involve not only an unwieldy number of design variables but also two possibly conflicting objectives (volume vs. efficiency). This research, in addition to seeking answers, is also intended to demonstrate that such questions can and *should* be answered satisfactorily in any design process. Further, they can and should be answered without ad hoc or a priori assumptions regarding the relative importance of the objectives. These are general issues, transcending hydrodynamics and submarine design, and will be dealt with as such. A likely field in which to begin is that of multiple criteria analysis, discussed below.

1.3 Multiple Criteria Analysis

Decisions are not based on isolated, independent criteria. Any choice among alternatives, be it the propulsor configuration for a submarine, foreign policy or what to have for breakfast involves prioritization of the consequences, either implicitly or explicitly. Certainly there are instances where a single criterion dominates (whether one should exit a burning building, say), but these are generally cases where no decision is required; the choice is obvious. In situations where a non-trivial decision *is* required, its difficulty may be assumed directly related to the number of objectives (implications, consequences) involved and the degree to which they are conflicting. This assumption has a direct impact on engineering, or design, optimization.

The impact is due to the fact that any optimization, including design, involves decision-making. In fact, a decision is usually required before the optimization process can begin; its purpose is to define the optimum. Take for example the linear and non-linear programming techniques which exist for the "optimization" (minimization, usually) of mathematical functions. Prior to invoking them, the solution must first be defined as the parameters which return the smallest value when evaluated by the function.[7] Such a definition—though implicit and trivial in the case of mathematical function minimization—is more obscure in engineering optimization, where multiple

[7] In a constrained problem, of course, the solution must also satisfy constraints.

conflicting objectives are likely to be present. The defining of optima in these situations is complex enough to warrant its own field, that of multiple criteria analysis.

Multiple criteria theory had its origins in the 1950's; the earliest use of multiple objectives in mathematical programming is usually attributed to Kuhn and Tucker in a 1951 paper [57]. Interest in the field intensified in the 1970's, as evidenced by the publishing of several new techniques and practical applications. As the number of techniques grew, basic procedural differences became apparent among them which allowed classification of the field. MacCrimmon [61], Hwang and Masud [49] and later Dlesk and Liebman [20] attempted to establish a general classification system by distinguishing between the terms multi-*objective* (describing an engineering or design-type problem, generally having at least a partially continuous solution space) and multi-*attribute* (describing a decision-type problem with a finite number of predetermined alternatives), classifying both as sub-categories of multi-*criteria* analysis. Hwang and Masud [49] further classified then-current optimization techniques depending on the point in the process at which subjective preference information (the decision mentioned above, essentially) is required. This remains an important distinction among techniques, and it is emphasized in a recent comprehensive review of the field by Miettinen [65]. The present research takes the position that it is advantageous to delay subjectivity as long as possible (it can in fact be avoided completely in rare cases). Prematurely applied subjectivity excludes regions of the solution space from consideration before the *form* of the solution space has been investigated. This can result in poor decisions, as will be seen shortly.

Among the techniques falling under the general heading of multiple criteria analysis (listed here without regard to classification by subjectivity requirements) are global criterion, ϵ-constraint, bounded objective, goal criterion, utility analysis, analytical hierarchy process and goal programming.[8] Each of these is a unique philosophy of what constitutes optima in the presence of multiple objectives. In choosing among these techniques (again, making a decision), one accepts a definition of optima. A definition combined with a search mechanism is an algorithm, which may be used to locate the defined optima given a mapping between parameter and objective space (i.e., given an objective function).

Such algorithms require not only a definition of the solution but also a way of distinguishing between *superior* and *inferior* candidates during the search. Again, this distinction is trivial for single objective minimization—one solution is obviously better than another if its objective value is lower, and the optimum is the feasible so-

[8] Generally, whether a method is intended mainly for decision-making or for design optimization is not obvious from its name; both types are represented among these examples.

lution having the lowest objective value of all feasible solutions.[9] The global optimum is normally sought, although some problems require identification of both global and local optima (see Figure 1-7).

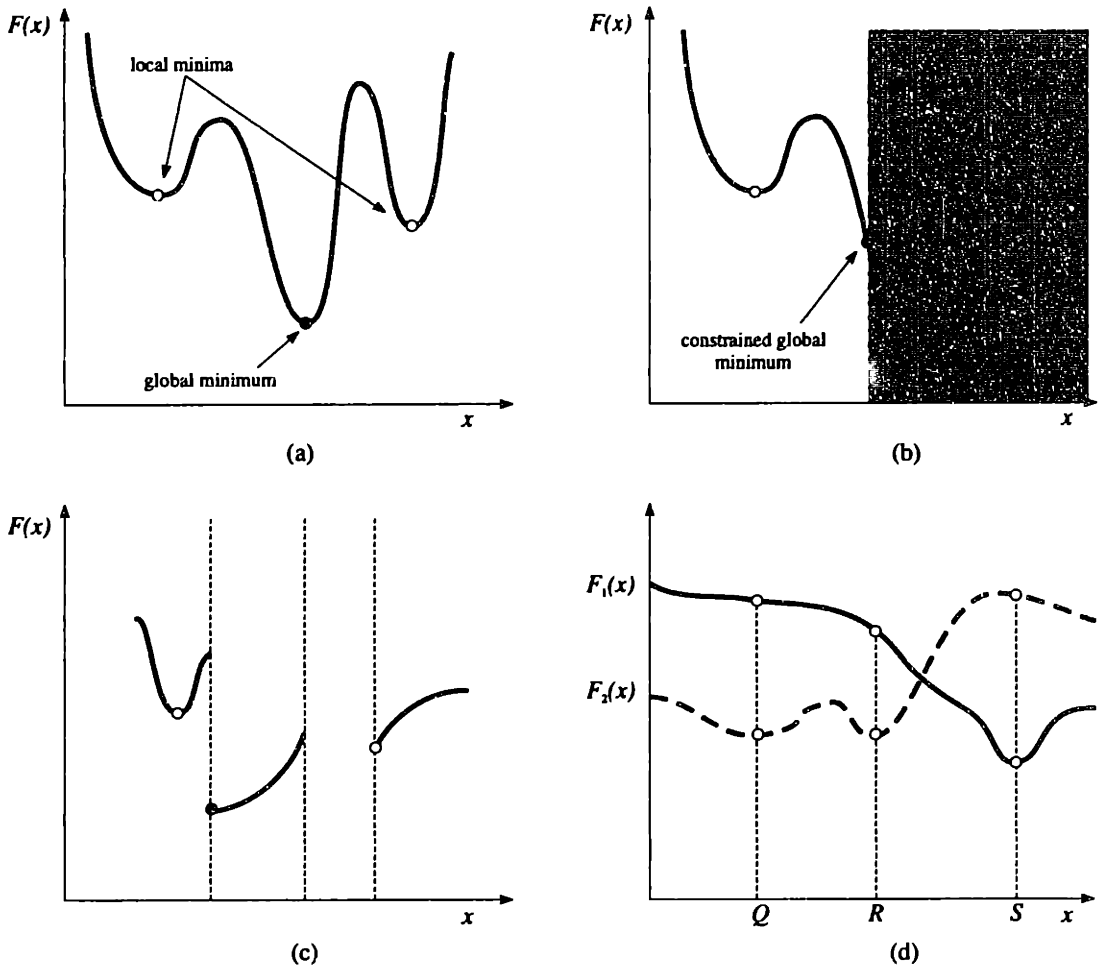

Figure 1-7: Comparison of optima for various types of objective functions. (a) Single objective with global and local minima. (b) Single objective subject to constraint. (c) Single discontinuous and disjoint objective. (d) Ambiguous situation with two objectives.

Unfortunately, such simple definitions of "better" and "best" are not suitable in the presence of multiple objectives. The reason is shown graphically in Figure 1-7d. Two objective functions are present, F_1 and F_2. Although both of the candidate solutions Q and S minimize one of the objectives, there is no rationale for declaring either solution "better" than the other.[10] The ambiguity cannot be resolved by assuming the objectives to be independent and performing separate optimizations. Solutions

[9] The definition also applies to single objective problems where the objective is to be maximized, by simply changing the sign of the objective function.

[10] Simply defining the better solution as the one which minimizes the *sum* of the objectives is equivalent to assigning them equal importance, which of course is subjective. Even if this were considered acceptable, there remains the problem of normalizing non-commensurate objective dimensions (the cooling requirements and weight of an electric motor, for example [84]).

Q and R, for example, both minimize objective F_2 but neither is very attractive in terms of F_1.

In fact, even the term "optimum" itself is ambiguous in the presence of multiple criteria, as evidenced by the number of published methods for defining it. This ambiguity may be better appreciated with a simple example. Consider a team of engineers tasked with designing an optimal nuclear power plant, in terms of construction cost and safety, for a given power rating. They are hopefully not interested in the cheapest plant regardless of safety nor are they likely to be interested in the safest plant regardless of cost. While these two extremes may provide useful information by bracketing the feasible design space, neither is likely to be considered a "good" design because of the conflicting nature of the objectives (i.e., an acceptable level of safety will not be achievable at low cost). Variations *between* these two extremes must be investigated, but "between" requires definition itself. Certainly the team is not interested in any design which is less safe *and* more costly than some other valid design. In fact, they are interested only in those for which no safer *and* less expensive valid design exists, i.e., those which are not *dominated* by any other feasible design. This, essentially, is the concept of Pareto optimality and is the only way that multi-criteria optimality may be defined without subjectivity.[11]

Although usually attributed to Pareto [69], this concept was proposed in an earlier work by Edgeworth [24]. The definition may be stated formally as follows [74]:

> *"A feasible solution to a multi-criteria (multi-objective) optimization problem is Pareto optimal if there exists no other feasible solution that will yield an improvement in the performance of one criterion without causing a decrease in performance of at least one other criterion."*

This means that there are multiple, and perhaps infinitely many, optimal solutions in the Pareto sense to most engineering design problems. These solutions are all equally "good"; without subjective preference information there is no distinguishing among them in terms of desirability. The set of all Pareto solutions is variously known as the Pareto set, the efficient set, the Pareto frontier, or the non-dominated frontier. It always lies at the boundary between feasible (realizable) objective space and infeasible (non-realizable) objective space, although the converse is not necessarily true, as shown in Figure 1-8. These plots represent hypothetical situations where every conceivable combination of design parameters has been evaluated (or built) and each combination's relevant objective values (A and B) have been determined. The

[11] Cost and safety are obviously not the *only* factors which would be considered when designing a nuclear power plant, but they are certainly primary factors. One difficulty in using the Pareto definition of optimality is how to interpret and display results when several objectives are present.

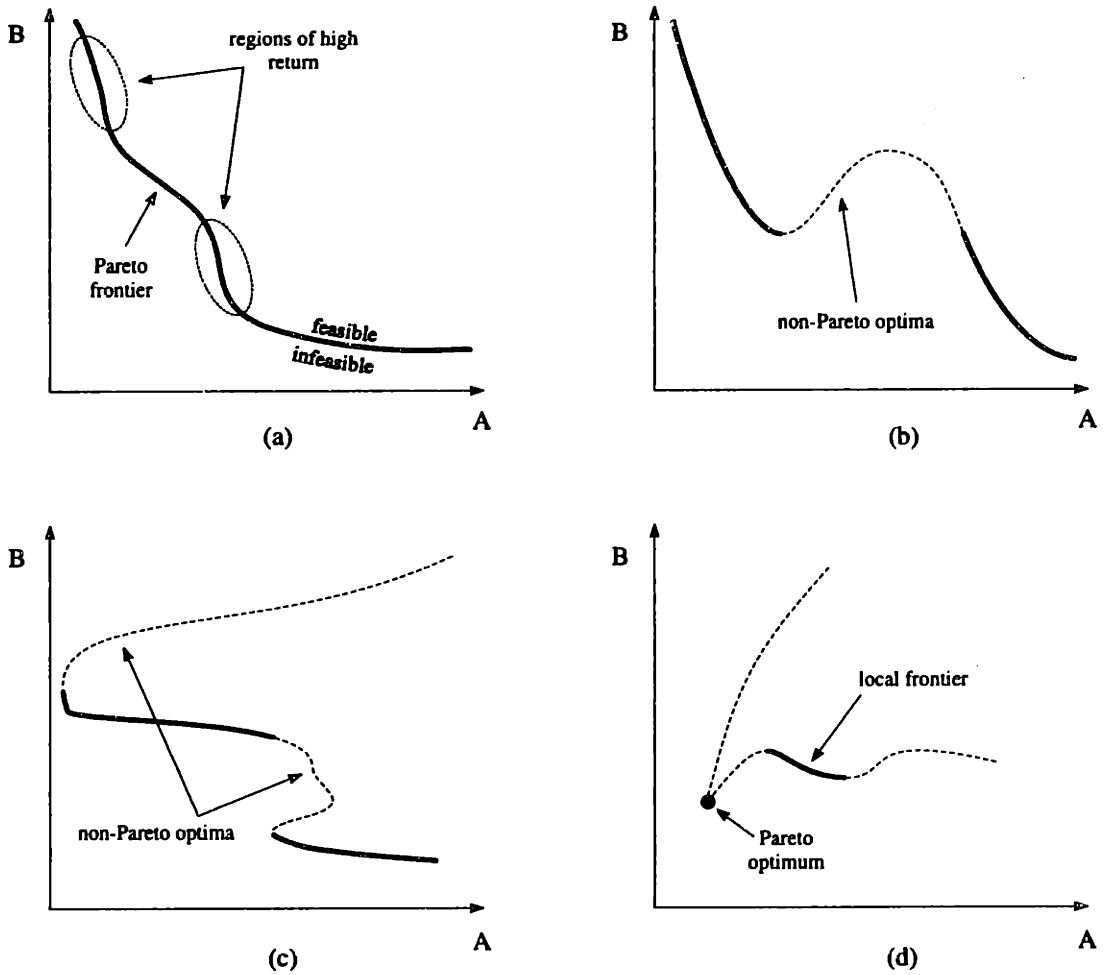

Figure 1-8: Notional Pareto frontiers in two-objective space, with objectives A and B to be minimized. (a) Continuous frontier, showing regions where objective B can improve significantly with little tradeoff of objective A. (b) Discontinuous frontier, with non-Pareto optima at the concave boundary of the feasible solution space. (c) Discontinuous frontier with two non-Pareto, feasibility-limited regions. (d) Non-conflicting objectives, showing local optima "trap" at the feasibility boundary.

non-dominated frontier in each case makes up a portion of the feasibility boundary, obtained by plotting all of the feasible variations in objective (A-B) space.

It is difficult to imagine how an algorithm could be designed to search for infinitely many (or even several) different solutions simultaneously. This is why optima-defining methods such as those previously listed apply some form of user-supplied subjectivity to reduce the number of candidates—in most cases to one. Depending on the method, this subjectivity will involve the assumption of one or more of the following: a global optimum and its location, the existence and values of independent optima for each objective, the mathematical formulation of a decision-maker's preferences, objective thresholds and/or goals, or convexity of the Pareto set. These assumptions are ma-

nipulated such that the set of all objective values for any candidate solution can be combined into a scalar (for example, in the global criterion method, this scalar is the candidate's Euclidean distance in objective space from the location of the assumed global optimum). The combining of multiple objective values into a single measure of goodness is known as *scalarization*, and methods which combine objectives are known as scalarization methods. The value of the scalar provides the search process with a means of distinguishing among Pareto optima. The search through design (parameter) space is thus guided toward a particular solution in objective space by the definition of the optimum. When and if this particular Pareto optimum is located by the search process, the corresponding design parameters are declared to be the solution.

A serious objection to scalarization is that the form of the feasibility boundary is not taken into account. Accepting a definition of optima a priori amounts to an uninformed choice of one Pareto solution from among many. This implicitly assumes that gradients on the frontier are uniform, when in reality the frontier may contain concavities or even discontinuities. Rapidly changing gradients, or "knees," in the frontier should certainly be considered in decision-making, as these are regions where a slight trade-off of one criterion may provide relatively large returns in others (the "regions of high return" in Figure 1-8a). This information is not available when using scalarization. Depending on the assumptions required for the chosen scalarization method, there is also a possibility that the search process will become trapped at a local optimum, such as that of Figure 1-8d.[12] None of the methods listed above can guarantee global Pareto optima [65].

Also, while the solutions defined by these methods are Pareto optimal *if they exist*, they do not *necessarily* exist in an arbitrary solution space. For example, the assumption that a decision-maker's preference function is a linear combination of objectives—a common feature among weighting methods—will not allow the identification of any optima in concave regions of the Pareto frontier, regardless of the search method used [65, 80].[13] Even if the solutions are identifiable using the chosen definition, the search mechanism may not be able to locate them. Most search processes rely on gradient information and therefore will have trouble with discontinuous or disjoint objective spaces, these being not uncommon in engineering design.[14] Thus a multi-objective optimization algorithm can suffer from two general limitations: that

[12] Figure 1-8d also shows the only situation where subjectivity can be completely avoided in multi-objective optimization: compatible objectives with a single, feasible non-dominated solution.

[13] Note that concavity of the feasibility boundary does not necessarily indicate a discontinuous frontier, as may be seen by comparing Figure 1-8a and b.

[14] The gradient referred to here is the change in an objective value with respect to a design parameter, as opposed to the gradient of the frontier itself which is the change in one objective value with respect to another.

of its scalarization and that of its search mechanism. The degree to which these affect solution quality, of course, is dependent upon the nature of the particular objective space and cannot be generalized.

Even if these limitations are acceptable, the user of a scalarization method must allow that any solution obtained is non-unique and is subject to change. Any new or altered assumptions, whether due to new information, a new decision-maker, or changing opinions of the same decision-maker will require that the problem be resolved to locate a different non-dominated point.

Based on these considerations, it seems that the ideal situation from a decision-maker or designer standpoint is to begin with a complete and accurate rendering of the Pareto frontier and proceed on the basis of this information. This has not gone unrecognized; there have been methods proposed which are capable of generating some or all Pareto solutions simultaneously for discrete problems (e.g., Rosenman and Gero [74]). Similarly, some authors of scalarization methods recommend that the assumptions be systematically varied and multiple solutions found prior to final selection (e.g., Dlesk and Liebman [20]). An algorithm which produces such information in a single pass, however, would seem preferable to one which requires repeated applications. Indeed, in light of all the issues discussed thus far in this section, the ideal multi-objective optimization algorithm would be capable of using the Pareto definition of optimality exclusively (thus avoiding a priori subjectivity), would locate all Pareto optima simultaneously (thus exposing regions conducive to tradeoffs without repetition), and would not be hindered by pathological objective spaces.

One approach which meets these criteria is a Pareto genetic algorithm. Genetic algorithms process populations rather than individual solutions and are therefore uniquely suited for defining a Pareto frontier in a single run. Since they do not rely on gradient information, they are unaffected by discontinuities and are not as likely to become trapped at local optima. This research attempts to exploit these characteristics in obtaining the frontier for ducted propeller submarines. The goal here will be validation of the method itself and demonstration of its superiority over scalarization, as well as drawing hydrodynamic conclusions from the results.

1.4 Chapter Summaries

Chapter 2 describes the submarine design and analysis program Ducted Propulsor Lifting Line (DPLL), which is eventually used as the evaluator for the genetic algorithm. The initial phase of this research involved an extensive re-write and upgrade

of DPLL, with the intention of making it suitable for use in full stern optimization.

Chapter 3 provides validation of the revised DPLL by modeling published experiments and comparing the calculated results to measured data. The program's thrust and torque calculations are verified, as well as the performance of a new boundary layer routine in predicting displacement thickness and flow separation. Sample output is included from a run typical of those in the optimization phase of the research, to provide qualitative verification of DPLL's accuracy. Convergence characteristics of the program are presented.

Chapter 4 includes a brief history of genetic algorithms and the theory behind them, along with a description of the algorithm developed during this research. The three alternative Pareto methods of selection used in this algorithm are described. Incremental and generational genetic algorithms are compared. The concept of competitive species in a genetic algorithm, apparently unique to this research, is introduced.

Chapter 5 documents the results and presents the Pareto frontier obtained by each of the three selection methods. The cumulative results of the GA are used to estimate the location of the dominated feasibility boundary. The composition of the entire feasibility boundary, in terms of design parameters, is presented. The results of a three-objective optimization are given, with minimal cavitation included as the third objective.

Chapter 6 presents hydrodynamic and optimization-related conclusions. Several specific topics are recommended for future work, including possible improvements to DPLL and further submarine optimization issues.

Chapter 2

DPLL version 2.0

2.1 Description and Motivation

Ducted Propulsor Lifting Line (DPLL) is an exploratory design and analysis code for underwater vehicles, in particular those with ducted propulsors. Using input geometry, design parameters, and operating conditions, it calculates all torques, thrusts and drags acting on the vehicle and its propulsion components (see Figure 2-1). Since these forces are assumed to be strongly coupled, the solution process is iterative.

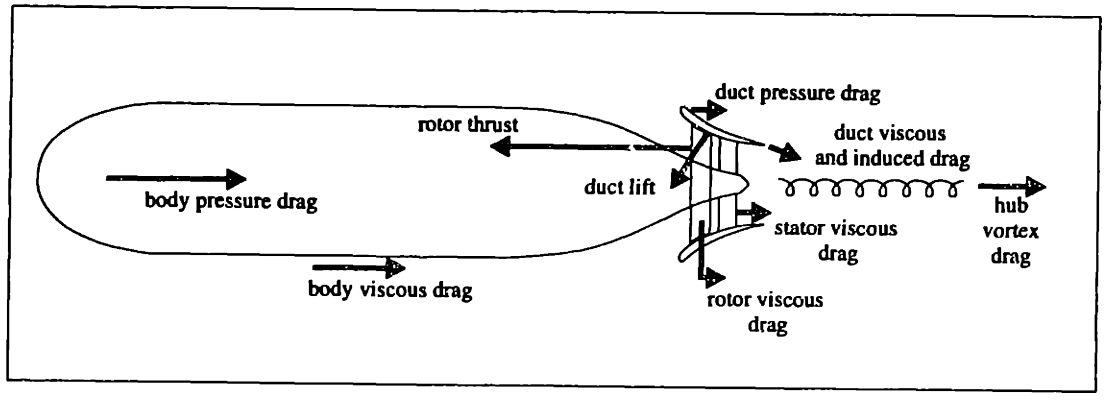

Figure 2-1: Coupled forces acting on an underwater vehicle.

Version 2.0 of DPLL, developed during this research, is the result of an extensive re-write of Taylor's original version [82] and includes several new features. Basic methodology remains largely the same as that described in Taylor's written thesis [83]. Hereafter, version 2.0 will be referred to simply as DPLL.

The distinguishing characteristics of DPLL are its ability to model an underwater vehicle as a system, thereby capturing the interaction among the various components involved, and to account for the effects of a contracting wake. It is also capable of converging to a self-propelled condition (i.e., balancing thrust against drag), predicting flow separation on the body, and estimating the effect of wake fraction. These

features make it more accurate and adaptable than simple parametric estimation such as [4] or non-coupled lifting line/actuator disk analysis such as [59, 93, 46] in the exploratory design phase.

Like any model, DPLL is characterized and limited by its assumptions. In keeping with its intended use as an exploratory design tool, these assumptions are aimed at achieving the greatest accuracy possible while emphasizing simplicity and minimal run time. The use of a lifting line model for propeller blading, rather than a three-dimensional vortex lattice, is an example of a tradeoff between these objectives. Other simplifications include the use of assumed friction coefficients for calculating viscous drag and a zero thickness, mean camber line duct model. For these reasons, DPLL is less precise than state-of-the-art force/flow analysis tools (e.g., Reynolds-averaged Navier-Stokes, or RANS, routines coupled with propeller design codes such as PBD [56, 5, 6]).[1] Such tools are computationally intensive and require considerable setup and run time (experienced users may require several days, given some arbitrary configuration, to obtain an accurate converged solution). In comparison, DPLL trades off high-order accuracy for greater speed, simplicity and ease of use.

In short, DPLL is intended to fill a current gap between over-simplification and unnecessary precision in design at the exploratory level. It is motivated by the need for a relatively simple code with enough accuracy, flexibility and speed to be useful in preliminary submarine design.

2.2 Methodology

Propeller-induced wake vorticity in the DPLL model is concentrated into discrete streamtubes shed from lifting segment endpoints, as shown in Figure 2-2. The paths of these streamtubes (or streamlines, when represented in two dimensions) are determined by the local velocity calculated at numerous intermediate points. These local velocities are affected by the presence of the body and duct as well as the vorticity contained in the streamlines themselves. Once their paths have been determined, the streamlines are discretized into free vortex segments. Propeller-induced velocities in the wake are then calculated by summing the effects of all bound and free vortex segments at numerous wake field points between the streamlines. The field points compose a propeller-induced velocity grid, which is interpolated or extrapolated to control point locations on the duct and body surfaces to give the propeller induction

[1] The propeller blade design program PBD–14.3 and its earlier versions were developed under the supervision of Prof. Justin Kerwin at the MIT Marine Hydrodynamics Lab.

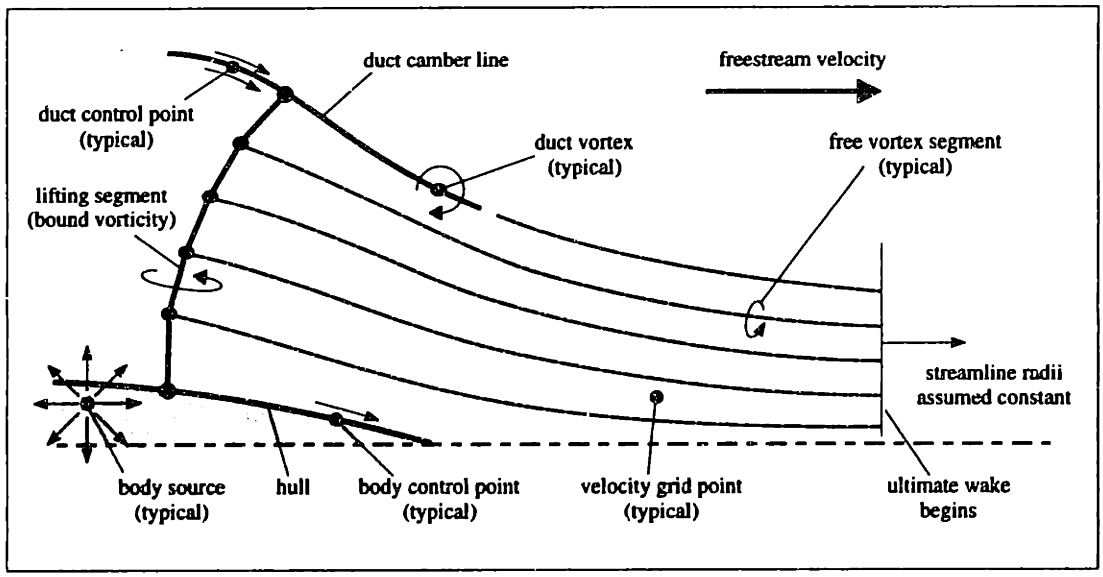

Figure 2-2: Centerline section of a notional DPLL wake, showing a typical grid point and the various components which affect velocity. No stator is present in this configuration. The rotor is modeled by five discrete lifting segments, each with its own input value of circulation.

there.[2] Since normal velocity at control points is required to be zero—no flow may pass through the hull or duct—body and duct singularity strengths must be adjusted whenever new propeller-induced velocities are calculated. When such adjustments are made, the local, non-propeller-induced velocities in the wake are affected and streamline paths must be recalculated. This process is repeated until changes in grid velocities from one iteration to the next (among several other criteria) are within some tolerance. Convergence of the process results in an accurate total velocity field from which forces and torques on the lifting segments may be calculated.

2.2.1 Lifting Line Model

DPLL will model up to three propeller stages; these may be rotors or stators in any combination. The stages may have up to ten blades. Each blade is modeled by a lifting line, composed of vortex segments which induce rotational velocities normal to their axes in the surrounding flow. The magnitudes of these induced velocities are specified by the input circulation (Γ), or strength, of the segments. Figure 2-3 shows an example vortex segment oriented along the x-axis having length $|x_2 - x_1|$, zero thickness, and circulation Γ. At any point P in the vicinity, the velocities induced in the x and y directions (u and v respectively) are zero. The tangential component of

[2]Propeller induction is extrapolated rather than explicitly calculated at body and duct control points because these points lie on streamlines. Influence calculations become singular when the field point is on the axis of a vortex segment. This is also why propeller grid points in DPLL are located between streamlines.

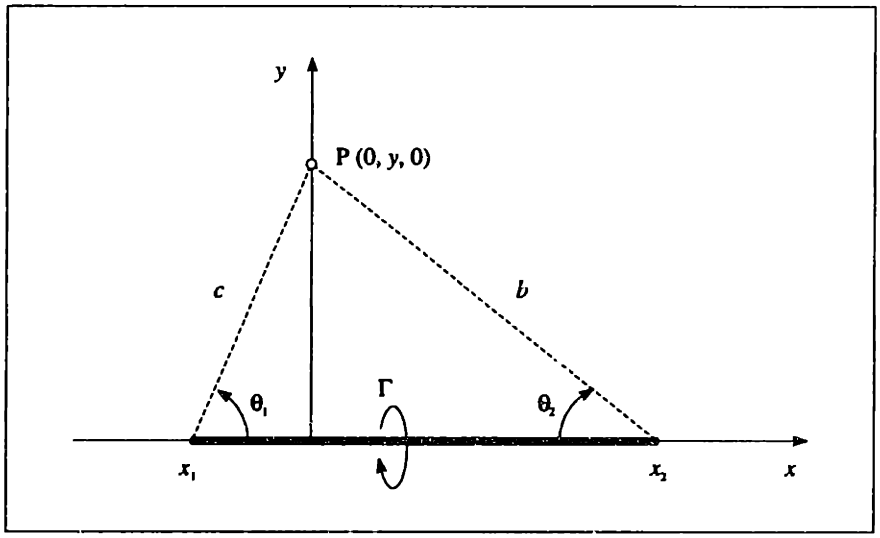

Figure 2-3: Vortex segment parameters, from Kerwin [54].

induced velocity w is a function of the field point's position relative to the segment and the circulation strength [54]:

$$w(y) = \frac{\Gamma}{4\pi y}\left[\frac{x_1}{b} - \frac{x_2}{c}\right] \tag{2.1}$$

$$= \frac{\Gamma}{4\pi y}\left[\cos\theta_2 + \cos\theta_1\right] \tag{2.2}$$

Placed in a flow field, such a segment will not only induce velocities but will also be subject to a force according to the Kutta-Joukowski law:

$$\vec{F} = \rho\vec{U} \times \vec{\Gamma} \tag{2.3}$$

In a lifting line propeller model, "blades" made of vortex segments are spaced symmetrically and aligned radially about a common axis of rotation. In rotating through the surrounding velocity field and advancing forward with the vehicle, these segments experience forces in the plane of rotation and in the direction of the axis, creating torque and thrust respectively.

DPLL assumes an ideal fluid and potential (zero curl) flow.[3] As a consequence, vortex lines cannot begin or end anywhere in the flow field; any vorticity present on the lifting segments must be shed into the wake.[4] This allows calculation of streamline vorticity based on lifting segment strengths, which are DPLL inputs. Figure 2-4 illustrates this relationship. A typical blade tip is shown, having a finite chord with an arbitrary distribution of bound circulation, γ_B. This bound circulation, essentially

[3] The only exception is when the hull boundary layer is calculated; see Section 2.3.1.

[4] Kelvin's theorem of the conservation of circulation states that for an ideal fluid acted upon by conservative forces (e.g., gravity) the circulation is constant about any closed material contour moving with the fluid. Thus, any motion that starts from a state of rest at some initial time will remain irrotational for all subsequent times [67].

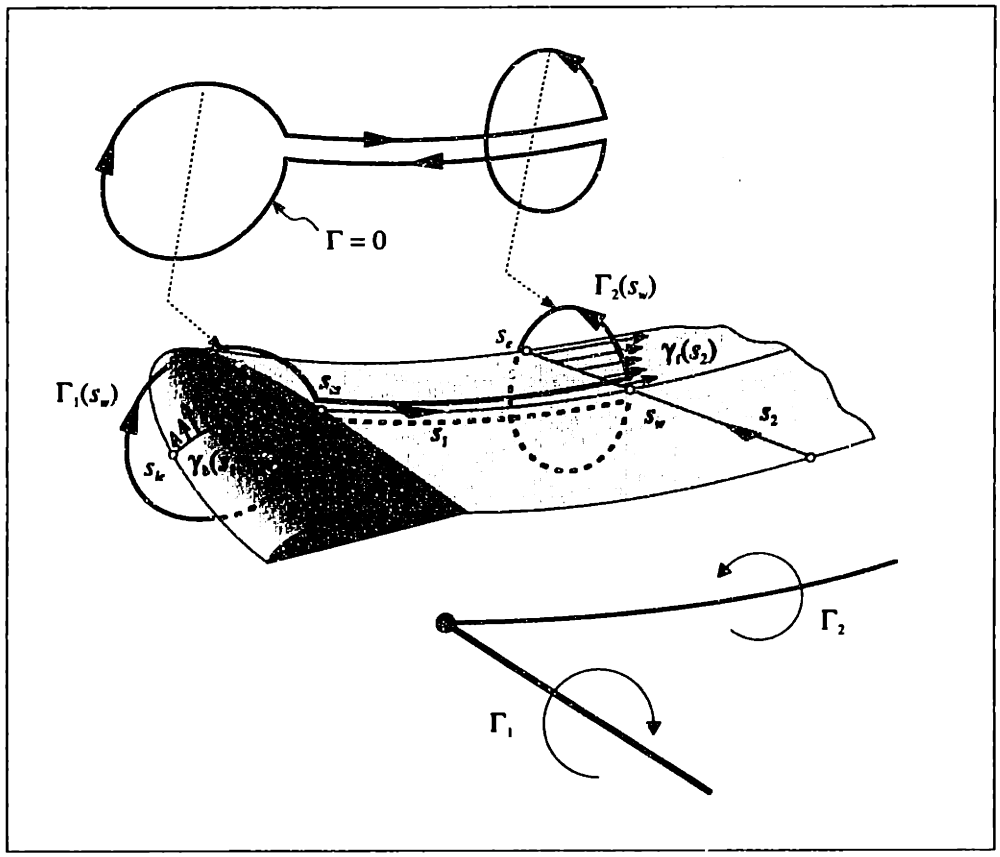

Figure 2-4: Bound circulation related to free vorticity in wake for a general lifting surface and for the corresponding lifting line model, from Kerwin [54]. In practice, wake vorticity sheets tend to "roll up" into concentrated tip vortices.

the velocity difference between the upper and lower faces of the foil, has dimension of circulation per unit length, or length per unit time. According to Kelvin's theorem, the circulation around the isolated closed path in the upper part of the figure is zero. If the path is moved onto the blade and wake as shown without cutting any lines of vorticity, the *total* circulation remains zero and the following relationship must hold:

$$\Gamma_1(s_w) = \int_{s_{le}}^{s_{te}} \gamma_b(s_1, s_w) ds_1 \tag{2.4}$$

$$\Gamma_2(s_w) = -\int_{s_w}^{s_e} \gamma_f(s_2) ds_2 = -\Gamma_1(s_w) \tag{2.5}$$

Differentiating this with respect to s_w gives:

$$\frac{d\Gamma_1}{ds_w} = \frac{d}{ds_w} \int_{s_w}^{s_e} \gamma_f(s_2) ds_2 = -\gamma_f(s_w) \tag{2.6}$$

The same relationship between bound and free vorticity applies when the blade is approximated by a lifting line. The chordwise distribution of circulation present on the actual blade is simply compressed to a vortex segment with the same value of

total circulation $\Gamma(s_w)$. The two-dimensional blade surface, with finite chord and span, is replaced by a one-dimensional line (as shown in the lower part of the figure), with the assumption that the effect on field points off the blade surface is unchanged.

If a lifting line is segmented and Γ is made constant (but not necessarily equal) on each segment, non-zero circulation gradients may occur at the nodes and the end-points. Lines of wake vorticity, or trailers, must begin at these locations and continue back through the wake to infinity in order to satisfy Kelvin's theorem. The strengths of the trailers are known from Equation (2.6), and their paths are determined by local velocities in the wake as discussed at the beginning of this section. They will induce velocities on each other and on the bound lifting segments themselves, in accordance with Equation (2.1). The sum effect of all lifting segments and all trailing vorticity at a field point may be calculated numerically (by discretizing the free vorticity, as is done in DPLL) or analytically. Analytic methods require the assumption of a constant radius wake, however, and are therefore inappropriate for use in DPLL.

A similar lifting line model is used for stators, which are non-rotating blades or guide vanes used to manipulate tangential velocities. Stators, like rotors, require input circulation distributions and are subject to torque and thrust. Stators may be placed upstream of the rotor to provide pre-swirl (with the intention of decreasing rotor torque), downstream of the rotor to recover wake energy in the form of tangential velocities, or both.[5]

2.2.2 The *H* Function

The circulation distribution of a lifting line propeller may be defined in terms of the velocity induced at the propeller plane by the sum effect of its helical trailers [46]:

$$\Gamma(r) = \frac{4\pi r \cdot \bar{u}_t}{Z} \tag{2.7}$$

where $\bar{u}_t$ is circumferential mean induced tangential velocity and Z is the number of blades. This quantity may be non-dimensionalized using a convenient reference length and velocity:

$$G(r) = \frac{\Gamma(r)}{L_{\text{ref}} V_{\text{ref}}} \tag{2.8}$$

DPLL uses maximum hull circumference and forward speed as the reference quantities (the rationale for using body radius as opposed to the more traditional propeller radius

[5]DPLL will also model multiple rotors, e.g., a contra-rotating configuration. This capability has not been tested or validated.

for normalization is discussed in [83]):

$$G(r) = \frac{\Gamma(r)}{2\pi R_B V_s} \tag{2.9}$$

which becomes, using Equation (2.7),

$$G(r) = \frac{2\frac{\bar{u}_t}{V_s}\frac{r}{R_B}}{Z} \tag{2.10}$$

The derivation of Equation (2.7) assumes an infinite number of blades uniformly distributed around the hub; its accuracy falls off rapidly for propellers of less than about ten blades [54]. If finite blade effects are to be taken into account, it is more appropriate to specify circulation in terms of *local* induced velocity (u_t^*) on the lifting lines, which is generally greater than the circumferential mean. This is the motivation for the H function proposed by Kerwin [55]:

$$H(r) = \frac{2\frac{u_t^*}{V_{ref}}\frac{r}{L_{ref}}}{Z} \tag{2.11}$$

Using the DPLL normalization quantities from above, this becomes:

$$H(r) = \frac{2\frac{u_t^*}{V_s}\frac{r}{R_B}}{Z} = G(r)\frac{u_t^*}{\bar{u}_t} \tag{2.12}$$

Lifting line segment strengths in DPLL are specified by H values at the hub and tip—the innermost and outermost segments, respectively—of each stage. These values are interpolated linearly to determine the circulation at intermediate points.

2.2.3 Goldstein Factors

Induced velocities in a propeller wake become circumferentially uniform downstream, regardless of the number of blades present. If local H values are used to specify bound vorticity strengths, as they are in DPLL, a relationship between local and circumferential mean velocity is needed in order to calculate the strength of wake vorticity using Equation (2.6). Such a relationship may be obtained by borrowing results from idealized propeller analysis.

In 1929, Goldstein first solved for the self-induction of an optimum, finite-bladed lifting line propeller in uniform flow [38, 8]. The solution proved to be a function only of geometry—the number of blades, the pitch-to-diameter ratio, and the radius of shed vorticity. Its relationship to the infinite-bladed solution is known as the Goldstein reduction factor [54]:

$$\kappa(r) = \frac{\bar{u}_t(r)}{u_t^*(r)} \tag{2.13}$$

where $\bar{u}_t$ is circumferential mean induced tangential velocity and u_t^* is local induced tangential velocity.

If a wake of constant radius is assumed, the Goldstein factor can be calculated directly using Equation (2.7) for $\bar{u}_t$ and the individual filament (single blade) equations of Lerbs and Wrench [59, 93] for u_t^*. Unfortunately, the constant radius wake assumption is not appropriate in DPLL and neither of these calculations is suitable. However, it is reasonable to believe that the *relationship* between the two velocities is nearly the same regardless of wake contraction or the optimality of circulation. If that is so, then induced velocity at a lifting line can be approximately related to mean induced velocity in the plane of the propeller for contracting wakes. The velocity formulas mentioned above are simply used as if the configuration were ideal; the ratio of the two velocities is taken and used to convert known local induced velocity on a blade to unknown mean velocity in the propeller plane, or vice-versa.[6]

Such use of idealized ratios to compute velocities in non-ideal configurations was proposed by Taylor [83]. He referred to these ratios as "generalized Goldstein factors" when describing their use in the original DPLL. The methodology remains the same in the current version. Non-dimensional local circulation values are used to determine the corresponding circumferential mean values via the generalized Goldstein factor κ:

$$G(r) = \kappa(r)H(r) \tag{2.14}$$

where a value of κ is calculated for each lifting segment once per iteration as if idealized conditions held. The G values thus obtained are used to determine the strengths of the trailing vorticity per Equation (2.6).

2.2.4 Calculation of Propeller Induction

The assumption of a circumferentially uniform wake simplifies the calculation of trailer paths and allows the wake to be represented in two dimensions, as in the centerline slice of Figure 2-2. The actual velocities, however, are induced in three dimensions. Each segment shown in Figure 2-2 represents a three-dimensional vortex ring, formed by sweeping the segment about the axis of symmetry. A typical, simplified ring is shown in Figure 2-5. The z and r coordinates of the two-dimensional segment's endpoints are known from the growing and discretizing of the streamline. If the streamline were represented in three dimensions, however, it would be seen to follow a helical path; that is, the endpoints of the segment would be displaced tangentially.

[6] A parameter needed for the Lerbs and Wrench formulas which does not appear in Equation (2.7) is the pitch angle. This is the angle between the intersection of the chord line of the section and a plane normal to the propeller axis, assumed equal to the departure angle of the trailer at the lifting line.

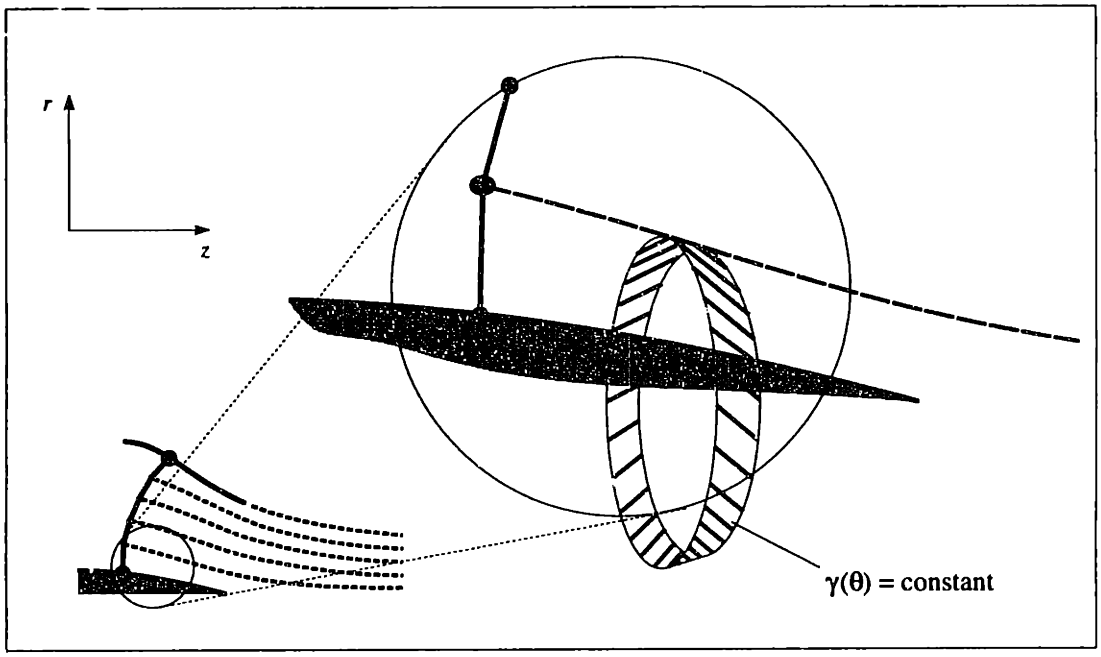

Figure 2-5: Relationship between 2-D trailer segments and 3-D vortex rings.

The tangential (θ) coordinates are found by dividing the segment's axial span (dz) by the latest calculated local axial velocity to obtain elapsed time. Local tangential velocity is then calculated and multiplied by this elapsed time to obtain the tangential span of the segment. A subroutine in DPLL written by Buchoux [9] uses the three-dimensional endpoint coordinates to convert the segment to its ring equivalent and then calculates the influence of the ring at any given field point. This process is repeated to obtain the influence of all trailing segments (rings) in the wake on all grid points; it comprises the bulk of DPLL's processing time.[7]

2.2.5 Duct Modeling

Propeller stages in DPLL are surrounded by a non-rotating duct of zero thickness, modeled by discrete vortex rings distributed along the mean camber line. Like the propeller lifting lines and the wake vortex rings, these singularities experience forces and induce velocities in the surrounding flow. Duct vortex rings are similar in effect and form to the free vortex rings used to model the wake (Figure 2-5), but have zero axial span.

DPLL can treat the duct as a design or an analysis problem, according to a user-specified flag in the input file. In design mode, the total duct load and a preliminary

[7]The sum effect of all trailing segments at some field point does not result in the *total* velocity there, as it does not include the effects of the body and duct singularities or the freestream. These additional effects are necessary for growing new streamlines, but are calculated separately.

camber line shape are input. The duct's overall angle of attack and scaling factors for the input shape are re-calculated at each iteration to result in the required load and zero normal velocities at duct control points.[8] Providing the option of duct design complicates the program considerably; however, it also avoids some potential problems which will be discussed presently.

Design mode requires the axial extents of the camber line as well as the load and an initial shape as inputs. The radial position of the duct is also specified, although indirectly, in that it must intersect the tip of a selected governing propeller whose radius is fixed. The camber line scaling factors, angle of attack and vortex ring strengths are computed given the input parameters and the latest velocity field around the duct. When the camber line is moved to its newly solved position, however, the mutual induction between the duct vortex rings and the body source rings is altered. The result is non-zero normal velocities at control points, meaning that the system must be re-solved, starting from the new duct position and using updated induced velocities. Duct design is therefore an iterative sub-process within DPLL's main loop, involving repeated load computation and camber line adjustment until the calculated load is within some tolerance of the input.

Figure 2-6 illustrates the freedom DPLL has in adjusting duct camber. A notional camber line which might result from an input B-spline vertex file (the preliminary camber shape mentioned above) is shown and denoted as "base camber."[9] The lighter lines are a few of the shapes that this input camber line might be allowed to assume during the duct design process. These lines represent changes in camber *magnitude* and are produced by scaling the radial coordinates of the defining vertices, thus redefining the B-spline itself.

Camber scaling is one of two degrees of freedom available to DPLL when re-designing the duct. The other is the duct angle of attack, involving a rigid rotation of the camber line about its pivot point (the pivot point is the previously mentioned intersection of the camber line with the governing propeller tip). For a given velocity field, base camber and design load, there is a unique combination of camber scaling and angle of attack which results in zero normal velocity at all duct control points.[10]

Alternatively, DPLL may be run in analysis mode. This simply causes the duct design routine to be bypassed; the camber line specified by the input B-spline vertices is never altered, except for an initial radial translation to intersect the governing

[8] In DPLL, total duct load is the algebraic sum of the non-dimensional strengths of all duct vortex rings. Duct control points lie on the camber line between rings.

[9] A camber line with an inflection point is shown for purposes of illustration. In practical foils, the direction of curvature normally does not change.

[10] This unique solution is not necessarily feasible; see Section 2.3.6.

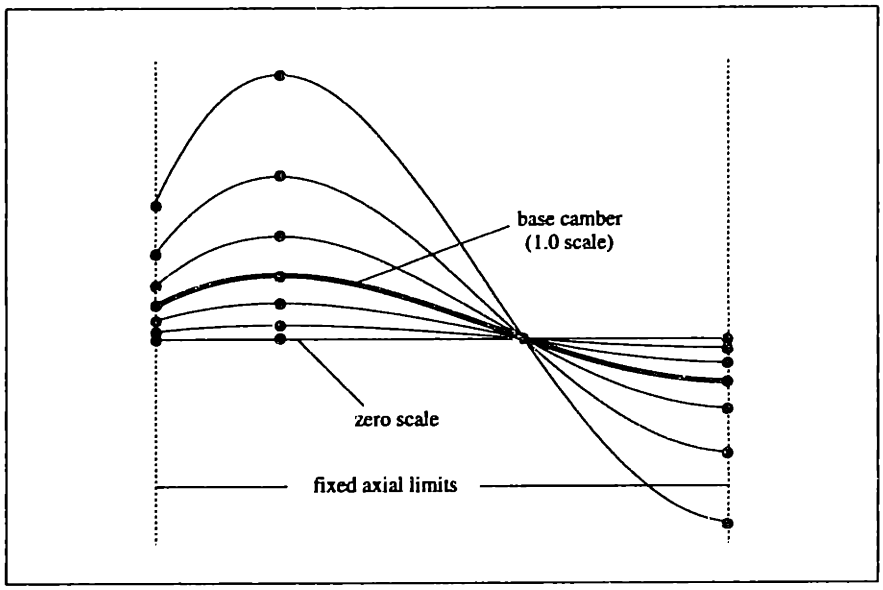

Figure 2-6: Possible variations of an input duct shape due to scaling. The nodes shown are at the axial locations of the camber line's B-spline vertices. The vertices themselves, to which the radial scalings apply, are not shown.

propeller tip.[11] The duct angle of attack is constant and implicit in the curve defined by the input vertices. Duct load is calculated rather than specified; it will fluctuate and settle, along with the surrounding velocity field, as the iterations progress.

It is an apparent contradiction that an arbitrary camber line shape *and* angle of attack may be specified in analysis mode, while a unique combination of camber scaling and angle of attack exists for a given load in design mode. This is explained by a previously unmentioned constraint on DPLL's duct design solution—the requirement of small leading edge load, or shock-free entry. The requirement of shock-free entry simply ensures that the leading edge of the camber line is reasonably well aligned with the local inflow. It involves the two most upstream vortex rings on the camber line. If the circulation values of these rings are Γ_1 and Γ_2, and the distances between the first two duct control points (i.e., the one between the rings and the one just downstream of the second ring) are Δz_1 and Δz_2 respectively, then the shock-free constraint is simply

$$\Gamma_1 = \Gamma_2 \frac{(\Delta z_1)^2}{(\Delta z_2)^2} \tag{2.15}$$

The vortex rings and control points are cosine spaced on the camber line, so the forward panel is always the smaller of the two. This makes Γ_1 small and thus forces the leading edge toward the direction of the inflow.

[11] This initial translation occurs in design mode also.

In a real flow, a misaligned leading edge can cause a cavitation bubble or flow separation, depending on the degree of misalignment, the fluid velocity and the ambient pressure. Currently, DPLL has no way of predicting cavitation or separation anywhere on the duct. The solution will therefore converge regardless of leading edge alignment, but will be invalid if the corresponding physical duct is subject to leading edge cavitation. The shock-free constraint on design mode simply helps prevent the user from specifying physically infeasible ducts.

Clearance must exist between a physical, non-rotating duct and the blade tips. This clearance, known as the tip gap, is not explicitly modeled in DPLL; that is, circulation of practically any magnitude is allowed at the blade tips and the lifting lines intersect the duct camber line. A non-zero tip load is one of the advantages of a ducted propulsor, as discussed in Section 1.1. While DPLL will accept vanishing circulation at the blade tips (i.e., an input of zero H at the outermost lifting segment, as would be expected for a wide gap or an open propeller), this condition may cause negative propeller-induced velocities in the wake due to the grid extrapolations used. If the magnitudes of these negative velocities happen to be greater than that of the potential flow in the vicinity, negative *total* velocity results and the program will be unable to grow valid streamlines in the following iteration. DPLL is therefore best described as a small gap model, carrying load at the blade tips but assigning secondary importance to tip gap effects.[12]

2.2.6 Hull Geometry

The hull profile is input as a set of B-spline vertices and modeled by submerged source rings. These rings are similar in concept to the vortex rings discussed above, but they induce radial rather than tangential velocities (a typical body source ring is shown in cross-section in Figure 2-2, along with the direction of its induced velocities). The strengths of these source rings are re-calculated during each wake iteration to zero the normal velocities at all hull and duct control points. The number and density of control points on the hull is controlled by the user; a source ring corresponds to each control point so that the system is properly constrained.

2.2.7 Thrust and Torque Coefficients

Thrust and torque coefficients in DPLL (C_T and C_Q respectively) are normalized to reference velocity and body cross-sectional area for reasons discussed in [83]. This

[12] For an analysis of the effect of tip gap variation on propeller performance, see McHugh [63].

is in contrast to the use of propeller rotation rate and diameter for normalization, resulting in the more common coefficients K_T and K_Q.

The *total* tangential velocity seen by a lifting segment is due to its rotation, its self-induction, and the induction of all other lifting segments present. For an incremental annulus of a Z-bladed lifting line propeller, the thrust produced according to Equation (2.3) is:

$$\delta T = Z\rho(\omega r + \hat{u}_t)\Gamma\,\delta r \tag{2.16}$$

where $\hat{u}_t$ is the total local induced velocity and Γ is the circulation of the lifting segment across the increment. Using the quantity $\frac{1}{2}\rho V_s^2 A_B$ as the reference force, where A_B is the cross-sectional area of the body at its maximum radius, and substituting the definition of G from Equation (2.9) gives the incremental thrust coefficient:

$$\delta C_T = 4Z\left(\frac{\omega r + \hat{u}_t}{V_s}\right) G\frac{\delta r}{R_B} \tag{2.17}$$

Thus the total thrust coefficient for the propeller is:

$$C_T = 4Z\int_{R_H}^{R_T}\left(\frac{\omega r + \hat{u}_t(r)}{V_s}\right) G(r)\frac{dr}{R_B} \tag{2.18}$$

where R_H and R_T are the hub and tip radii, respectively.

Likewise, the total axial velocity at a lifting segment is due to the combined effects of forward speed and induced velocities from the hull, duct and propeller stages (including self-induction). The torque on an incremental annulus of a lifting line propeller, by Equation (2.3), is:

$$\delta Q = Z\rho(V_s + \hat{u}_a)\Gamma\, r\delta r \tag{2.19}$$

where $\hat{u}_a$ is the total local axial induced velocity. Using the quantity $\frac{1}{2}\rho V_s^2 A_B D_B$ as reference torque, where D_B is the maximum body diameter, and again applying the definition of G, the incremental torque coefficient is:

$$\delta C_Q = 2Z\left(1 + \frac{\hat{u}_a}{V_s}\right) G\frac{r}{R_B}\frac{\delta r}{R_B} \tag{2.20}$$

and the total torque coefficient for the propeller is:

$$C_Q = 2Z\int_{R_H}^{R_T}\left(1 + \frac{\hat{u}_a(r)}{V_s}\right) G(r)\frac{r}{R_B^2}\,dr \tag{2.21}$$

where R_H and R_T are again the hub and tip radii, respectively.

2.2.8 Summary of Inputs

Table 2.1 summarizes the primary inputs for the stand-alone version of DPLL, most of which have been discussed in the preceding sections. The modified version of DPLL used for the optimization portion of this research requires essentially the same inputs, but some are held constant. These will be noted in Chapter 4.

DPLL Input	
Parameter	Format
hull geometry	B-spline vertices
number of propeller stages	1–3
number of blades[a]	2–10
propeller rotation rate	$J_B = \frac{V_s}{nR_B}$ (advance coefficient)[b]
circulation at blade root[a]	$H = \frac{2\frac{u_i^*}{V_s}\frac{r_{hub}}{R_B}}{Z}$ (non-dimensional local circulation)[c]
circulation at blade tip[a]	$H = \frac{2\frac{u_i^*}{V_s}\frac{r_{tip}}{R_B}}{Z}$ (non-dimensional local circulation)[c]
propeller tip location[a]	percent duct chord length
duct length and position	B-spline vertices
duct camber	multipliers for radial scaling of B-spline vertices
duct load	$\sum_{i=1}^{N} \frac{\Gamma_i}{2\pi R_B V_s}$ (non-dimensional circulation)[d]
governing propeller tip radius[e]	r_{tip}/L
viscosity	$\mathrm{Re} = \frac{V_s L}{\nu}$ (Reynolds number)
miscellaneous	number of control points, lifting segments, and wake panels, location of ultimate wake, friction coefficients, speed, etc.

[a] One value required for each propeller stage present.
[b] V_s = forward speed, n = rotation rate, R_B = max body radius. $|J| > 10.0$ implies a stator.
[c] u_i^* = local self-induced tangential velocity; Z is the number of blades.
[d] N is the number of vortex rings used to model the duct camber line.
[e] The governing propeller is an arbitrary designation when multiple stages are present.

Table 2.1: DPLL input parameters and their format

2.3 Major Revisions and New Features

Following are the significant changes incorporated during the re-write of DPLL v1.0. New features include hull boundary layer modeling, hub vortex drag, self-propulsion, calculation of cavitation index, and several new convergence criteria. The duct design process is extensively revised and includes failure recovery and convergence criteria

of its own. General normalization procedures for both internal and external values are standardized throughout. Hereafter, version 2.0 will again be referred to simply as DPLL.

2.3.1 Boundary Layer Modeling

DPLL computes boundary layer displacement thickness, momentum thickness and shape factors at hull control points and uses these calculations to predict flow separation and estimate the effect of wake fraction. It employs a boundary layer marching routine (hereafter, MRCHBL) originally written as a stand-alone program by Prof. Mark Drela[13] and used by permission [22]. MRCHBL is run in direct mode, meaning that edge velocity—the freestream velocity just outside the viscous boundary layer, often given the symbol u_e—is prescribed. This input comes from DPLL's inviscid solution for velocities at the hull surface.

It is an approximation to assume that the outer velocity in the viscous boundary layer is equal to inviscid surface velocities on the hull. The error involved is not prohibitive at DPLL's level of accuracy, however, and this technique gives very good results to first order as will be seen in Chapter 3. In general, it is possible to iterate a process such as this, by returning the calculated boundary layer thickness to the inviscid solver, adjusting the source strengths appropriately, calculating new inviscid "edge" velocities at the displaced surface, and so on. This is known as a weakly or loosely coupled process.[14] The boundary layer routine in DPLL, however, is run only once, after all inviscid iterations are complete. This is because coupling will cause direct interacting methods such as MRCHBL to diverge if separation occurs [66], and separation must be expected at times when modeling full stern vehicles.

Estimation of Wake Fraction

Wake fraction is a beneficial phenomenon associated with propellers operating in viscous boundary layers. It results from the velocity reduction and resulting boundary layer growth near the surface due to friction and pressure effects. If the velocity parallel to the surface of the hub (i.e., normal to the lifting line) is reduced without affecting tangential velocity, then Equation (2.3) reveals that less torque is required to produce the same amount of thrust. This could be considered an efficiency "increase"; it is more precisely a partial recovery of energy lost to viscous friction. In any event, it is a condition which becomes more prominent as the fullness of the stern increases and which cannot be captured by the inviscid calculations of DPLL alone.

[13] Associate Professor of Aeronautics and Astronautics, MIT.

[14] A strongly coupled scheme solves the boundary layer and inviscid flow equations simultaneously.

Velocity within a viscous boundary layer increases from zero at the surface (the empirically determined "no-slip" condition [75]) to the freestream value at its edge, as shown schematically in Figure 2-7. A rotor operating within a boundary layer will

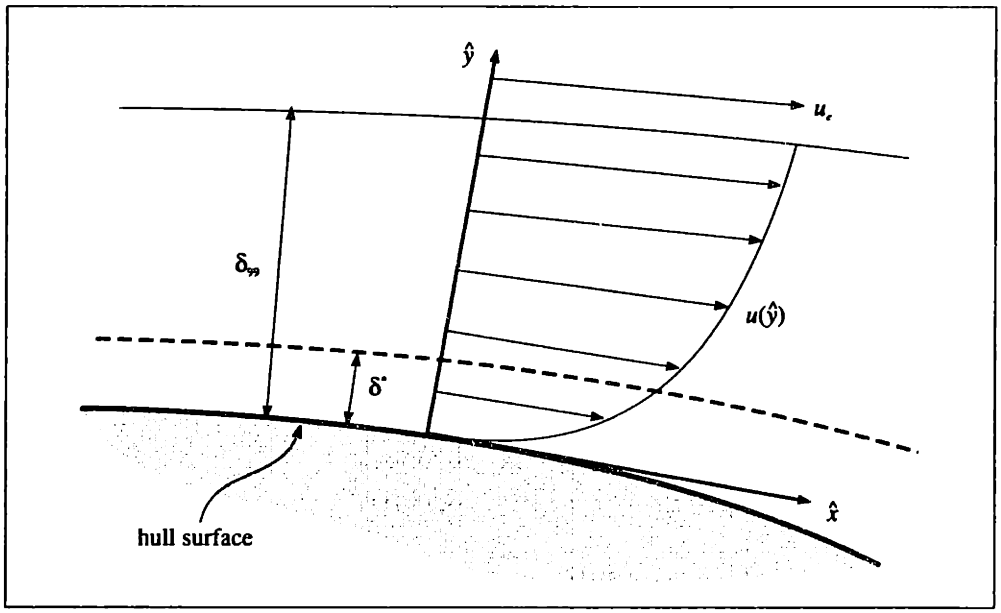

Figure 2-7: Typical boundary layer velocity profile.

therefore see a radius-dependent velocity reduction as compared to the corresponding inviscid flow. The velocity profile shape is in general dependent upon the local pressure gradient and the curvature of the surface. It is also a function of edge velocity, the viscosity of the fluid, and whether the flow in the boundary layer is laminar or turbulent.

Laminar and turbulent boundary layer flows are governed by the same integral momentum equation, obtained by integrating a combination of the continuity and simplified x-momentum equations term by term across the boundary layer [21]:

$$\frac{d\theta}{d\hat{x}} + (2+H)\frac{\theta}{u_e}\frac{du_e}{d\hat{x}} = \frac{\tau_w}{\rho u_e^2} = \frac{C_f}{2} \tag{2.22}$$

where

$$\theta = \text{momentum thickness} = \frac{1}{u_e^2}\int_0^\infty (u_e - u_{\hat{x}})u_{\hat{x}}\, d\hat{y} \tag{2.23}$$

$$\delta^* = \text{displacement thickness} = \frac{1}{u_e}\int_0^\infty (u_e - u_{\hat{x}})\, d\hat{y} \tag{2.24}$$

$$H = \text{momentum shape factor} = \frac{\delta^*}{\theta} \tag{2.25}$$

$$\tau_w = \text{wall shear stress} = \rho\nu \left.\frac{\partial u}{\partial \hat{y}}\right|_w \tag{2.26}$$

$$C_f = \text{skin friction coefficient} = \frac{2\tau_w}{\rho u_e^2} \tag{2.27}$$

In laminar flow, the three variables θ, H and C_f can be reasonably well related with one-parameter velocity profile approximations. For submarine-size vehicles moving at typical speeds, however, the transition from laminar to turbulent flow within the boundary layer occurs near the bow and the boundary layer remains turbulent over the remainder of the vehicle. Unfortunately, the shape of the velocity profile for a turbulent boundary layer is not as easily resolved. Many different correlations and empirical relations have been put forth to allow closure of Equation (2.22) as it applies to turbulent flow; there are in fact over 50 different proposals in the literature [90].

One of the simplest approaches is to assume a profile shape consistent with empirical data. A surprisingly accurate approximation, first proposed by Prandtl [72] and recommended for general use by White [90], is the 1/7-power law:

$$\frac{u}{u_e} = \left(\frac{\hat{y}}{\delta_{99}}\right)^{1/7} \tag{2.28}$$

where δ_{99}, the boundary layer thickness, is the value of $\hat{y}$ at which there is a 1% difference between the boundary layer velocity and the freestream velocity.[15] Substituting Equation (2.28) into Equation (2.24) and replacing the upper limit of infinity on the integral with δ_{99} gives a necessary relationship for estimating the effect of turbulent wake fraction:

$$\delta_{99} = 8\delta^* \tag{2.29}$$

DPLL makes use of the power law to estimate the velocity profile of the boundary layer at the rotor. Upon convergence of the main duct/propeller/wake loop, inviscid surface velocities at all hull control points are known. DPLL sends the control point coordinates, their inviscid surface velocities and the global Reynolds number to MRCHBL, which returns the corresponding values of δ^* (among other parameters). The value of δ^* at the rotor is extracted and used to find the displacement thickness there using Equation (2.29). Assumption of the 1/7-power law profile then allows estimation of velocity reduction at any radius (i.e., any $\hat{y}$ value) on the lifting line.[16] This reduction is calculated at each lifting segment control point, located midway

[15] There is no well-defined interface between the boundary layer and the surrounding flow, as the viscous effects theoretically extend to infinity.

[16] The fact that the rotor lifting segments are always oriented normal to the local inflow and are therefore not necessarily exactly normal to the hull is not taken into account.

between the segment endpoints. Rotor torque, which during DPLL's main loop is calculated using inviscid velocities, is then updated using the reduced viscous velocities. Rotor thrust, which depends only on tangential velocities by Equation (2.3), is assumed to be unaffected by the hull boundary layer.

An approach such as this is clearly a low-order approximation; its results must be evaluated with an understanding of its limitations and assumptions. For example, it must be noted that the power law profile is most appropriate for internal pipe flows and for flat plates with zero pressure gradient. The local radius of hull curvature in the vicinity of the propellers is reasonably large even for full stern submarines, which makes the flat plate assumption tolerable. The pressure gradient in the vicinity, however, is generally non-zero and in fact may be considerable due to the combined influence of the hull, duct and propeller(s). Favorable pressure gradients ($dp/dx < 0$) tend to "fatten" the profile relative to the power law approximation, but so slightly that the effect may be safely ignored at this level of accuracy.[17] Adverse pressure gradients ($dp/dx > 0$), which are much more likely to be present near the stern of a submarine, have the opposite effect as shown in Figure 2-8. The normal-

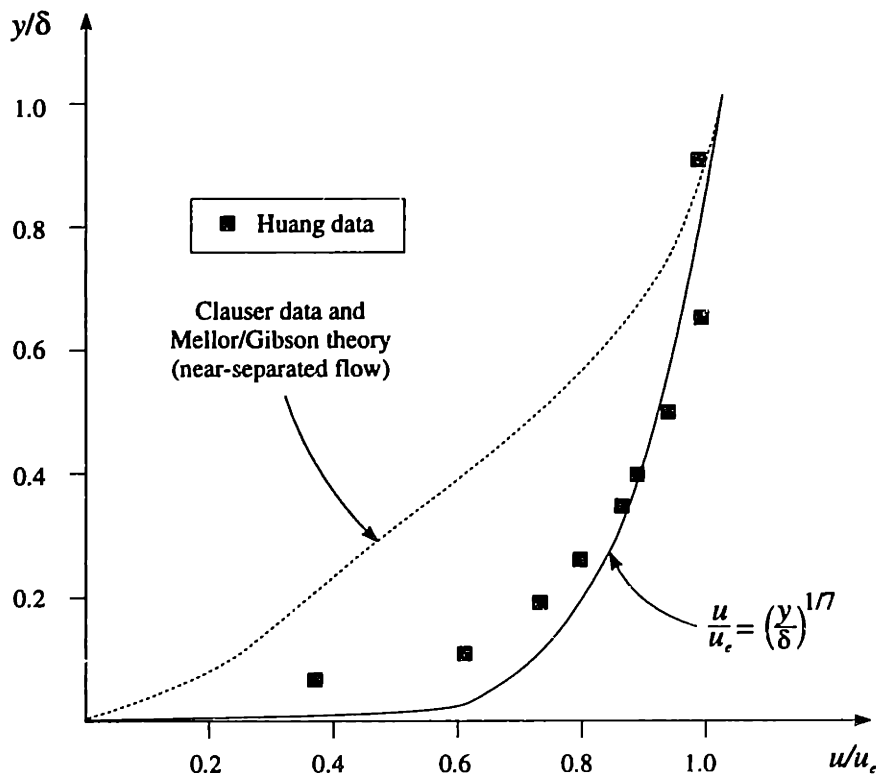

Figure 2-8: Turbulent boundary layer data and the power law approximation. Huang data is from the 1976 experiments, afterbody 3, $J = 1.07$ and $x/L = 0.915$.

[17] See White [90] Section 6-4 for quantification of these and the following comments.

ized 1/7-power law profile is shown along with two sets of data. The first of these, from Huang [48], is taken on a propelled axisymmetric full stern body immediately upstream of separation. A strong adverse pressure gradient exists at the data location. The power law under-estimates the velocity reduction in the lower portion of the profile, but the error introduced by the approximation is not severe. The other data is from a 1954 experiment by Clauser [14], shown as the dashed line in the figure (the profile is actually a prediction from the theory of Mellor and Gibson [64], but represents the Clauser data extremely well). This data is taken in the presence of *very* strong adverse pressure gradient, presumably bordering on separation. The power law approximation is obviously quite poor across the entire boundary layer in this case. What is important to note from the figure, though, is that the change in the boundary layer profile is not proportional to the magnitude of the pressure gradient. The power law approximation, which is very good in the absence of a pressure gradient, is only slightly less accurate in a strong gradient such as that associated with a full stern vehicle (e.g., the Huang data). Only on the brink of separation does the model become inappropriate, as illustrated by the Clauser data. It is reasonable to conclude, then, that the use of this approximation in DPLL provides acceptable accuracy in all but the most extreme cases. For configurations which border on separation, some of which will be investigated during the optimization phase of this research, the effect of wake fraction will be under-estimated but should at least allow accurate relative comparisons to be made.

Prediction of Flow Separation

The boundary layer integral momentum equation (2.22) is obtained by combining the two thin shear layer (TSL) equations and integrating the result across the boundary layer to solve for skin friction. The TSL equations are (1) continuity and (2) a first-order approximation of the x-directed Navier-Stokes equation as the Reynolds number goes to infinity. These two equations can also be combined and integrated so as to solve for dissipation, resulting in a second differential equation [21]:

$$\frac{d\theta^*}{d\hat{x}} = 2C_D - 3\frac{\theta^*}{u_e}\frac{du_e}{d\hat{x}} \tag{2.30}$$

where

$$\theta^* = \text{kinetic energy thickness} = \frac{1}{u_e^3}\int_0^\infty (u_e^2 - u_{\hat{x}}^2)u_{\hat{x}}\, d\hat{y} \tag{2.31}$$

$$D = \text{dissipation} = \int_0^\infty \tau(\hat{y}) \frac{\partial u_{\hat{x}}}{\partial \hat{y}} \, d\hat{y} \tag{2.32}$$

$$C_D = \text{dissipation coefficient} = \frac{D}{\rho u_e^3} \tag{2.33}$$

If Equation (2.30) is divided through by a new parameter H^*, defined as:

$$H^* = \text{kinetic energy shape factor} = \frac{\theta^*}{\theta} \tag{2.34}$$

and Equation (2.22) is subtracted from the result, the kinetic energy shape parameter relation is obtained:

$$\frac{\theta}{H^*} \frac{dH^*}{d\hat{x}} = \frac{2C_D}{H^*} - \frac{C_f}{2} + (H-1) \frac{\theta}{u_e} \frac{du_e}{d\hat{x}} \tag{2.35}$$

$$\therefore \quad \frac{dH}{d\hat{x}} = \frac{dH}{dH^*} \frac{H^*}{\theta} \left[\frac{2C_D}{H^*} - \frac{C_f}{2} + (H-1) \frac{\theta}{u_e} \frac{du_e}{d\hat{x}} \right] \tag{2.36}$$

which is normally used with Equation (2.22) in lieu of (2.30) in two-equation boundary layer methods [21]. Given the inviscid velocities $u_e(\hat{x})$ from DPLL's converged solution, MRCHBL solves these equations for H and θ using internal correlations for H^*, C_f, and C_D:

$$H^* = H^*(H)$$
$$C_f = C_f(H, \theta, \nu, u_e)$$
$$C_D = C_D(H, \theta, \nu, u_e)$$

If the calculated value of H exceeds a pre-specified turbulent limit—set to 2.7 in the DPLL implementation—separation is declared.[18] If separation occurs upstream of the rotor, the inviscid flow field and force coefficients calculated by DPLL are to some degree invalid. This situation would justify rejection of DPLL's solution when it is used in a stand-alone mode. In the optimization of DPLL described later, where relative rather than absolute criteria take priority, it is more efficient to simply penalize separated variants. This will be taken up in more detail in Chapter 4.

Like the wake fraction estimate of the previous section, the prediction of flow separation by this method provides consistent results and allows accurate relative comparisons among variants. It does not match full viscous flow solvers (such as RANS routines) in terms of absolute accuracy, but runs several orders of magnitude faster and is adequate for exploratory design. Validation of the routine's ability to predict separation and estimate wake fraction are included in Chapter 3.

[18] In this event, MRCHBL continues with the boundary layer calculations in "inverse" mode, extrapolating H values from points upstream and ignoring the input edge velocities. The invoking of inverse mode is taken to indicate separation in this research.

2.3.2 Hub Vortex Drag

The drag due to the hub vortex contributes to total drag on the hull. A hub vortex occurs when circulation at a blade root is shed onto the hub in accordance with Equation (2.6). This vorticity generally follows a hull streamline for some distance downstream before detaching, in contrast to tip vortices which leave the blade surface immediately. Figure 2-9 shows a hub vortex produced by a stator acting alone.

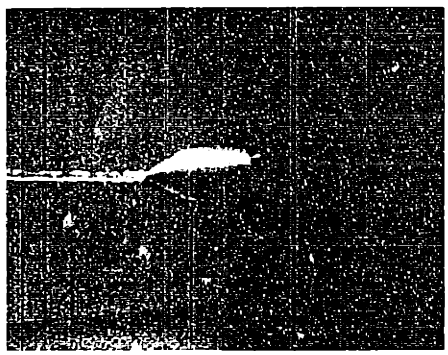

Figure 2-9: Hub vortex downstream of a stator.

Wang [87] determined that a Rankine vortex structure is an appropriate model for a hub vortex. The velocity and circulation of a Rankine vortex of radius r_0 are given by:

$$\bar{u}_t(r) = \begin{cases} \frac{\Gamma_0}{2\pi r_0^2} r & \text{if } r \leq r_0 \\ \frac{\Gamma_0}{2\pi r} & \text{otherwise} \end{cases} \tag{2.37}$$

$$\Gamma(r) = \begin{cases} \frac{\Gamma_0}{r_0^2} r^2 & \text{if } r \leq r_0 \\ \Gamma_0 & \text{otherwise} \end{cases} \tag{2.38}$$

and Coney [15] gives the drag force due to the vortex as:

$$F_h = \frac{\rho}{16\pi} \left(\ln \frac{r_h}{r_0} + 3 \right) \Gamma_0^2 \tag{2.39}$$

where r_h is the radius of the hub and Γ_0 is the circulation on the blade at the hub. In DPLL's hub vortex model, Γ_0 is taken as that of the innermost lifting segment. When multiple stages are present, it is the algebraic sum of the innermost segments of all stages. The value of r_0 is empirical; a reasonable value is one-quarter the radius of the final (most downstream) propeller hub.

Inviscid body drag in DPLL is determined by summing pressure forces over body panels. When the point of departure of the hub vortex is reached (where hull radius $\rightarrow r_0$), the summation is terminated and ambient pressure is applied to an imaginary panel normal to the z-axis at that point. The force due to the hub vortex from Equation (2.39) is then added to the truncated pressure summation to result in the total inviscid drag.

2.3.3 Self-propulsion

DPLL provides the option of floating the advance coefficient of the rotor, allowing the program to search for the rotor rotation rate which results in a self-propelled

condition—where the sum of the thrusts produced by all components of the propulsor balances all drags acting on the vehicle.[19] If two consecutive iterations of DPLL's main loop result in an over- or under-propelled condition, the advance coefficient is adjusted accordingly and the iterations continue until total axial force on the vehicle is within some tolerance of zero.[20] All other convergence criteria (see Section 2.3.5) remain in effect. Convergence is disallowed during the iteration immediately following an adjustment of advance coefficient in order to allow settling time.

2.3.4 Cavitation Index

DPLL performs a simple, parametric estimation of the rotor's cavitation index. Cavitation occurs when fluid pressure is reduced below its vapor pressure and therefore changes abruptly from a liquid to a gas (fluids are essentially incompressible, but cannot withstand any significant tension [67]). This is a fairly common phenomenon on the low pressure (suction) side of propeller blades and at blade tips; it is often accompanied by increased propeller noise due to the eventual collapse of the cavitation bubbles. Cavitation can also result in erosion of blade surfaces [30]. The method used by DPLL to relate rotor operating parameters to a cavitation index is adapted from a correlation by Farell and Billet [25] and detailed in Appendix A. The cavitation index defined there is simply $-C_{p_{\min}}$, the negative of the minimum pressure coefficient in the flow. This minimum is assumed to occur at the blade tips, where the clearance flow between the tips and the duct interacts with the primary through-flow. Since the pressure coefficient for pressures lower than the ambient is negative, larger values of this cavitation index equate to increased likelihood and severity of cavitation.

2.3.5 Convergence Criteria

The main loop of the code is tested for convergence during each iteration. Simultaneous satisfaction of *all* criteria terminates the iterations and sends the program to the boundary layer and post-processing routines. The convergence criteria are summarized below.

Goldstein factors

New generalized Goldstein factors are calculated for all lifting segments during each iteration. These values are then subjected to several constraints intended to prevent

[19] An alternative would be to float the circulation distribution on the blades, but this would be a much more complicated process when multiple stages are present.

[20] Consecutive iterations must *both* be either under- or over-propelled to trigger an adjustment. This prevents time-consuming oscillations about the self-propelled point.

unstable oscillations. The constraints place absolute as well as dynamic limits on the new factors, based on their values from the two previous iterations. If the newly calculated factors must be constrained, the constrained values are used for the remainder of the iteration but the initial *unrestricted* values are used for convergence checking. This convergence criterion is satisfied only when the unrestricted values of all segments are within some tolerance of the previous iteration's values. For this research, the convergence criterion is a change of less than 2% in any Goldstein factor from one iteration to the next.

Propeller induction on body

Convergence of velocities in the wake is assumed to be directly related to the change in propeller induction at hull control points during consecutive iterations. These changes require adjustment of body source ring strengths so as to maintain zero normal velocity at the hull, and therefore indirectly affect the entire flow system. It is necessary that these induced velocities become relatively constant in order to declare convergence. For this research, the criterion for maximum change in propeller induction at any body control point from one iteration to the next is 1% of reference velocity (vehicle speed, V_s).

Duct load

When DPLL is run in design mode, the duct load is analyzed at the beginning of each iteration by summing the non-dimensional strengths of all duct vortex rings. Since wake streamlines will have changed since the previous duct design, the analyzed load will generally differ from the design load even if the previous duct design was successful. The program proceeds with a new duct design, which modifies the camber line and angle of attack to re-acquire the input load. When the iterations have progressed to the point where the duct's post-design load is within some tolerance of the pre-design analyzed load, this convergence criterion is satisfied. This essentially corresponds to stabilization of the velocity field in the vicinity of the duct. The tolerance used in this research is a change in non-dimensional circulation (G_{duct}) of less than 0.001. This is at most 0.25% of the input target for the range of loads used here.

2.3.6 Duct Design Failure

The duct design process adjusts the *magnitude* of the duct camber (not the basic *form*, which is input; see Section 2.2.5) and the angle of attack to result in the specified

load given the current local velocity field. The process may fail to converge for either of two reasons: large variation in the velocity field from one iteration to the next (causing the design process to produce a camber line which intersects the hull profile; this is more common in the early stages) or numerically valid but physically infeasible input parameters.

Damped Convergence

Failure of the duct design process is usually a result of the duct leading edge encountering the body profile. If the duct design is having difficulty converging, DPLL will attempt to relax the design constraints until a valid solution is obtained. This is done by temporarily resetting the target (input) duct load to a value closer to its latest analyzed load. If the duct design then converges, the wake calculations proceed using the relaxed conditions and the original input load is attempted again during the next iteration, when the required camber line shift will presumably be less radical. Since the velocity field should settle as the iterations proceed, this allows "damped" convergence of some configurations which would otherwise fail.

Invalid Configurations

A valid solution does not necessarily exist for the duct design process. The unique solution for a particular duct load, propeller tip radius and velocity field may place the leading or trailing edge of the duct within the body envelope. This can be an unforeseen consequence of apparently valid input parameters.

The version of DPLL used for the optimization phase of this research is slightly modified to handle this situation. It will attempt to increase the radius of the duct if the relaxation routine described above is unsuccessful. This involves a permanent change to the input governing propeller tip radius. Fortunately, this is not often necessary. When the situation does occur, it typically involves the combination of negative duct load, small propeller radius and a very full stern. Adjusting the duct radius is considered preferable to non-convergence (and certainly to run time errors) in these rare cases. The stand-alone version of DPLL v2.0 does not have this feature; it will simply fail when these conditions occur.

2.3.7 Normalization

All spatial values within DPLL are normalized to body length. The body is assumed to be axisymmetric on the horizontal z-axis and facing left; the bow and stern are at $z/L = 0.0$ and 1.0 respectively. A non-physical "sting" extends aft of the stern

through the ultimate wake to prevent infinite tangential velocities due to radial contraction.[21] This is simply a small radius continuation of the body profile and is not considered part of the body length for normalization purposes.

All velocities are normalized to the far field velocity, which is assumed to be purely axial and equal to the vehicle's forward speed. Non-dimensional coefficients are generally normalized using maximum body radius (or diameter, in the case of advance coefficient), which is itself normalized to body length.

[21] The ultimate wake location is user-specified. Downstream of this location, the wake and its streamlines are assumed to be of constant radius and streamline discretization terminates. Induced velocities due to the portion of the wake extending from this location to $+\infty$ are calculated analytically, using the expressions of Hough and Ordway [46].

Chapter 3

Validation of DPLL

This chapter documents the modeling of published experiments in DPLL and compares the resulting calculations with experimental data. Although the primary use of DPLL in this research will be for optimization—requiring accuracy only to the extent that *relative* comparisons can legitimately be made—the code is also intended for use as a stand-alone design tool. Some indication of its absolute accuracy is therefore needed as well.

The validations to follow begin with DPLL's calculation of thrust and torque, using well-known hydrodynamic experiments. These are followed by testing of the program's boundary layer routines, using published wind tunnel data. Sample flow field output and force calculations are then presented for a notional full stern submarine with a ducted multi-stage propeller, typical of those considered in the optimization phase of this research. This is included to qualitatively support the conclusion that DPLL results are reasonable and valid. Finally, simple convergence characteristics in terms of the number of body and duct control points are given.

Unless otherwise noted, quantities in this chapter are expressed using the nomenclature of the original experiments. This will usually require a conversion of DPLL's normal output values. A complete list of symbol and variable definitions and descriptions may be found beginning on page 15.

3.1 Thrust and Torque Calculations

3.1.1 Ka–4–55

A series of five screw propellers was used by van Manen at the Netherlands Ship Model Basin (NSMB) to investigate the effect of radial load distribution on the performance of ducted propellers [86]. These propellers were derived from the $P/D = 1.018$

variant of the K-4-55 series by varying the pitch distributions and profile sections.[1] The results of this investigation led to the development of a new series having one of these five propellers (denoted as screw B in the report) as its reference. The new series became known as Ka-4-55.

The five experimental propellers (A through E) were tested in combination with four different duct designs. Three of these designs, referred to as ducts 18, 19 and 20 in the report, differed only in the curvature of their outer surfaces and only by a small amount. There was no reported difference among them in terms of performance. The fourth design, referred to as duct 19A, was of similar shape but had a flat outer surface and a relatively thick trailing edge. The performance of 19A was slightly worse than that of the other ducts.

The report by van Manen documents the performance of these five propellers in duct 19 and the performance of the entire new Ka-4-55 series in both ducts 19 and 19A.[2] From among these possibilities, the particular configuration of propeller B in duct 19 is chosen here for modeling (Figure 3-1). This choice is driven by three

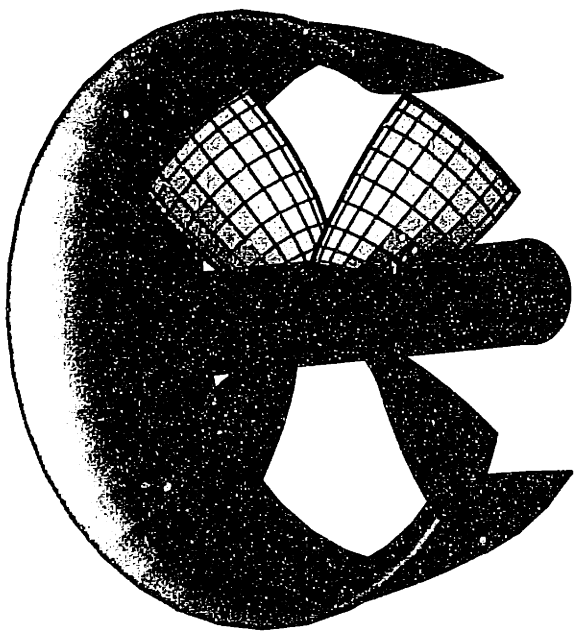

Figure 3-1: Ka-4-55 series reference propeller B with duct 19.

[1] Propeller series are designated by a group name, number of blades, and expanded area ratio (EAR; roughly the ratio of blade surface area to $\pi D^2/4$). The K-4-55 series, for example, is from the Kaplan (K) group, has four blades and an EAR of 0.55.

[2] The report also presents cavitation as a function of thrust, and efficiency loss as a function of tip clearance for the original five propellers. Unfortunately, this level of detail is not captured by the DPLL model.

factors. First, the blade circulation distribution of propeller B, a necessary input for the DPLL model which is not documented for any of the propellers, has been calculated by McHugh [63] and is readily available. Second, van Manen's data is presented in graphical form and the resolution of the A–E data, given at a single P/D value in duct 19 only, is higher than that of the full Ka–4–55 series data, which is given at six P/D values in both ducts 19 and 19A. Third, the Ka–4–55 series data is primarily intended to compare the performance of duct 19 with that of duct 19A. DPLL's simple camber line model cannot distinguish between these two ducts.

DPLL normally operates on input values of circulation at the propeller hub and tip, interpolating linearly to determine values at intermediate lifting segments. However, a linear approximation will not model the data well in this case, as can be seen in Figure 3-2. DPLL's source code is therefore temporarily modified so that circulation

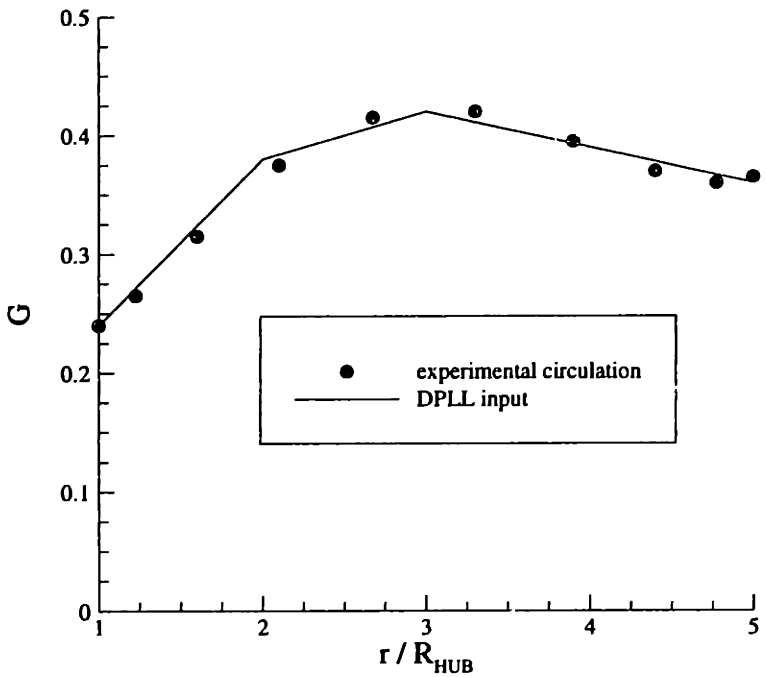

Figure 3-2: Ka–4–55 propeller B blade circulation vs. non-dimensional radius, as calculated and modeled. The DPLL model uses six discrete lifting segments.

at each lifting segment may be individually specified. Unfortunately, this modification requires the bypassing of Goldstein factor calculations (see Section 2.2.3). This is a limitation of the current version of the code.

The experimental circulation distribution shown is not explicitly provided in the report. Rather, it is taken from an independent calculation by McHugh using a

coupled DTNS/PBD-14 analysis [63] and the given experimental blade, hub and duct geometry. The DTNS (David Taylor Navier-Stokes) algorithm, developed at the David Taylor Model Basin, is a member of the class of routines known as Reynolds-averaged Navier-Stokes (RANS) viscous flow solvers. As the name implies, these use the full incompressible Navier-Stokes equations to solve for a system's velocity field. In a coupled approach such as that used by McHugh, the velocity field obtained from a RANS solver is routed to a propeller code—in this case PBD—which computes the resulting forces on the blades.[3] The forces are then fed back to the RANS code, which re-computes the velocity field accounting for the new reaction forces in the fluid. The process is iterated until some convergence criterion is met. In this case, the converged solution provided the experimental blade circulation shown in Figure 3-2.

The inflow velocity profile is not mentioned in the report, but is assumed here to be axial and uniform. To obtain this effect in DPLL, the hull of a long and slender pseudo-vehicle is used to represent the experimental hub and shaft. The lifting line blades and the camber line representing the duct are located at the axial midpoint of the hub-body, as shown in Figure 3-3.

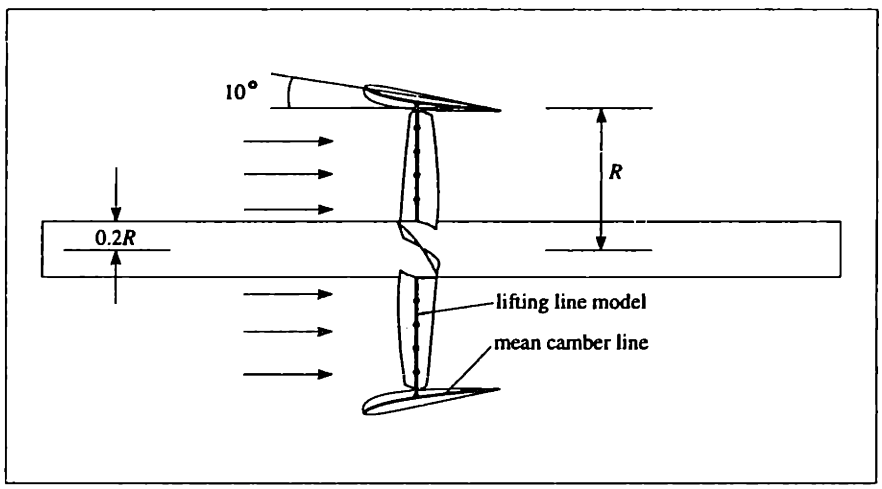

Figure 3-3: Graphical representation of Ka–4–55 as modeled in DPLL, showing use of long "body" to simulate the experimental hub and shaft and produce uniform inflow velocity at the rotor.

Determing the correct duct parameters for modeling the experiment is somewhat involved, because DPLL requires that the camber line be defined by B-spline vertices. The use of B-splines is usually beneficial in that it consistently results in smooth curves; unfortunately, it is difficult to accurately reproduce a given curve without tedious trial and error. Also, minor changes in camber line vertex coordinates tend to have noticeable effects on duct thrust in the DPLL model. For these reasons,

[3] The general method used by McHugh was documented earlier by Black [5, 6].

attempts to input the absolute geometry of the Ka–4–55 duct with any degree of accuracy are unsuccessful, and DPLL is instead run in design mode (described in Section 2.2.5). Vertices producing a camber form visibly similar to that of the experiment are used, but more importantly the duct *load* is now input and varied so as to produce the experimentally measured duct thrust upon convergence. Some trial and error is required during this process, in order to obtain a converged solution which matches the experimental duct thrust, camber line and angle of attack. The results of the best match are shown in Table 3.1. The accuracy of the duct thrust coefficient

Coefficient	Experimental	DPLL
$K_{T\text{duct}}$	0.1054	0.1007
$K_{T\text{total}}$	0.3411	0.3440
K_Q	0.0385	0.0370

Table 3.1: Measured and calculated force coefficients for Ka–4–55 reference propeller, at $J = 0.36$.

is irrelevant since it is intentionally matched. However, by matching experimental geometry and duct thrust, DPLL also calculates an *overall* thrust coefficient (due to duct and rotor) very near the measured value. Calculated rotor torque is within 5% of measured, but proves to be quite sensitive to small variations in the duct input parameters. Seemingly insignificant changes in the form of the input duct camber line can alter the calculated K_Q by 5% or more. Calculated propeller thrust, on the other hand, is relatively unaffected by small variations in duct parameters.

3.1.2 HIREP

Zierke et al. used the High Reynolds Number Pump (HIREP) facility to investigate forces and flows in a multiple blade row machine [95]. This facility was installed in the test section of the Garfield Thomas water tunnel, located at Pennsylvania State University. A schematic of the experimental configuration is shown in Figure 3-4. The experiments were undertaken to provide a high Reynolds number database for the validation of numerical codes. This was motivated by the relative scarcity of such information, particularly for incompressible flows, and the numerous analysis codes under development.

The design advance coefficient of the experimental setup was $J = 2.31$, as normalized to the rotor diameter. Tests were conducted at this and two bracketing values. Thrust and torque measurements were taken on the pump rotor; velocities and pressures were measured in the region of the pump rotor and stator. Measured velocities were the source for blade circulation distributions given in the report and used here.

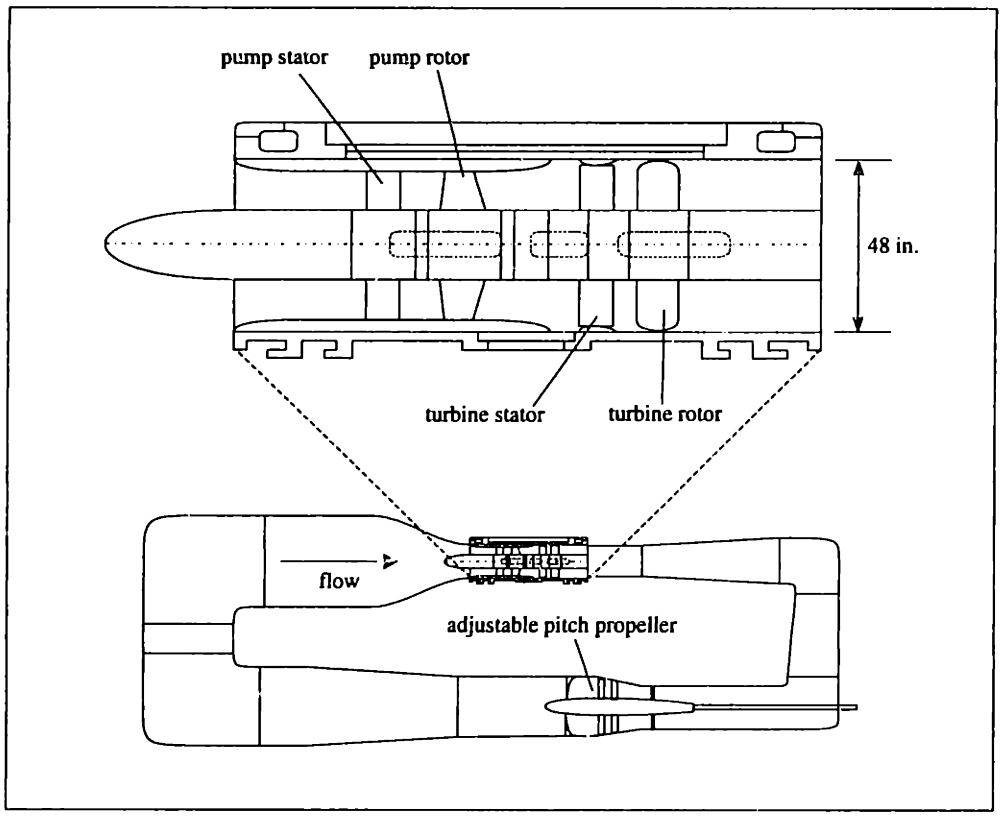

Figure 3-4: HIREP experimental setup, showing the pump facility as installed in the Garfield Thomas water tunnel.

The HIREP system is modeled in DPLL using a long, constant radius body to represent the hub in a similar manner to that used for the Ka–4–55 validation (see Figure 3-3, page 64). In this case, however, a horizontal duct camber line extending well forward and aft of the blades is used to represent the tunnel inner wall and DPLL is run in analysis mode in order to maintain its position. Only the pump stator and rotor are modeled in DPLL as propeller stages; the turbine stator and rotor, which served to turn the pump rotor in the experiments, are not considered here.

As in the Ka–4–55 validation, blade circulation is forced to a known experimental distribution. The measured circulation on the pump rotor and stator blades is approximated by six discrete values in DPLL, applied at the control points of the six segments making up each lifting line (see Figure 3-5). Note the attempt to average the measured circulation drop at the tip over the outermost lifting segment in the model. This is a crude way of approximating tip gap effects using DPLL's small-gap model.

A summary of the measured, design and calculated results at the design advance coefficient is given in Table 3.2. Measured and design torque were very similar; the

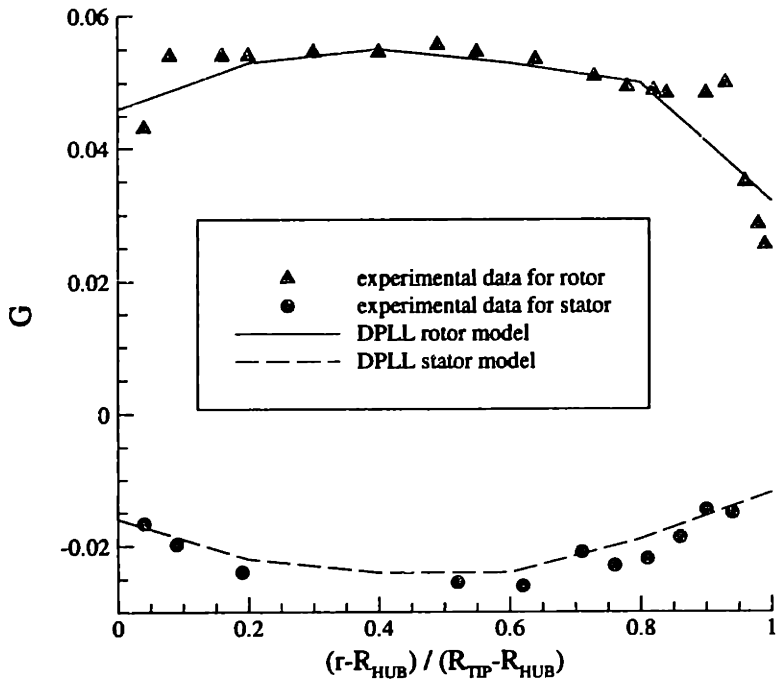

Figure 3-5: HIREP experimentally measured circulation (via laser Doppler velocimeter) and its approximation for input to DPLL.

Coefficient	Measured	DPLL	Design Point
K_Q	0.310	0.335	0.316
K_T	0.890	0.806	0.780

Table 3.2: Summary of measured, calculated and design force coefficients for HIREP at design advance coefficient ($J = 2.31$). Experimental problems were encountered in measuring thrust.

DPLL torque calculation is 8% greater than the measured value and 6% greater than the design value. Measured rotor thrust was 14% greater than the design value. This experimental discrepancy is not fully explained in the report, although it is mentioned that problems were encountered in performing thrust measurements [95]. A sampling window smaller than the period of shaft rotation and a pressure differential across the hub, which may not have been included in the measurements, are cited as possible causes. In any case, the thrust calculated by DPLL is 3% greater than the HIREP design thrust and within 10% of measured thrust, falling between the two values. If the reported measurement problems did in fact result in inaccurate readings (as the text of the report seems to indicate), then design thrust must be taken as the more accurate. The small difference between measured and design torque lends support to this assumption. In that case, DPLL's torque and thrust calculations are both quite

close to the actual experimental data.

3.2 Boundary Layer Modeling

3.2.1 Shear Stress and Separation

Huang et al. performed wind tunnel tests on three axisymmetric bodies having identical forward offsets but different stern geometries [48]. The objective of these tests was to obtain boundary layer characteristics and pressure distributions for comparison with theoretical calculations. The data obtained have since been widely used for validation of hydrodynamic and aerodynamic analysis codes. An example of the experimental models is shown in Figure 3-6 (note that in this and the remaining figures of this chapter, the axial coordinate is x rather than z in accordance with the Huang reports). The models utilized a common nose cone but different mid-body and stern

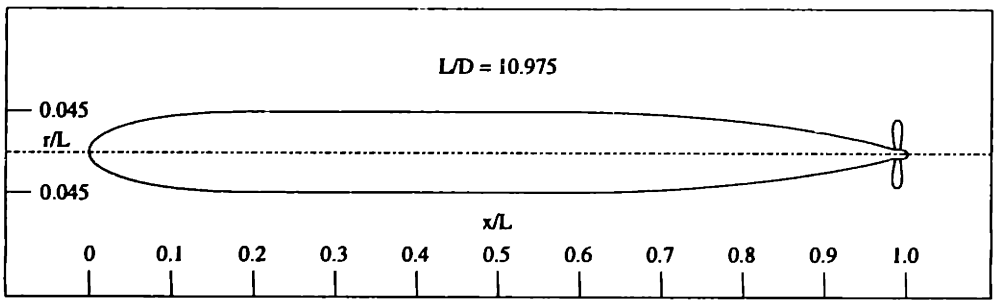

Figure 3-6: An example of the slender bodies used by Huang. Taken from Huang et al., 1976 [48].

sections. The stern sections, shown overlaid in Figure 3-7, varied in their rate of taper and were all of different lengths. Overall model length (3.066 m) was kept constant by changing the length of the parallel mid-body section as necessary. The tests relevant to the current research were performed at a Reynolds number of 5.9×10^6, or air speeds of 28.9 m/s, assuming air viscosity of 1.5×10^{-5} m^2/s. Therefore the data and calculations to follow should correspond, by a similitude argument, to a full-size submarine at a bare crawl or a 3 m autonomous underwater vehicle (AUV) at about 2 m/s, or 4 knots.

The stern section designated as afterbody 1 was slightly tapered and designed for a smooth transition of flow to the wake. Afterbody 2 was shorter with more abrupt taper and was intended to verge on separation under the experimental conditions. Both afterbodies 1 and 2 had profiles defined by polynomials. Afterbody 3 was the shortest and most abrupt of the three. Unlike the others, its profile was defined by a cosine curve; it included an inflection point aft of the shoulder and was intended

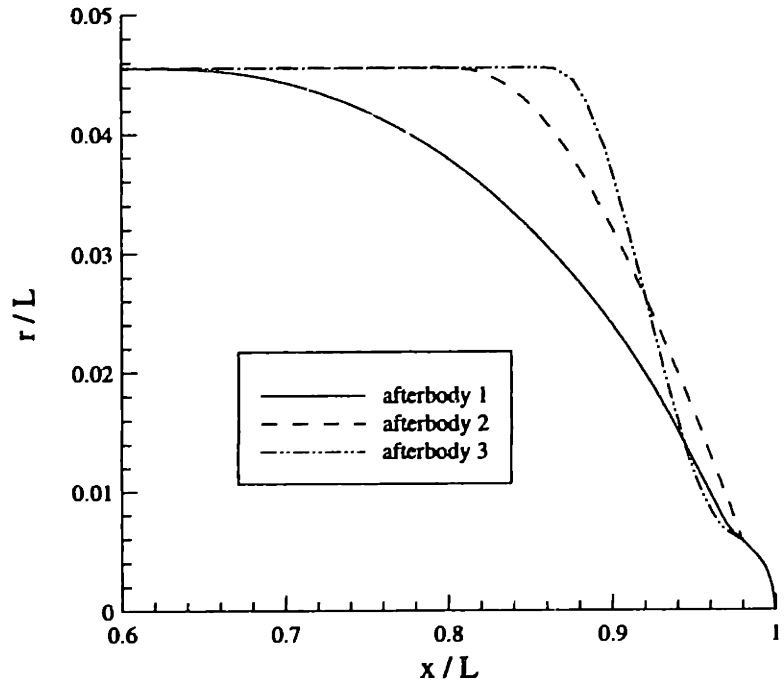

Figure 3-7: Afterbody geometries from Huang experiments (1976). Note the axis scaling.

to cause flow separation. All three afterbodies were adapted to accept a common propeller hub, which began at $x/L \approx 0.97$. Shear stresses were measured on the body surfaces with and without a propeller installed and operating.[4]

This experimental setup is modeled in DPLL by developing B-spline vertex files to result in the three profiles and supplying the correct Reynolds number for the boundary layer calculations. For the propelled body tests, the propeller geometry and circulation given in the report are input also. Again, the circulation distributions are forced and Goldstein calculations are therefore bypassed.

These experiments did not involve ducted propellers. Since DPLL requires a duct if a propeller is present (propeller tip locations in the program are specified as a fraction of duct chord), the propelled tests are approximated by using a very short, zero load duct. This is intended to represent a non-ducted configuration as accurately as possible.

In all Huang tests, a tripwire was used to induce the transition from laminar to turbulent flow at $x/L = 0.05$. This effect is easily modeled, as MRCHBL allows the input of a manual trip location. As the routine begins marching downstream from

[4]The propeller was driven by a 9-kW high-speed motor mounted inside the stern sections.

the bow of the vehicle, it continually checks for both the location of the manual trip and the first occurrence of conditions which would induce a natural transition to turbulence. If either is encountered, turbulent equations are substituted for laminar and are used thereafter. Thus an input trip location takes precedence if it is located prior to the point where natural transition would occur; otherwise, the trip location is irrelevant. Some test runs with the DPLL model indicate that the natural transition location for the Huang experiments would have been just downstream of the trip. Thus the manual trip takes precedence and natural transition plays no part in the experimental data or the calculations documented here.

Shear stress data taken from the Huang report is compared to the corresponding DPLL calculations in Figures 3-8, 3-9 and 3-10. For afterbodies 1 and 2, differences in shear distribution between the propelled and unpropelled conditions are noticeable in both the data and the calculations (the increase of shear stress on the hull due to propeller operation is a well-documented phenomenon known as the thrust deduction). Note that for both of these afterbodies, the propeller effect extends upstream to approximately $x/L = 0.9$, a distance of more than three propeller radii (the propeller has a radius of $r/L = 0.03$ and is located at $x/L = 0.983$ on all three bodies). This observation supports the assumption that propeller suction can allow some control over the boundary layer. Also note that this effect is due to an open propeller; a ducted propeller might be capable of producing a more pronounced effect.

Huang reported that no measurable difference existed between the propelled and unpropelled stresses on afterbody 3. As Figure 3-10 shows, the experiment resulted in flow separation at $x/L = 0.92$ (separation is indicated by shear stress dropping to zero). The shear distribution calculated by MRCHBL matches the data quite well, and inverse mode is invoked (thus predicting separation in the model) at $x/L = 0.93$.[5] Since boundary layer calculations are done only at body control point locations, which in this case are spaced at intervals of approximately $L/100$, MRCHBL actually predicts separation between 0.92 and 0.93.

The stress spikes in the calculations for afterbodies 2 and 3 at $x/L = 0.84$ and 0.87 respectively are in agreement with experimental observations. On page 35 of Huang's report, Figure 8 shows measured pressure distributions on the three afterbodies [48]. In all three cases, the form of the measured pressure distribution is very similar to the form of the stress calculations by DPLL, including the spikes on afterbodies 2 and 3. Unfortunately, these features are not reflected in the data due to low resolution. The experimental shear distributions of Figures 3-9 and 3-10 taken alone are therefore

[5] Shape factor, rather than shear stress, is the criterion used in this research to predict separation but the two indicators are strongly correlated (see Section 2.3.1).

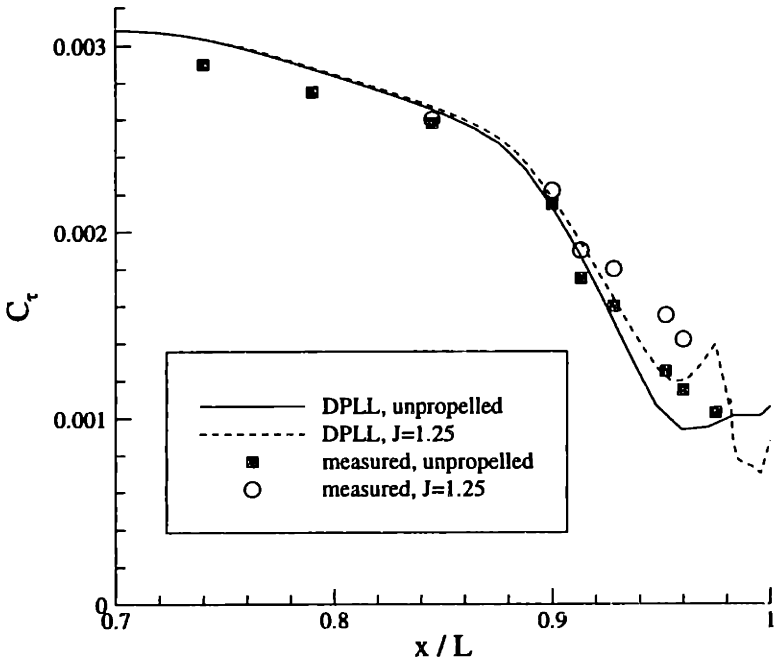

Figure 3-8: Measured and calculated shear stress on Huang afterbody 1 (1976).

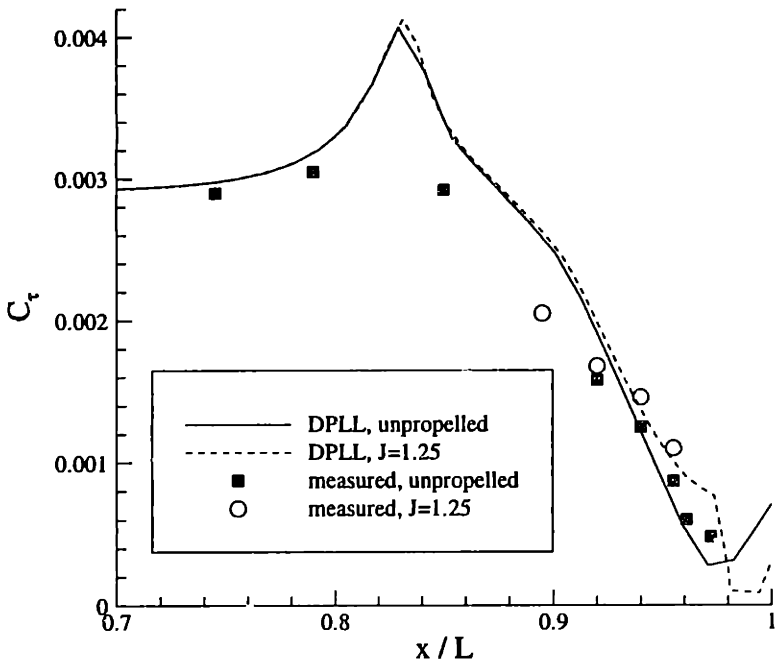

Figure 3-9: Measured and calculated shear stress on Huang afterbody 2 (1976).

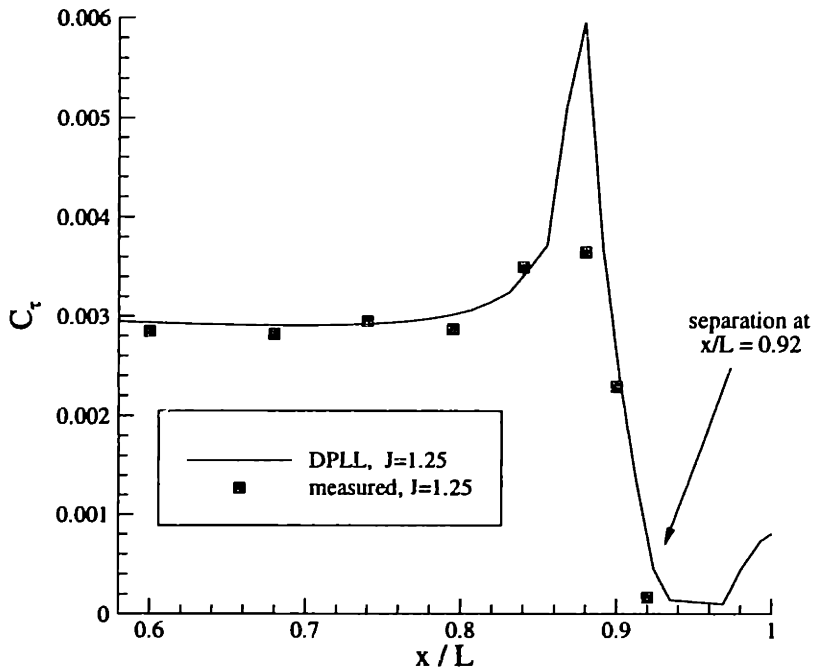

Figure 3-10: Measured and calculated shear stress on Huang afterbody 3 (1976).

somewhat misleading, in that they do not capture the full magnitude of the pressure rise.

Overall, the correlation between measured and calculated boundary layer behavior is fair for all three bodies in terms of absolute accuracy. The range of propeller influence is captured quite well, judging by the location at which the propelled stress diverges from the unpropelled stress in the data and in the DPLL calculations. Also, the relative differences between propelled and unpropelled stresses as measured and calculated are quite similar. Of particular importance to this research is the fact that DPLL correctly predicts the occurrence of flow separation on afterbody 3. The calculated location of separation also appears to be reasonably accurate.

The experiments on afterbody 3 present a situation very relevant to this research, namely a configuration which results in separated flow under the influence of an open propeller. Of particular interest is whether surrounding the propeller with a duct can affect separation in the DPLL model. To investigate, a duct with significant load and arbitrary camber is added to the model of the propelled afterbody 3. All other inputs remain unchanged. The final geometry of the converged solution (using design mode) is shown in Figure 3-11. The resulting shear distribution is shown in Figure 3-12,

overlaying the non-ducted calculations and data. Shear on the notional ducted system still decreases rapidly near the stern but no longer reaches zero; separation appears to be averted by the addition of the duct. Shape factor calculations confirm the absence of separation; that is, the shape factor does not exceed the prescribed turbulent limit of 2.7 at any location forward of the rotor. These results support the theory that ducts may be useful in avoiding separation on full stern submarines.

3.2.2 Displacement Thickness

Huang et al. later performed another set of wind tunnel tests on afterbodies 1 and 2 [47]. The purpose was to measure pressure and velocity distributions across the boundary layer, allowing calculation of displacement and momentum thicknesses. These tests, like the previous ones, were performed at a Reynolds number of approximately 5.9×10^6. Afterbody 3, which was known to cause separation at this speed, was not tested.

Turbulence was again artificially induced at $x/L = 0.05$. A complicated argument in the report seems to indicate that this trip location should actually be modeled as having occurred at $x/L = 0.015$. This latter value is used in the DPLL model; the difference this location makes in the results is insignificant. Figures 3-13 and 3-14 compare the displacement thicknesses (δ^*) measured by Huang to the results of MRCHBL for afterbodies 1 and 2 respectively. The calculations are quite accurate in both cases, the worst discrepancy being a slight over-estimate very near the stern of afterbody 2. The near-separated flow conditions on this model may be partially responsible for the inaccuracy; MRCHBL solutions generally become very sensitive near separation.

3.3 Sample Output

The experiments modeled in the previous sections, while useful for purposes of validation, provide little *qualitative* information regarding DPLL's performance in its intended role of submarine design. The HIREP and Ka–4–55 experiments, for example, involve isolated propellers and lack the effect of hull interaction. Both of the Huang experiments are at a very low Reynolds number relative to typical submarines and the propellers are not compound or ducted.

This section is intended to lend qualitative support to DPLL's validity, by documenting the analysis of a notional full stern submarine with a ducted, multi-stage propeller. The inputs used for this run are shown in Table 3.3. The duct is slightly

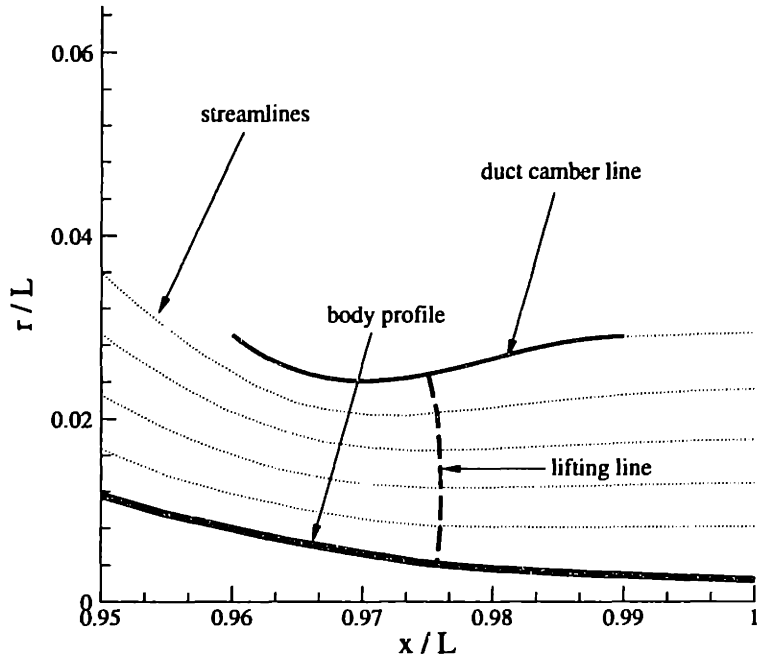

Figure 3-11: Duct and wake geometry of notional propulsion system, from DPLL output files.

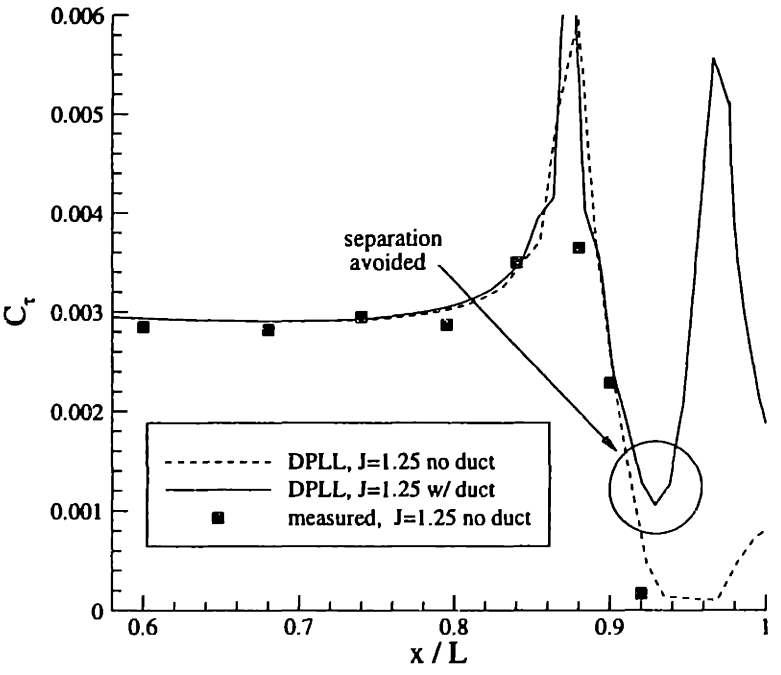

Figure 3-12: Shear stress on Huang afterbody 3 with notional ducted propeller.

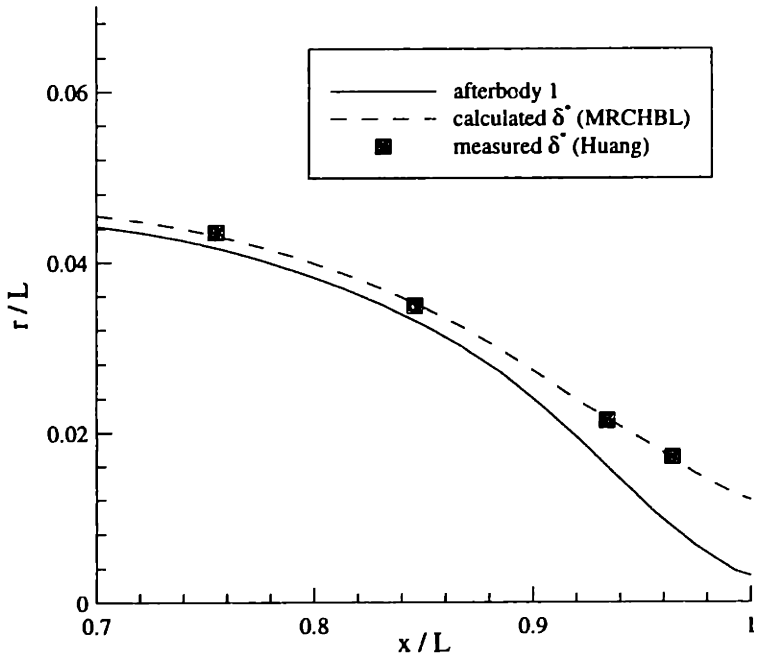

Figure 3-13: Measured and calculated displacement thickness on Huang afterbody 1 (1979).

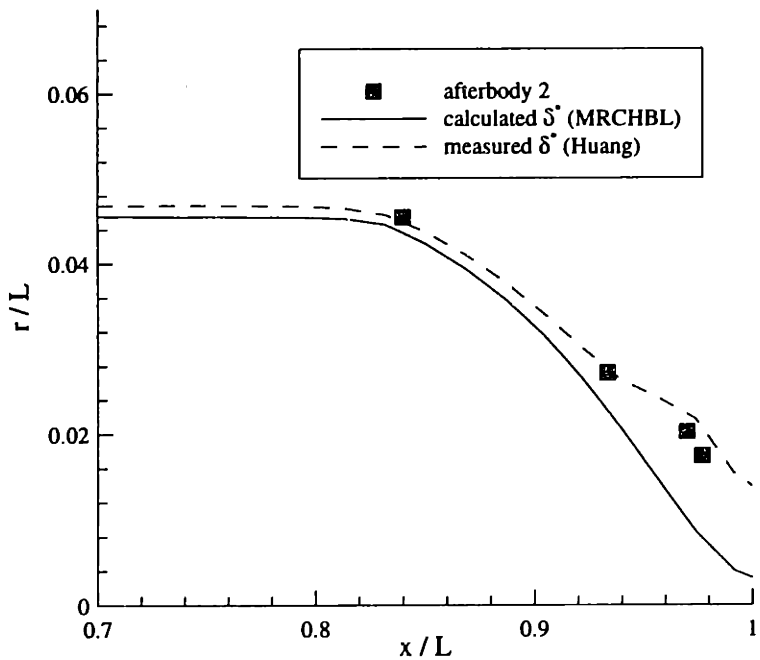

Figure 3-14: Measured and calculated displacement thickness on Huang afterbody 2 (1979).

Input Parameters			
duct load (G)	-0.020	rotor tip radius ($\%L$)	0.020
duct/rotor intersect (% duct chord)	0.80	duct chord ($\%L$)	0.069
rotor tip circulation (H)	0.024	stator tip circulation (H)	-0.035
rotor hub circulation (H)	0.021	stator hub circulation (H)	-0.031
Reynolds number	1.0×10^9	stator/rotor blades	5/5
ultimate wake location ($\%L$)	1.10	lifting segments per line	5

Table 3.3: Summary of DPLL inputs for notional submarine.

decelerative, meaning that the axial component of its lift should contribute to drag rather than thrust. A stator is employed forward of the rotor to provide pre-swirl; like the duct, it should contribute to viscous and non-viscous drag. The specified Reynolds number for this run is 1.0×10^9, which corresponds to an 80 m submarine traveling at 12.5 m/s, or about 24 knots.[6]

Calculated force and torque coefficients are listed in Table 3.4 (recall that forces in

Force Coefficients		Propeller Coefficients	
body pressure force	0.036	stator torque	-0.049
body viscous force	-0.150	stator thrust	-0.040
duct axial lift	-0.045	rotor torque	0.035
duct viscous force	-0.013	rotor thrust	0.221
hub vortex drag	-0.002	final rotor speed (J_B)	1.75
total drag	-0.172		
total thrust	0.181		
total force	0.007		

Table 3.4: Force and propeller coefficients for notional submarine.

DPLL are normalized by $\frac{1}{2}\rho V_s^2 A_B$, torques by $\frac{1}{2}\rho V_s^2 A_B D_B$). A positive force coefficient indicates a force on the vehicle in the $-z$ direction; a positive torque coefficient indicates a force on a component of the vehicle in the $+\theta$ direction. Body pressure force is small, but non-zero due to the induction of the duct and propeller on the stern. The duct contributes slightly to non-viscous drag, as expected due to its small negative load. Likewise, the stator produces a small negative thrust as a consequence of imparting pre-swirl to the rotor inflow. The force due to the hub vortex is negligible, as much of the vorticity shed onto the hub by the stator is removed by the rotor. The net force on the vehicle is not precisely zero; this is simply due to the tolerance used for satisfaction of the self-propelled criterion. The final advance coefficient, having been automatically adjusted to achieve self-propulsion, is 1.75 as normalized to maximum body diameter. This corresponds to a rotor rotation rate of about 130 rpm,

[6] The configuration modeled in this section is one of the frontier variants from Figure 5-2. It does not correspond to any particular full-size or model submarine.

given the forward speed of 12.5 m/s and assuming a full-size body diameter of 10 m.

Figure 3-15 shows the tangential and axial velocity due to propeller induction alone. The flow region affected by the propeller is bounded above by the duct stream-

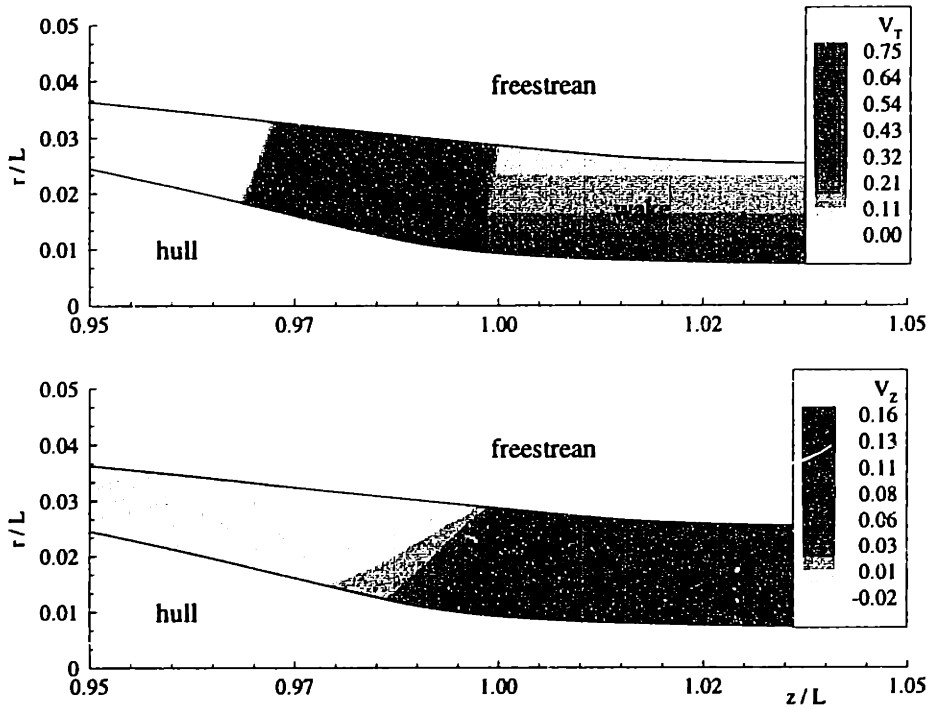

Figure 3-15: Propeller-induced tangential velocity (top) and axial velocity (bottom) for notional submarine.

line and below by the hull streamline. The locations of the lifting lines representing the stator and rotor are clearly visible in the upper tangential velocity plot. The pre-swirl induced by the stator is quite apparent; it increases as the hull contracts between the stator and the rotor. Not all of this velocity is removed by the rotor. This is not necessarily non-optimal in general, as it is possible that viscous losses incurred in removing all tangential velocity from the wake are greater that the realizable gains [88]. In the lower plot, only the location of the thrust-producing rotor is apparent since the stator induces no axial velocity on the flow.[7] It is interesting to note that the suction effect of the rotor extends upstream a distance of approximately two propeller radii, a result which agrees with the data and calculations of Section 3.2.1.

Figure 3-16 shows a schematic of the configuration along with *total* axial and radial velocity in the vicinity of the propulsor, where the effects of the body, duct and

[7]The x-component of the *total* velocity vector certainly changes when the flow encounters the stator, but the *prop-induced* component of axial velocity does not.

inflow (forward speed) have been included. The decrease in axial velocity forward of the rotor in this plot is quite pronounced, especially near the hull, as the total velocity vector becomes more radial there due to hull contraction. The location of the duct is clearly visible in the axial velocity plot, and it is obvious from the velocity jump across the camber line that the duct is subject to a force directed away from the hull. The axial component of this force, from the orientation of the camber line, appears to be directed toward the right (the $+z$ direction, thus contributing to drag). This is verified by the calculated duct force coefficient. The radial velocity plot shows the effect of the duct's slight negative load in "squeezing" the flow between the leading edge and the hull. There is little change in radial velocity across the camber line, as normal velocity across the camber line is zero and the angle of attack is small.

3.4 Convergence Properties

The full stern submarine of the preceding section is also tested for convergence by varying the resolution of the input geometry. During convergence testing, a self-propelled condition is not pursued—this prevents inconsistencies in final values due to tolerance in the force balance criterion. Instead, the program is simply run until calculated values are invariant from one iteration to the next. The fact that this results in a non-zero force balance for all tests is not relevant; the goal is to determine the minimum resolution at which calculations are reliable.

Calculations appear to be insensitive to wake discretization. DPLL allows specification of four wake discretization parameters. Three are axial—the number of wake stations forward of the duct leading edge, the number between propeller stages, and the number downstream of the duct trailing edge. The fourth is radial—the number of lifting line segments (and thus the number of streamlines in the wake). None of these variables appear to have a significant effect on the solution, even when they approach the minimum values allowed by DPLL's input-checking routines.[8]

Likewise, the solution appears to be unaffected by the location of the ultimate wake as long as it remains downstream of a critical value. Advancing the location of the ultimate wake upstream from a baseline location of $z/L = 1.10$ produces very little change until the point $z/L = 1.02$ is reached. Any location closer to the stern than this causes the program to fail. That such a close location allows accurate results is not entirely surprising; as seen in Figure 3-16 the wake streamlines of this

[8] Valid results are obtained for this variant with as few as 5 wake stations forward, 5 between and 10 downstream of the propellers. The number of lifting line segments has very little effect at values greater than 5, which is the minimum number allowed by DPLL.

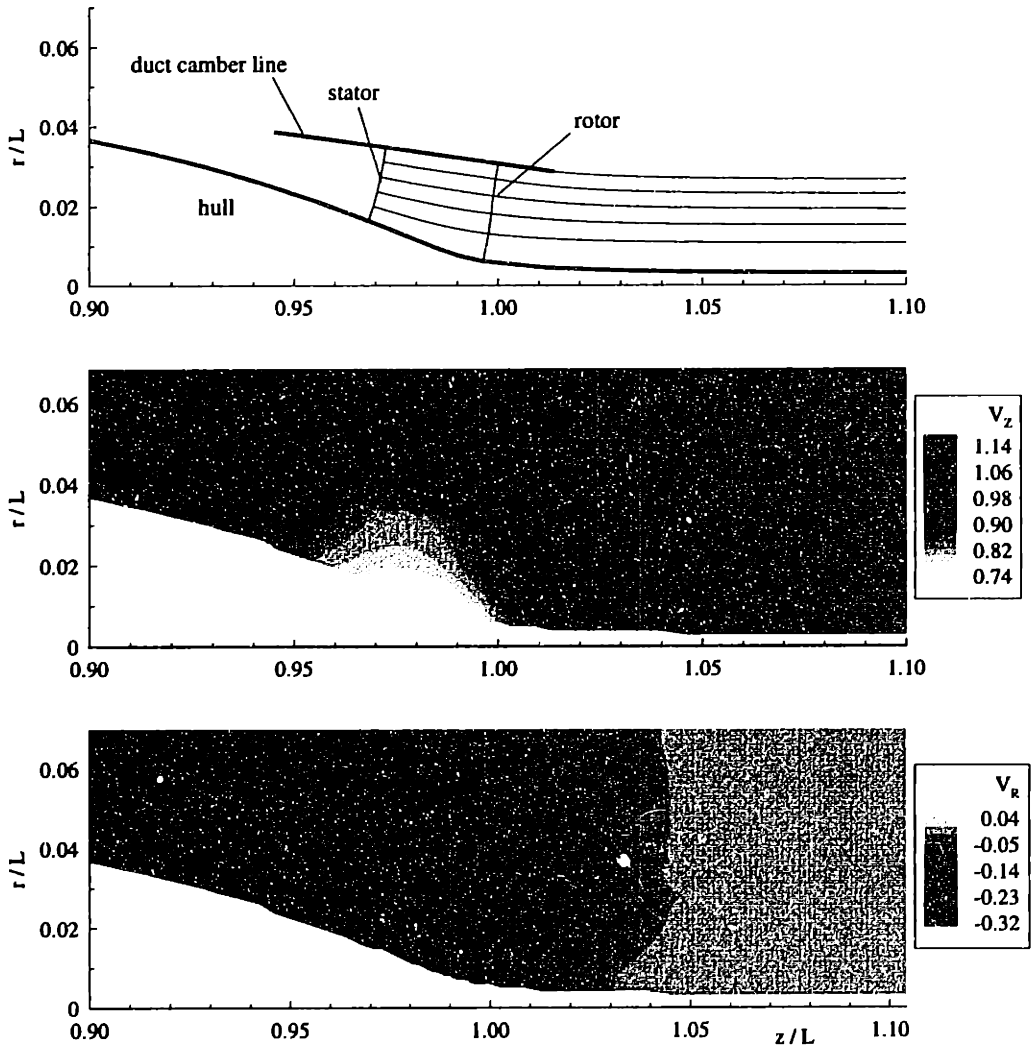

Figure 3-16: Total flow field for notional submarine. The upper schematic shows the locations of the hull surface, duct camber line, lifting lines for the stator and rotor and their wake streamlines. The middle plot shows total axial velocity in the same region; the lower plot shows the corresponding total radial velocity.

configuration attain constant radii very soon after clearing the duct trailing edge.

Discretization of body and duct control points—and therefore the source and vortex rings representing them—does have a significant effect on the solution. Figure 3-17 shows the body pressure force and the rotor thrust as functions of body and duct resolution. The total number of control points on the body, like the stations in the

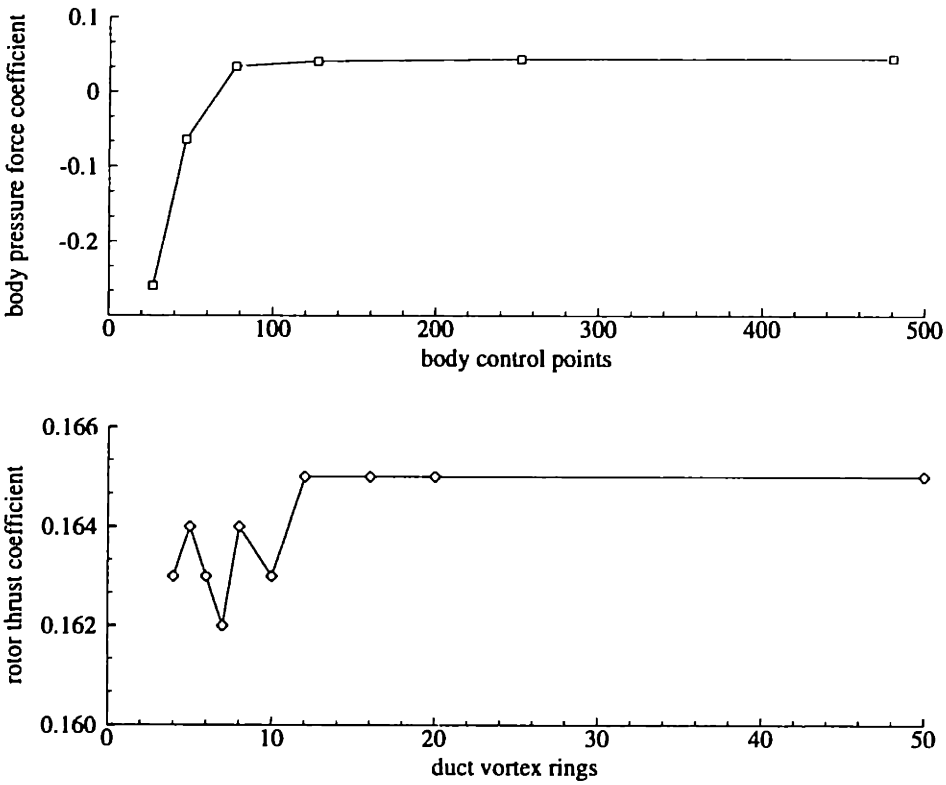

Figure 3-17: Convergence of body pressure force and rotor thrust, as functions of body control points and duct vortex rings. Rotor thrust is calculated to three digit precision.

wake, must be specified by three values—the number forward of the duct, between propeller stages, and downstream of the duct. In these tests, each of these values is increased in proportion with the total number of points on the body. Fluctuation at low resolution and good consistency as resolution increases is apparent in both plots. Other calculated values, in particular the torque on both the stator and the rotor, are also affected by these variations but to a lesser degree and are not shown.

3.5 Summary

Based on the results documented in this chapter, DPLL's accuracy in force and torque calculations and in the modeling of the hull boundary layer is considered sufficient to

allow its use as an evaluator for optimization. Thrust calculations are quite consistent with experimental measurements, particularly when the simplifications of the model are taken into account. Torque calculations, while somewhat less consistent than those for thrust, are near the experimental data in all cases and can be expected to give accurate relative comparisons among notional configurations. The prediction of flow separation, a crucial aspect of exploratory full stern design, appears reliable as does the calculation of boundary layer displacement thickness.

Chapter 4

Pareto Genetic Algorithms

4.1 Overview and Motivation

As mentioned at the end of Section 1.3, a Pareto genetic algorithm is the search mechanism chosen here for optimization of the DPLL model. Genetic algorithms (GAs) are stochastic, non-linear optimization routines loosely based on theories of biological evolution. In contrast to more traditional optimization methods which use gradient information to move between successively better points in solution space, GAs operate on populations of solutions using models of natural selection, reproduction and mutation. As a consequence, they are uniquely adaptable to multi-solution problems such as the defining of Pareto frontiers.

The goal of a GA is to continually evolve its population in the direction of improvement. In single-objective and scalarized multi-objective GAs, improvement is measured by either the population's average fitness and/or its most fit member; in the less common Pareto GAs, by the number of non-dominated solutions and their distribution in objective space.[1] In all types, natural selection, or "survival of the fittest," is simulated by giving preference in breeding to those members of the population who more closely resemble the assumed optima. Offspring are produced by mixing the design parameters (genes) of these preferred members, with some small probability of mutation during the reproductive process. The offspring are then evaluated for fitness and inserted back into the population. These steps are repeated until a termination criterion is met, such as some number of generations without improvement or attainment of a pre-defined objective goal.

Although rigorous proofs of their convergence characteristics remain somewhat elusive and controversial, GAs are increasingly popular in academia and have begun to see application in industry. They are inherently attractive, perhaps due to their

[1] Fitness in a genetic algorithm (or in the biological sense, for that matter) may be thought of as the relative ability to survive and produce offspring.

simplicity and novelty, and are therefore prone to indiscriminate usage. It is advisable, when faced with a practical optimization problem, to ensure that a GA is not chosen simply on the basis of its subjective appeal. This requires a basic understanding of what GAs are and are not, and of what they can and cannot do.[2]

GAs are not reasonable alternatives to deterministic methods of optimization. Linear and non-linear programming techniques, if applicable, will out-perform the best GA. Neither are GAs necessarily more precise or efficient than other non-deterministic methods, such as simulated annealing. In fact, their performance in certain test cases has been shown inferior even to simple stochastic hill-climbing [52]. They are not well suited for applications that require guaranteed response times (such as such as real-time control systems); their response time variance is in fact quite high in relation to other stochastic methods [62].[3]

Rather, the distinguishing characteristics of GAs are their adaptability and robustness; they are not hindered by pathological objective (cost) functions.[4] They are capable of optimizing discontinuous. disjoint, and other poorly-behaved functions and are one of the few alternatives available when no information is available a priori regarding the form of the solution space—in experimental optimization, for example.

In searching an unexplored solution space, a tradeoff must be established between two conflicting objectives: exploiting the information contained in the best known solutions and exploring new regions. Pure exploitation makes exclusive use of existing information; pure exploration abandons all known solutions on the premise that better solutions exist elsewhere. Hill climbing, or perturbation, routines are good examples of exploitative methods where the best known solution is always used as a starting point for improvement.[5] Such techniques take a very narrow view of where opportunities for improvement lie; they are extremely dependent upon their starting locations and are susceptible to becoming trapped at local optima. At the other extreme, totally random search processes provide excellent exploration but make no use whatsoever of acquired knowledge. A properly implemented GA strikes a balance between these two extremes by simultaneously processing information obtained from many potential solutions dispersed throughout the solution space.

The use of populations provides GAs with the ability to locate multiple optima

[2] Actually, GAs are *capable* of optimizing nearly anything, with the exception of anomalous "deceptive" functions which can confound their mechanisms. The question is really how well they perform on a given problem relative to other methods. Goldberg [32] provides an introduction to GA deceptive functions.

[3] They have been shown, however, to be quite suitable for control system design [13, 12].

[4] The GA literature often refers to the evaluator as the "objective" or "cost" function, although it often is not a function in the mathematical sense and usually has nothing to do with monetary cost. Evaluation is discussed in Section 4.3.2.

[5] These examples and this method of describing a genetic algorithm are taken from Booker [7].

simultaneously in the presence of single or scalarized multiple objectives. It also makes them excellent candidates for defining Pareto frontiers, alleviating the need for numerous independent runs to obtain the desired frontier resolution.

Despite being burdened at times with overly optimistic expectations, GAs have demonstrated excellent utility in practice. Recent successes have been documented in such diverse applications as the tuning of power system stabilizers, scheduling of road projects and control of spacecraft rendezvous [71, 73, 53]. Multi-objective applications are now beginning to appear in the literature; examples include the design of broadband microwave absorbers, scheduling the production of chilled ready meals and structural synthesis of VLSI circuits [89, 81, 1].[6]

Due to the stochastic nature of GAs and their relatively recent introduction, there are few absolutes in the implementation of basic GA operators. This is particularly true in the case of Pareto GAs, for which no documented database of results has been established. Implementation decisions during the design of a GA may include, for example, the sizing of the population, the proper mutation rate and the methods by which individuals are chosen to breed, live and die. General implementation guidelines based on published experimental results are scarce and at times contradictory even for non-Pareto GAs; scarcer still are analytical results and recommendations. This research is intended to supplement the Pareto GA performance data by comparing Pareto implementations. In particular, the effect of varying the selection method is investigated, as this is considered the most critical to performance. Variations will be compared based on their ability to locate and define the Pareto frontier for full stern submarines, using DPLL as the evaluator of propulsive efficiency and hull volume.

The Pareto GA developed and tested here operates on populations consisting of multiple, competitive species. The need for a multiple species population is due to the different propulsor configurations allowed by DPLL; as will be seen, these have non-compatible characteristics and cannot be permitted to interbreed. The multi-species concept is apparently unique to this research (a multi-sexual, multi-objective GA has been proposed, but involves no competitive aspect [60]). Further discussion of the concept and its implementation will be taken up in Section 4.4.

4.2 The Schema Theorem

The beginnings of numerical optimization using models of biological processes can be traced to the late 1950's, although credit is usually given to John Holland for pioneer-

[6] Documented multi-objective applications generally employ scalarization. The microwave absorber application by Weile, Michielssen and Goldberg is one of the few Pareto implementations [89].

ing the field in the late 60's and early 70's. Holland was the first to analyze applied evolutionary algorithms, and proposed what has become known as the fundamental theorem of genetic algorithms—the schema theorem—as the principle underlying their success [43]. In the presence of fitness proportionate selection (to be discussed in Section 4.3.3), the schema theorem may be written as follows:

$$m(\mathrm{h}, t+1) \geq m(\mathrm{h}, t)\frac{f(\mathrm{h})}{\bar{f}}\left[1 - p_c\frac{\delta(\mathrm{h})}{\ell - 1} - p_m O(\mathrm{h})\right] \tag{4.1}$$

where

m is the number of times a schema (h) appears in the population,

t is the generation index (time, essentially),

$f(\mathrm{h})$ is the average fitness of the individuals which contain schema h,

$\bar{f}$ is the average fitness of the entire population,

p_c is the crossover probability,[7]

$\delta(\mathrm{h})$ is the defining length of the schema in number of bits,

ℓ is the string length in number of bits,

p_m is the mutation probability, and

$O(\mathrm{h})$ is the order of the schema.

The inequality places a lower limit on the number of copies of a schema which can be expected to propagate to the next generation.[8] The manner in which this limit is exploited by a genetic algorithm is not immediately obvious; it is first necessary to understand what is meant by a schema and to become acquainted with some important schema properties.

A schema (plural *schemata*) is a similarity template common to a set of bit strings (i.e., encoded input or design parameters) which are identical at certain locations.[9] The schema alphabet consists of all the characters of the encoding alphabet plus a "don't care" symbol. For example, the four binary strings {1000, 1010, 1100, 1110} can all be represented by the binary schema [1##0]. This also happens to be the most specific schema which represents this entire set; the remaining representative schemata—all of them less specific—are [1###], [###0] and [####]. These last three schemata represent other strings as well as the set of four above.

[7] Crossover is the usual method of mixing the parameters of selected parent strings in a GA. Crossover probability is the likelihood that mixing will occur during a reproduction. Crossover and crossover probability are discussed further in Sections 4.3.4 and 4.3.5 respectively.

[8] The most common definition of a generation in a GA is a number of evaluations equal to the size of the population. Thus if P is the population size and the population immediately after initialization is called generation zero, then generation one is the population after P evaluations, generation two is the population after $2P$ evaluations, and so on. For non-generational GAs, such as the ones developed here, t may be assumed to increment each time fitnesses are recalculated for the population. Section 4.4 deals with generational issues in more detail.

[9] Some authors, such as Schaffer [78], refer to schemata as "hyperplanes"; Goldberg [33] calls them "building blocks."

A schema can span any fraction of total string length and, as just seen, can have varying degrees of specificity. The schema properties which quantify these concepts are *defining length* and *order.* The defining length of a schema is the inclusive number of bits between the first and last non-"#" characters. Thus the defining length of schema [1##0] is four, whereas the defining length of the schema [##101###] is three. The order of a schema is simply the number of fixed positions, or non-"#" characters, that it contains. The order of a binary schema may be determined by simply counting the number of 1's and 0's.

Genetic algorithms proceed by selecting and copying the strings representing members of its population (with bias toward better individuals), swapping substrings among them, and occasionally mutating a bit to preserve diversity—this is the reproductive aspect. In doing so, they essentially process schemata. GA theory maintains that the availability of a greater number of schemata corresponds to greater efficiency of the algorithm's search. The following discussion will show why this is thought to be true, and how a population's schemata content can vary depending on the encoding alphabet.

Simple analysis reveals that if a binary schema contains k "don't care" symbols, it is represented by 2^k specific strings; a higher-cardinality q-ary ($q > 2$) schema is represented by q^k specific strings. Conversely, 3^ℓ schemata represent any ℓ arbitrary binary bits, although any *particular* sequence of ℓ binary bits will be represented by only 2^ℓ different binary schemata. For a q-ary system of the same capacity, the string will be $\ell_q \leq \ell$ bits long, allowing the formation of $(q+1)^{\ell_q}$ schemata; any particular sequence of ℓ_q q-ary bits will be represented by 2^{ℓ_q} schemata. For example, $3^3 = 27$ schemata can be formed from three arbitrary binary bits as shown in Table 4.1. Of

000	00#	0#0	#00	0##	#0#	##0	###
001		0#1	#01			##1	
010	01#		#10		#1#		
011			#11				
100	10#	1#0		1##			
101		1#1					
110	11#						
111							

Table 4.1: The possible schemata of a three-bit binary string

these schemata, only $2^3 = 8$ are represented by any particular string. The string 000, for example, is representative only of the schemata in the top row of the table. Contrast this to a coding alphabet of higher cardinality; octal, for instance. Only *one* octal bit is necessary to obtain the same capacity (eight possible values, that is) as

the three-bit binary string. Thus $q = 8$ and $\ell_q = 1$ for the equivalent octal system; there are only nine possible schemata, these being any of the single digits 0–7 or the character #. Of these nine, only $2^1 = 2$ are represented by any particular one-bit octal "sequence."[10]

In general, a bit string encoded using a low cardinality alphabet will be representative of a greater number of schemata than the same information coded with a higher cardinality alphabet. A simple example of function minimization, adapted from Goldberg [34] and shown in Table 4.2, will illustrate why this is advantageous. A small

Binary	Octal	Function value
000	0	8
011	3	22
101	5	3
111	7	11

Table 4.2: Binary vs. octal structures. A function value of zero is optimum.

sample population has been evaluated by a hypothetical cost function. The function (objective) values—the result of sending the design parameters to the evaluator—are shown in the right hand column. The actual values of the design parameters themselves are irrelevant to this example, but their notional binary and octal encodings are shown in the first two columns. A scan of the octal encodings and their function values reveals no obvious correlation; it is difficult to imagine how one would proceed with the search based on this information. The binary strings, on the other hand, offer more insight. It is possible that the first string is highly fit because of the 00 in its last two positions or perhaps because of the rightmost 0 alone. Alternatively (or perhaps in addition), it may be that the first and third strings are both relatively fit because they have a 0 at the middle position. Whatever the actual correlation may be, it is apparent that the correlation *possibilities* are increased when the cardinality of the encoding alphabet is decreased. This observation holds in general. The amount of information contained in a lower-cardinality encoding is in effect greater than that contained in higher-cardinality encoding, even though the non-encoded parameters themselves are the same in each case. In Goldberg's words,

> ... many hypotheses can be formulated regarding the association between substring values and high fitness [when low cardinality alphabets are used],

[10] There are obvious problems in attempting to make such a comparison between arbitrary bases; base 4 and base 5, for example. Both require at least two bits to encode eight values but are under-utilized to different extents (base 4 has the capacity to encode 16 values in two bits; base 5 will allow 25). Suffice it to say that higher cardinality in general means fewer representative schemata, and that an under-utilized alphabet presents implementation as well as analysis problems.

and it is this information that is recombined to speculate on possibly better structures during the normal course of genetic search. [32]

By evaluating individuals in the population, a genetic algorithm is in a sense sampling all of the schemata present in their bit strings. The greater the number of schemata in those strings, the more information the algorithm acquires per evaluation. The significance of the schema theorem now becomes more clear. Schemata which tend to be present in above-average individuals will promulgate and multiply while those which tend to be present in below-average individuals will diminish.

The rate of propagation is also affected by the middle term in the brackets of Equation (4.1), the probability of schema disruption by the crossover operator (crossover is the mechanism by which substrings from selected parents are mixed during reproduction). This term favors schemata of short defining length δ, but makes no distinction among encoding cardinalities—the numerator and denominator of this term decrease in proportion, for a given schema, as cardinality decreases. The final term in the inequality, the mutation operator, further reduces the propagation rate of otherwise fit schemata but is somewhat of a necessary evil. It is intended to allow all possible schemata some chance of being produced and evaluated, even those that do not exist in the initial population and cannot be manufactured by the crossover operator.

If the more fit individuals in the population are given preference in breeding and the less fit are more likely to die—a situation realizable with some sort of fitness-biased selection process—the schema theorem dictates an exponentially increasing number of trials for short, low-order, above-average schemata as the generations proceed. Holland conservatively estimated this rate to be $O(n^3)$, where n is the number of function evaluations [32]. This gives the GA tremendous processing "leverage" compared to other search mechanisms; Holland named this leverage *implicit parallelism.* The availability of a large number of favorable schemata multipying at such a rate is obviously conducive to the algorithm's search efficiency. As Schaffer puts it,

> ... [implicit parallelism] constitutes the only known example of combinatorial explosion working to advantage instead of disadvantage. [78]

4.3 GA Functions

The functions of a typical GA are surprisingly simple. In fact, it is partly this simplicity combined with their remarkable search power that makes GAs attractive. Simplicity also makes for relatively easy programming and thus invites variation, as a custom GA can realistically be written for any new application that comes along.

Most original GAs—even if they employ canonical basic fuctions—have their own quirks and novelties, some due to their particular application and some based on the intuition of the programmer. GAs in the literature thus share basic features but are usually unique in some way, more often in implementation than in theory. This makes it difficult to define what exactly is meant by a "typical" GA. Nonetheless, the concept of a typical GA is needed here so that it can be compared to the Pareto GA developed during this research. For this purpose, a typical GA will be taken to mean one having all of the most common features among those in the current literature.

A typical GA minimizes a single objective. The evaluator or objective function returns a measure of "goodness" for each individual and the algorithm seeks to increase the goodness of its population through recombination (swapping of substrings, with mutation). The less common multi-objective GA (MOGA) has a notable difference: the evaluator returns more than one objective value per individual; this objective "vector" is then scalarized to result in a single measure of goodness. Pareto GAs, the least common of the three types, also differ from typical GAs primarily in the way they measure goodness.[11] In contrast to scalarized MOGAs, however, goodness in a Pareto GA is measured not by combined objective values but by relative dominance. To a Pareto GA, all non-dominated individuals are equally good, regardless of their absolute objective values.

Apart from the number and assessment of objectives, basic functions of typical and Pareto GAs are essentially the same. These are shown in the flow diagram of Figure 4-1 and discussed in the following sections. In most cases, a brief description of the function will be given, followed by a mention of how it might be implemented in a typical GA. Any differences between the typical implementation and that used in this research will also be noted. Application-specific issues resulting from the use of DPLL as the evaluator will be covered in detail.

4.3.1 Establishing Generation Zero

Design Parameters, Ranges and Discretization

An initial population, or generation zero, is usually established by generating random combinations of input (design) parameters, evaluating them and storing the results until the desired population size is attained. During the random generation of each new individual, the input parameters are allowed to take on any one of a pre-determined set of values with equal probability. A full set of input parameters defines a variant, or individual, and the evaluation of a set of parameters determines

[11] Schaffer [76, 77] was apparently the first to adapt a GA to search for Pareto optima in 1984.

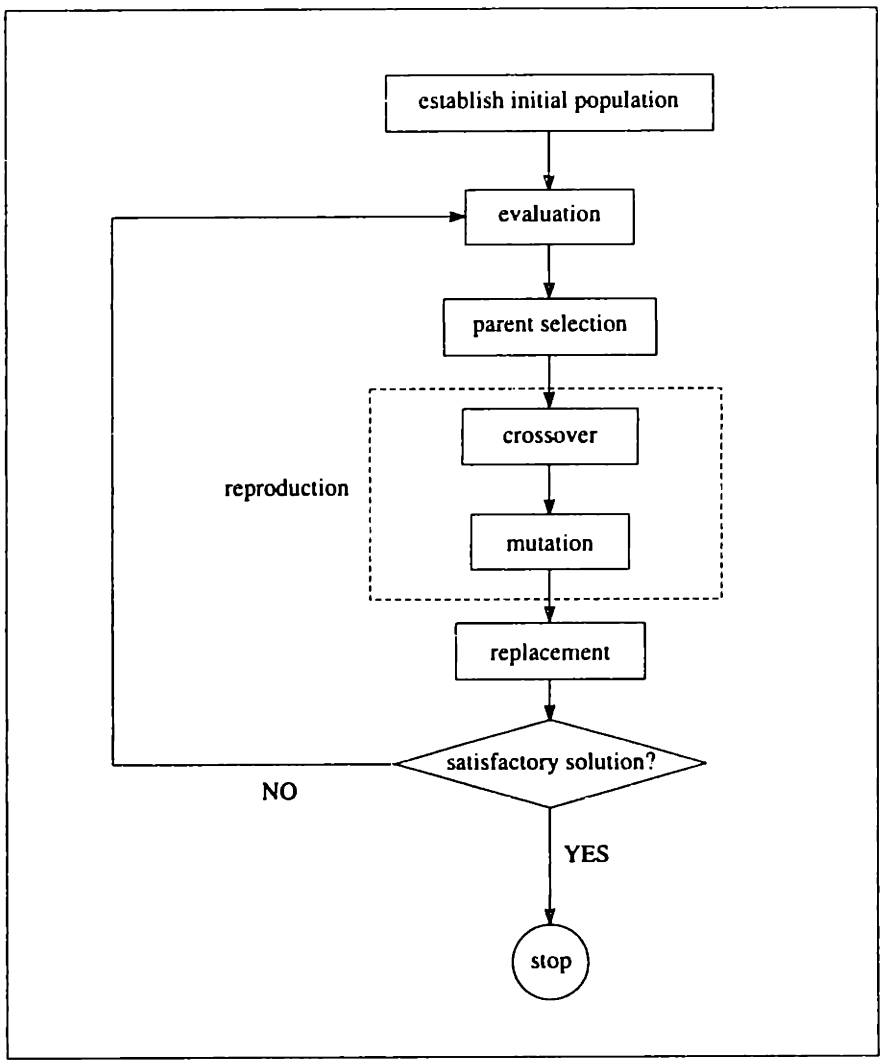

Figure 4-1: Flow diagram for a typical genetic algorithm.

its objective value (or values, in multi-objective optimization) and in some cases its feasibility.

For example, if one sets out to design a crate which maximizes internal volume while minimizing surface area, the likely design parameters are the crate's length, width and height. The set

$$[\, l = 3,\ w = 2,\ h = 4\,]$$

defines one particular variant which, when processed by the evaluator functions

$$\text{volume} = l \times w \times h$$
$$\text{surface area} = 2\,[(l \times w) + (l \times h) + (w \times h)]\,,$$

yields a volume of 24 and a surface area of 52. If the design requirements specify

a volume of, say, at least 18 and a surface area of no more than 50, evaluation has revealed the objective infeasibility of this variant as well as its objective values. In some cases feasibility may be a function of design parameters as well as objective values; for example, the crate may be required to fit through a door of given dimensions. This would place upper limits on the crate's length, width and height, effectively establishing their feasible ranges since zero is the obvious minimum for all three. In cases where no obvious input limits exist, they must be determined by heuristics or simply guessed and adjusted later if necessary. Once established, parameter ranges must be discretized such that only particular values within each range are allowed.

Allowable values of design parameters must be discrete and pre-determined because of the mechanics of the GA reproductive process. During reproduction, input parameters are converted to ordinals and then encoded into bit strings. The resulting strings are commonly referred to as *chromosomes*, and all the bits making up a single parameter value are collectively known as a *gene*.[12] Table 4.3 shows how the allowable values of a notional real design parameter (the length of the crate in the previous example, say) might be represented as ordinals and as binary genes. Width

Parameter	Ordinal	Binary
2.015	0	00
3.030	1	01
4.045	2	10
5.060	3	11

Table 4.3: Notional real parameters, as ordinals and binary genes

and height would be encoded similarly (but not necessarily with the same resolution) and the resulting genes strung together to form a chromosome. If each design parameter were not restricted to discrete values within a given range (for example, if the length of the crate were allowed to take on *any* real value between 2.015 and 5.060), the number of bits required for the gene could not be determined beforehand.[13] Genes (and therefore chromosomes) of unspecified length present several difficulties, particularly in mating, and make the programming of the GA much more cumbersome. Although such problems are not insurmountable ("messy" GAs have been developed which successfully operate on chromosomes of variable length [37]), the documented benefits of variable string lengths are not conclusive. This research will deal only with

[12] Much of GA terminology is understandably borrowed from the biological sciences. Two less frequently used but related allegorical terms are *allele*, meaning the actual value of a bit, and *locus*, meaning a bit's position on the chromosome.

[13] It should be noted that discretization of parameters does not place any limitations on resolution beyond the precision and capacity of the hardware used. Greater resolution simply requires longer chromosomes and generally slows the convergence of the algorithm, as is typical of any numerical method.

chromosomes of equal length.

While the feasible ranges of input parameters may be known or estimated in most cases, little guidance is generally available concerning their proper resolution. Obviously the choice of base is a factor; in binary coding one would be well advised to choose some power of two for each parameter's resolution. Beyond this, the choice must be based on the trade-off between resolution and computation time. Discretizing each parameter to maximum precision will likely result in very long chromosomes and search times. GAs are very efficient search engines, but they cannot be expected to sample enormous design spaces effectively in limited time. When determining parameter discretizations, then, it is advisable to keep the time vs. precision trade-off in mind. Lengthening a binary gene by one bit, for example, doubles the size of the design space. Discretization decisions should generally favor low to moderate resolution, at least until preliminary results have been reviewed.

DPLL Parameters

Table 4.4 shows the design parameters used in this research with DPLL as the evaluator; they are described in detail below. Parameter ranges here are chosen via a

Input Parameters			
Parameter	Range	Resolution	Type
configuration	1 : 4	4	integer
fullness factor	0.5 : 1.0	16	real
tip radius	0.02 : 0.05	8	% body length
hub circulation	0.010 : 0.035	8 per stage	H
tip circulation	0.010 : 0.035	8 per stage	H
duct/rotor intersect	0.2 : 0.8	8	% duct chord
duct chord length	0.001 : 0.080	8	% body length
duct load	−0.04 : 0.10	8	G

Table 4.4: Input parameters and ranges used for DPLL optimization. Blade circulation values are positive for rotors, negative for stators.

combination of heuristics and trial runs so as to allow all reasonable possibilities. Most parameters are given a resolution of eight; this is assumed to be a reasonable initial compromise between precision and computation time. The greater resolution of the fullness factor is due to its strong correlation to one of the objective values (volume); this will be discussed further in Chapter 5. For the purposes of this optimization, other inputs required by the stand-alone version of DPLL (forward speed and number of propeller blades, for example) are either fixed for all variants or are calculated as functions of the parameters shown. The total number of variations possible (i.e., the size of the design space) using this resolution is 1.77×10^{10}.

1. Configuration determines the number, type and relative locations of propeller stages. Configuration 1 is rotor-only, 2 indicates a stator upstream of the rotor, 3 a stator downstream of the rotor, and 4 indicates two stators, one upstream and the other downstream of the rotor. These four different configurations are treated by the reproductive mechanisms of the GA as separate, co-located species. They are not allowed to interbreed but do compete for the same "resources" (fitness values) and are therefore subject to dislocation and extinction. The concept of competitive species is discussed further in Section 4.4.

2. Fullness factor determines the shape of the stern. Each variant begins with a common set of B-spline vertices defining the body profile. This template produces a gently sloping profile from mid-body to stern and is not modified if the fullness factor is 0.5 (the minimum).[14] Any fullness factor greater than 0.5 scales up the z coordinates of the vertices defining the stern, but leaves the r coordinates unchanged.[15] The result is a greater extent of parallel mid-body and a more abrupt transition at the stern. The relationship between vertex scaling and fullness factor is set such that a fullness factor of 1.0 (the maximum) results in a transition abrupt enough to cause separated flow for nearly all combinations of the remaining parameters; this is one instance where trial and error is required in setting parameter limits. Figure 4-2 shows the range of hull profiles generated by varying the fullness factor between 0.5 and 1.0. The actual resolution is about twice that shown, as intermediate profiles have been omitted from the plot to avoid clutter.

3. Tip radius is the r coordinate, as a fraction of body length, of the governing propeller tip. It also specifies the radial offset of the duct, as the duct camber line is required to intersect the governing propeller tip (DPLL uses a preliminary rigid translation of the input camber line to ensure this intersection). In the modified version of DPLL used in this optimization, the governing stage is automatically specified as the one which is farthest forward on the body. Tip radii of any additional stages are calculations based on the solved shape of the duct camber line during each iteration.

4. Hub circulation specifies the non-dimensional local circulation on the lifting segment nearest the hub. As discussed in Section 2.2.1, circulation is essentially the

[14] There is no particular reason for the lower limit being 0.5 other than to avoid a lower limit of zero, which might be construed as zero volume. In any regard, the numerical value itself is meaningless; it simply represents one of eight possible sets of body profile vertices to DPLL and an ordinal number, for encoding, to the GA.

[15] The vertex which closes the stern at $z = 1.0$ and the vertices defining the sting are unaffected (the sting is described in Section 2.3.7).

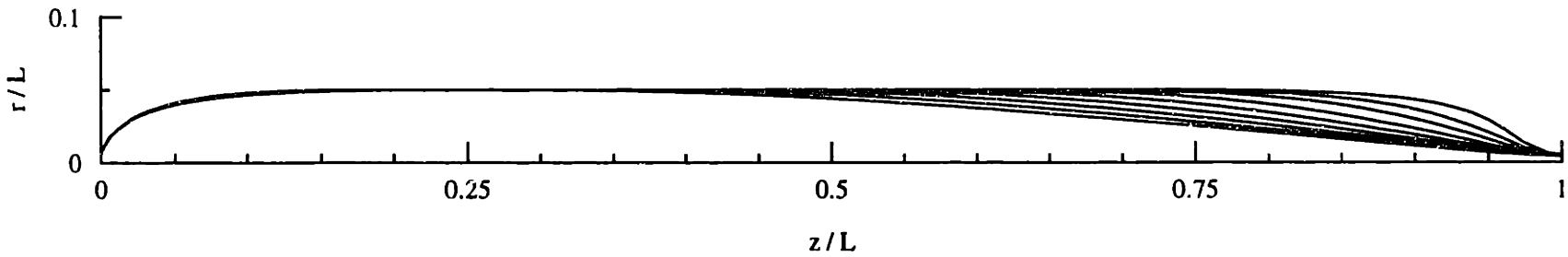

Figure 4-2: Range of allowed hull profiles for optimization.

velocity difference across a foil and is thus a measure of blade "load." Independent hub circulation values are specified for all three propeller stages, regardless of how many are actually present (that is, regardless of configuration number), in order to maintain uniform chromosome length. Extraneous values are simply disregarded during evaluation. Circulation values for stators are made negative so as to induce tangential velocities opposing those of the rotor.

5. Tip circulation is the non-dimensional local circulation on each stage's outermost lifting segment, specified in the same manner as the hub circulation. Values of circulation at intermediate lifting segments are determined by interpolating between the hub and tip values, resulting in a linear distribution across the blade span.

6. Duct/rotor intersect specifies the intersection of the duct camber line and the rotor tip as a percentage of duct chord length. It effectively positions the duct axially, as the rotor location is fixed and constant for all variants.[16]

7. Duct chord is the axial distance between the duct leading and trailing edge, as a fraction of body length. If the configuration value is 1 (no stators present), the minimum duct chord allowed is $0.001L$. This negligible value is used to approximate a non-ducted system, as the current version of DPLL requires the presence of a duct. For configurations which include stators, the no-duct approximation is not allowed. This is because all propeller stages must be enclosed by the duct, and negligible duct chord would result in an unrealistically small separation between successive stages.

8. Duct load is non-dimensional total circulation on the duct, defined as the sum of the camber line vortex ring strengths. This definition of duct load is closely correlated to lift; a negative load with a positive angle of attack will produce negative thrust (i.e., opposing the rotor).

Sample Chromosome

Table 4.5 shows a notional binary chromosome, such as might be generated by the initialization or reproduction routines and sent, in its real form, to DPLL for evaluation. The configuration gene (cfg) of this variant decodes to ordinal 2, meaning that it is the third of the four possible configurations and that its propulsor consists

[16] Rotor location is an example of a design parameter which is fixed simply to keep combinatorics to a manageable level. It would no doubt be an interesting parameter to vary in further studies.

	cfg	fulfac	r_{tip}	H_{hub1}	H_{hub2}	H_{hub3}	H_{tip1}	H_{tip2}	H_{tip3}	S_{rotor}	c_{duct}	G_{duct}
binary	1 0	0 1 1 1	0 1 0	0 0 1	0 1 1	1 1 0	1 0 0	0 1 0	1 1 1	0 1 1	1 0 0	1 0 1
ordinal	2	7	2	1	3	6	4	2	7	3	4	5
real	3.0	0.7100	0.0286	0.0136	0.0207	0.0314	0.0243	0.0171	0.0350	0.4570	0.0461	0.0600

Table 4.5: Example chromosome with decoded values.

of a rotor followed by a stator.[17]

The fullness factor (fulfac) decodes to ordinal 7, out of 16 possible values. The real number, an arbitrary scaling of the ordinal to the range 0.5–1.0, is 0.71 by interpolation.

The tip radius decodes to ordinal 2, out of 8 possible values between $0.02L$ and $0.05L$ inclusive (L is body length). Thus the real tip radius by interpolation is $0.0286L$, or 2.86% of body length. Since this is a rotor-stator configuration, the rotor is the most forward and therefore the governing propeller stage to which this tip radius applies.

The first and second hub and tip circulation values are assigned to the rotor and stator respectively; the third hub and tip circulations in this chromosome are ignored. The rotor tip circulation gene (H_{tip1}) will be used here as an example. This gene decodes to ordinal 4, out of 8 possible values between 0.010 and 0.035 inclusive. The real non-dimensional circulation H is thus 0.0243 by interpolation. Using the decoded value of the tip radius gene from above and Equation (2.12), the local tangential velocity at the tip is found to be $0.106V_s$ or about 10% of ship speed (as for the remaining unknowns in Equation (2.12), all variants considered in this optimization have a maximum hull radius of $0.05L$ and five propeller blades). Note that this is *induced* velocity only and does not include the rotor's angular velocity. The remaining hub and tip circulation genes decode similarly, except that the radial coordinates of these points vary slightly as the iterations progress and the solved shape of the duct camber line changes.

The duct/rotor intersect gene (S_{rotor}) specifies the axial position of the duct camber line so that it intersects the rotor tip at 45.7% of camber line arclength. The duct chord gene (c_{duct}) indicates that the chord length is $0.0461L$, or 4.6% of body length. Since the rotor location is fixed, these two parameters along with the tip radius fully describe the initial position of the duct. Stators, which must be individually located with respect to the duct chord when DPLL is run as a stand-alone program, are automatically positioned in this optimization. For the rotor-stator configuration of

[17]The four possible configurations dictate a two-bit binary gene, which has ordinal values of zero through three. Configuration numbers are the ordinal values plus one; this is simply an aesthetic avoidance of zero as a configuration number. In retrospect, the translation is probably more confusing than helpful.

this example, the stator tip is placed midway between the rotor and the duct trailing edge and the lifting line is grown inward, normal to the local velocity, to intersect the hull profile.

The duct load gene G_{duct} decodes to a non-dimensional circulation of 0.06, which is essentially the velocity delta between the duct's upper and lower surface, non-dimensionalized by the radii of the duct control points and ship speed and summed over all duct control points. The positive value indicates counter-clockwise circulation about a centerline ($\theta = 0$) section through the duct, so by Equation (1.1) it should produce lift with a component in the forward ($-z$) direction.

Population Sizing

Population size is among the parameters having the greatest impact on GA performance, second perhaps only to selection method. There are no firmly established guidelines for determining the correct population size; in fact, sizing is somewhat of a dilemma. If the population consists of only a few individuals, the selection process is hindered by a lack of choices. The number of schemata available to the GA is very limited and improvement will almost certainly be intolerably slow. If by some fortunate accident an exceptionally fit individual is produced, it stands a fair chance of being quickly eliminated due to the quirks of stochasticity. At the other extreme, if the population size is huge (approaching the size of the design space, say), then the GA is irrelevant because establishing generation zero requires a near-exhaustive search and the optima will become evident as by-products of this process.

Population size also affects convergence time. Goldberg [35] has shown that typical GAs converge in $O(\log P)$ generations, where P is the population size and a generation involves P cost function evaluations. This would indicate that smaller populations require fewer evaluations. Convergence, however, is not necessarily equivalent to locating the global optima. Again considering the extreme cases, a two-member population will probably converge (stop improving, that is) quickly since there are very few schemata to process. Convergence will most likely occur at a point near the original two members. In contrast, a huge population will be far from converging after the same number of evaluations, but will probably have better solutions available at that time due to the extensive initial "random search" performed while establishing the population. These general observations are supported by the work of De Jong, who performed some of the first experiments on GA convergence properties in 1975:

> Increasing the population size was shown to reduce the stochastic effects [of random sampling on a finite population] and improve long-term perfor-

mance at the expense of slower initial response. [19]

The "correct" population size in terms of efficiency and efficacy obviously lies somewhere between the two extremes. De Jong suggested a population size of 50–100, based on his suite of test functions. Grefenstette [40] took the novel approach of using a GA to determine optimal GA performance parameters and suggested a population size of only 30. In one of the first analytical investigations, Goldberg [31] proposed an optimal population size based on the length ℓ, in number of bits, of the chromosome:

$$P = 1.65 \times 2^{0.21\ell} \tag{4.2}$$

Schaffer et. al [79] later proposed an empirically optimal P of 20–30, and Goldberg in a more recent analytical work suggested that with a few simplifying assumptions, the correct population size is probably of $O(\ell)$ [36].

A population size of 200 is used in this research. The chromosome length is 36 binary bits, which gives $P = 311$ using Equation (4.2). All of the remaining studies mentioned above, including Goldberg's more recent work, would indicate a much smaller population size; however, the Pareto GA developed during this research operates on four distinct species simultaneously (see Section 4.4). As this is an original concept, there is no way of predicting its effect on optimal population size. The value of 200 is assumed to be conservatively high; paying an initial response time penalty is considered preferable here to premature or false convergence.

4.3.2 Evaluation

Evaluation is the process of determining the objective value(s) of a member of the population, represented as a set of design or input parameters. Individuals are sent to the evaluator as a set of inputs in evaluator language, meaning that they must be converted from their encoded chromosome states back into actual parameter values. The evaluator is completely independent of the other GA functions; it has no memory or biasing features of its own. It is often referred to as the objective or cost function, although it need not be an explicit function in the mathematical sense and often is not. In fact, it can take almost any form, such as complex stand-alone analysis programs (such as DPLL) or even experimental testing (for an example of this, see Barrett's application to flexible underwater vehicles [2]). Objective values returned by the evaluator provide exclusive guidance for the GA's search process. This is in contrast to the majority of optimization algorithms, which utilize gradient information.

In this research, DPLL is used as the evaluator. The input parameters have been previously discussed; the calculated values of interest—the objective values—are usable hull volume, power coefficient and, eventually, cavitation index. Hull volume is determined by a trapezoidal integration of the body profile and adjusted by estimates of internal reduction gear and shafting volumes as functions of developed torque. Appendix B details the method used to estimate these internal volumes. Power coefficient is defined as:

$$C_P = \frac{Q\omega}{\frac{1}{2}\rho V_s^3 A_B} \tag{4.3}$$

where Q is shaft torque, ω is shaft angular velocity, V_s is forward speed of the vehicle and A_B is the cross-sectional area of the vehicle at the point of maximum radius. Shaft torque and angular velocity required for force-balancing are calculated by DPLL; forward speed and the body profile (and thus the maximum cross-section) are inputs. Since forward speed and maximum radius are identical for all variants in this optimization, the power coefficient serves as a measure of relative propulsive efficiency. Cavitation index is a function of lift at the rotor tips, relative fluid velocity, tip speed, and geometry. Appendix A describes this calculation in detail. A lower value of cavitation index as defined there corresponds to reduced cavitation and noise.

Thus the general objectives of this Pareto optimization—not to be confused with the objective *values* of any particular variant as per DPLL—are an *increase* in volume, a *reduction* in power and a *reduction* in cavitation at the given forward speed. Superlatives such as maximum or minimum are inappropriate, as the achievable limits are initially unknown and the GA operates only on comparisons. Even if the limits were known, the GA could not guarantee discovery of the corresponding design parameters, being non-deterministic. This is why GAs in general have no inherent termination point. Among the typical convergence criteria used are some number of generations without improvement (this corresponds to frontier stability in Pareto GAs and is the criterion used here), the attainment of some pre-defined objective goal, or the onset of "population takeover," where the population becomes homogeneous or nearly so. This last phenomenon occurs only in GAs which allow the existence of duplicates (more about this later).

4.3.3 Selection

Selection is the process by which members of the population are chosen to mate and reproduce. Selection and breeding of individuals with above-average schemata, as evidenced by their superior objective values or relative dominance, should generally

produce above-average progeny, thus moving the population as a whole toward improvement. The most obvious method of selecting parents, simply choosing the best phenotypes, produces poor results because the search quickly becomes localized.[18] This often leads to premature convergence near the best members of the initial population. In order to achieve the previously mentioned balance between exploitation and exploration, a selection method must *favor* the more fit individuals—this requires biased-random selection based on fitness.

Objective Values and Fitness

Objective value and fitness are generally not equivalent. An individual's objective value is a function only of its design parameters, with the function being defined by the evaluator. Fitness, on the other hand, is a relative term which indicates an individual's standing among its contemporaries. More formally, if $\vec{\psi}$ is a set of input parameters and $\vec{\Theta} = E(\vec{\psi})$ is the vector of objective values returned when $\vec{\psi}$ is sent to the evaluator, then the fitness F of $\vec{\psi}$ is given by:

$$F(\vec{\psi}) = f\left(E(\vec{\psi}), t\right) = f(\vec{\Theta}, t) \tag{4.4}$$

where t is the generation index of the algorithm. When all individuals in a population have been assigned fitness values, selection proceeds by making stochastic choices among them based on fitness. Those selected go on to produce offspring.

Fitness Scaling, Clones and Ranking

One of the simplest fitness functions f to implement is a scaling of raw objective values to the limits of the current population. This is known as *proportionate* or *linear fitness scaling* [41]. Fitness scaling followed by some form of stochastic selection is a decided improvement over simply selecting the individual with the best objective value; however, the problems of intense local selective pressure and premature convergence are not entirely eliminated. If one member of the population happens to be far superior to the others, fitness scaling tends to favor that individual to the exclusion of others. The result may be a population full of duplicates or near-duplicates of one early, fortuitous (but non-optimal) individual.

Some discussion is in order here about duplicates, or clones, and how they affect population takeover. Clones are individuals which are identical in terms of all active design parameters to some other member of the population. They usually arise

[18] Another example of borrowed biological jargon. The GA *genotype* is the set of design parameters (the evaluator input); usually it refers to an encoded chromosome but may also mean the set of non-encoded parameters. The *phenotype* is the set of corresponding objective values (the results of an evaluation). In biology, the genotype is the genetic code (an organism's chromosomes, basically) and the phenotype is the manifestation of the code (the expression of traits, or the characteristics of the organism) [85].

through reproduction but may occur during the initialization process as well. In single objective GAs and MOGAs, they do not present a major problem as long as the population size is sufficiently large and some mechanism, such as mutation, is in place to preserve diversity and prevent premature convergence. The proliferation of clones to the point where they comprise the majority of the population, making further improvement unlikely, is known as population takeover. In single objective GAs, this is expected behavior and is often used as an indicator of convergence. In Pareto GAs, however, clones are usually considered detrimental because they decrease the potential frontier resolution and population diversity [91].

There are three basic ways of dealing with duplicates: they can be tolerated, penalized or rejected. Since tolerating clones is counter-productive in Pareto optimization, the choice here must be between penalization and rejection, even though both methods require additional processing time for clone identification.[19] The only rationale for choosing penalization is that time already spent generating the clones is wasted if they are then identified and rejected.[20] This argument is not persuasive, considering that (1) even more time will be spent determining and assigning the penalty, (2) the problem of diminished resolution is only partially solved since some number of clones will generally be present in the population, and (3) reproduction, mutation and identification time—that is, the time required to generate and identify a clone—is usually insignificant compared to evaluation time. For the Pareto GA developed here, the evaluation of a single individual by DPLL requires at least ten minutes on a 400 MHz DEC Alpha workstation, while all GA functions combined (including clone identification and ranking of the entire population) require less than two seconds. Penalization is therefore not justified in terms of time savings, and rejection is used exclusively here.

Returning now to the discussion of proportionate fitness assignment: recall that proportionate assignment followed by a stochastic selection method alleviates some convergence pressure, but fitness remains based on relative phenotypical distance. This is not conducive to exploration and can lead to poor results, particularly for multi-modal functions. Even single objective/single optimum searches can be mislead by fitness scaling if a relatively fit individual arises in a flat, non-optimal region of objective space. Also, since fitness scaling requires a scalar objective value, it must be preceded by scalarization (a weighting method, for example) when used on multi-objective problems. It is therefore not compatible with a true Pareto GA, and is not

[19] Schott [80] performed convergence studies on a Pareto GA using each of these three methods, confirming that both penalization and rejection are significantly more efficient than tolerating clones.

[20] Note that this refers to reproduction, mutation, or initialization time, *not* evaluation time. There is no conceivable reason for evaluating an identified clone—its objective values are already known.

suitable for this research.

One way of smoothing out the intense pressures of fitness scaling is to employ a *ranking* scheme. This involves assigning fitness according to relative standing in the population, regardless of the phenotypical distances involved (e.g., best = 1, second best = $\frac{P-1}{P}$, ... , worst = $\frac{1}{P}$). If two or more individuals have equal standing, they are assigned an averaged rank. For example, if there were n "best" members, they would all be assigned a rank of

$$\frac{P + (P-1) + (P-2) + \cdots + (P-n+1)}{nP} = 1 - \frac{1}{nP}\sum_{i=1}^{n-1} i \tag{4.5}$$

Stochastic selection then proceeds as it would with proportional fitness scaling, but is based on rank rather than relative objective value. Ranking methods are well suited for Pareto optimization, where relative dominance is the only meaningful criterion for comparing individuals. The "best" members of the population in the Pareto sense are those that are non-dominated, the "second best" are those dominated only by the "best," and so on.

Ranking also allows control over the ambient selection pressure [91]. For example, if the simple ranking system mentioned above provides insufficient pressure—as might be indicated by excessive convergence times—the assignments can be made on an exponential rather than a linear ranking scale. The general implementation would be to assign fitnesses of 1, s^j, s^{j+1}, ..., from best to worst, where $s < 1.0$ and j is some positive integer. Both s and j may be varied to obtain the desired level of pressure [27, 41]. Ranking methods employed in this research use such a scale, with $s = \frac{P-1}{P}$ and $j = 3$.

Roulette Wheel and Tournament Selection

Stochastic selection mechanisms following either a proportionate fitness assignment or a ranking scheme appear in several variations in the literature. The most simple of these, used in this research, is known as "roulette wheel" selection. A vector of cumulative fitness values for the population, in arbitrary order, is constructed as follows:

$$\bar{\mathcal{F}} = \frac{\{F_1, (F_1 + F_2), \ldots, \sum_{i=1}^{P} F_i\}}{\sum_{i=1}^{P} F_i} \tag{4.6}$$

A random number r of uniform probability distribution between zero and one exclusive is then generated and the member whose index in $\bar{\mathcal{F}}$ is defined as follows is the chosen parent:

$$i_p = \{i : \mathcal{F}_{i-1} < r \leq \mathcal{F}_i\} \quad \text{with} \quad \mathcal{F}_0 \equiv 0 \tag{4.7}$$

A drawback of the roulette wheel method is that it is inherently noisy; i.e., the variance of the probability density function for selecting a string of given fitness is relatively large. This noise, along with that generated by the reproduction and mutation operators, tends to obscure the signal difference between good and bad schemata, thus hampering the GA's ability to select well. Goldberg [36] has analyzed the effects of noise and compared some of the more common selection methods using a suite of test functions of varying complexity. His results, in addition to confirming the advantages of ranking over fitness scaling, also show that some alternatives to roulette selection may be preferable.

One such alternative is tournament selection. A tournament is held among some pre-defined number of individuals—chosen randomly from the population—and the tournament winner is selected as a parent. No roulette-type routine is necessary, as the random selection of contestants provides the required stochasticity. Tournament victory can be based on scaled fitness values (which, for tournament methods, is equivalent to simply using raw objective values) or on relative rank. Dominance ranking is the only viable alternative when applying tournament selection to Pareto GAs.

Niching

Rank-based tournaments do not necessarily result in a unique winner and therefore must include some means of resolving a tie. Selecting randomly from the tournament winners is reasonable, but there are advantages to be had by applying greater discretion. These advantages are a result of the fact that multiple variants in close phenotypical proximity (those which are crowded together in objective space) contribute little additional information to the GA. In fact, they can be detrimental to performance as they take up space in the population which could be occupied by more diverse variants. In this sense, such "near neighbors" are quite similar to clones, although phenotypical proximity does not necessarily correspond to genotypical proximity. Pareto GAs are particularly affected by phenotype crowding, as their goal is a uniform distribution of the population across the non-dominated frontier.

Resolving tournament ties and discouraging phenotype crowding are compatible objectives. Ties can be broken by selecting the individual from among the winners whose region in phenotype space is the least crowded. The degree of crowding around an individual is determined by conducting a "niche count." The niche count is simply the number of other members which lie within some hyper-volume, or niche, centered on the individual in objective space. The obvious parameters necessary to define such

a niche are its size and shape.[21] Although there has been some investigation into the efficacy of various niche definitions, including asymmetric shapes [45], there remains little evidence as to the benefit obtained by such complications.

The simplest approach to niching—the box niche used here—utilizes pre-defined phenotypical distances to test for proximity. For example, individual X is said to be located within niching distance of individual Y if the difference in their values of objective A is less than some A_{niche} *and* the difference in their values of objective B is less than some B_{niche}. These tolerances may be static or dynamic according to latest population statistics. In either case, they are usually based on a calculated optimal separation distance, determined by dividing the assumed frontier surface area by the size of the population [27]. Note that if X is in the niche of Y then the converse is true, and that as the number of objectives increases, the time-averaged niche density decreases for a given population size.

Niching should be a concern in any Pareto GA, not only those employing tournament selection. Niche counts may be accounted for in ranked GAs, for example, by applying the count as a rank penalty prior to stochastic selection. This is the approach used in the ranking methods of this research.

Constraints

The selection process includes the handling of constraints on objective values. The crate example of Section 4.3.1 demonstrated that objective feasibility can only be determined by evaluation. What the example's simplistic objective functions failed to emphasize, however, is that significant computation time may have already been invested before a variant is known to be infeasible. This is certainly the case in this research, where a single evaluation by DPLL can require up to 30 minutes. Time invested must be a factor in determining how to deal with infeasible individuals. Three of the more common methods used are [18]:

1. immediate rejection
2. repair
3. penalization

Rejection—simply discarding the offending variant—guarantees the feasibility of the population, but is inefficient. Since no consideration is given to the degree of violation, rejection ignores the potentially useful information present in an "almost feasible"

[21] See Fonseca and Fleming [26] for a description of how niche shape can be affected by varying the degree of the Holder metric.

individual. It is therefore not well-suited for identifying the limits of feasibility and may overlook constrained optima. Another concern is that for problems having highly constrained solution spaces, the identification of *any* feasible solutions may be nearly as difficult as finding the optimal solutions. Since rejection extracts zero information from infeasible variants, it is likely to produce poor results in such cases [33].

Repair of infeasible variants is necessarily problem-specific and complicated. The type and extent of the repair must depend on which constraint has been violated and the magnitude of the violation; the repair algorithm, in order to be effective, must be a complex, dynamic process which "learns" and improves itself as the generations progress. Repair is not necessarily more efficient than rejection, as the repaired variant will require re-evaluation and the repair process itself may be computationally intensive.

The penalty method assumes that infeasible variants may contain useful genetic information and allows them to remain in the population, although with a fitness penalty. This method is more likely than rejection to locate constrained optima, as variants which are "barely" infeasible may provide schemata which lead to the optima. Unfortunately, the assignment of penalties is necessarily somewhat problem-specific. Over-penalization will produce essentially the same results as rejection, but with an added computational burden. Under-penalization can be even worse: if the penalty is not at least equal to the objective advantage gained by violating constraints, the population will quickly become saturated with infeasible members. It is not clear how a balance between these two undesirable extremes can be struck without prior knowledge of the solution space or some amount of trial and error.

This research employs the penalty method and originates a technique which may be helpful in generalizing penalty assignments. The constraint in this case is flow separation; it is violated if separation occurs upstream of the rotor. The requirement of attached flow may be considered a "soft" constraint, as the degree of violation increases with the upstream distance of separation rather than being a simple feasible/infeasible dichotomy. Variants which violate this constraint are penalized by decreasing their relative dominance in the population. The penalty is applied as an integer number of dominating "phantom" members, seen only by the individual to whom the penalty applies. This number increases with the distance between the rotor and the upstream separation, and has a dynamic lower limit equal to the total number of infeasibles in the current population. This floating lower limit is the generalization developed here and mentioned above; it essentially acts as negative feedback on the proportion of infeasibles in the population. If the number of infeasibles begins

to increase—possibly because the magnitude of the feasibility penalty has initially been set too low—the minimum penalty for infeasibility also increases. This gives a greater selective advantage to the feasible individuals, which are never considered dominated by constraint violators regardless of the relative objective values involved. The result is a decrease in the number of infeasibles in the population over time, with a corresponding decrease of the minimum infeasibility penalty.

Specific Methods Investigated

Three specific selection methods are investigated here, including two variations on ranking and a tournament method. All employ niching and constraint penalization. These will ultimately be compared in terms of their ability to locate and define the Pareto frontier for DPLL's volume and power calculations in Chapter 5.

1. Goldberg ranking[22] is based on dominance layers. Non-dominated individuals (those in the outermost layer, or layer zero, in objective space) are assigned the best rank, individuals which are dominated only by those in layer zero are assigned the next best rank, and so on. Niche counts are then performed and the population is again ranked, this time according to niche count. This results in two ranks per individual; these values are added and a penalty is applied to variants which violate the separation constraint. The population is sorted according to this combined and penalized rank. Fitness values are then assigned to the sorted population according to the exponential scale discussed in Section 4.3.3. Parents are selected by consecutive applications of a roulette wheel routine.[23]

2. Fonseca and Fleming ranking assigns each individual a rank based on the number of other individuals by which it is dominated. As with the Goldberg method above, these ranks are modified by niche counts and constraint violation; the population is then sorted and assigned fitness according to an exponential scale. Parents are selected by the same roulette wheel routine.

3. Tournament selection randomly chooses 10% of the population and selects the dominant member. A tournament contestant which violates constraints will incur a relative dominance penalty. This penalty will reduce a constraint violator's tournament standing by at least the number of infeasible contestants. The net effect is that a constraint violator tied for first place prior to penalty assessment will instead be eliminated from consideration. A constraint violator which oth-

[22] Ranking methods are named here according to the authors who proposed them. This terminology is not universal.
[23] Selection of the second parent is somewhat complicated by the presence of other species, as discussed in Section 4.4.

erwise would have won the tournament outright will finish in a tie for first place at best. Post-penalty ties are resolved by niche count.

It should be noted that these three methods are not arbitrarily chosen from among several possibilities. There are apparently no other general means of applying purely Pareto criteria in selection—at least there are none documented [28]. The main Pareto GA framework developed here incorporates these selection methods as subroutines. Switching among them is done prior to compiling. The objective here is to maintain uniformity in all non-selection functions, allowing valid comparison of relative performance among the three methods.

4.3.4 Reproduction

Reproduction is the process by which selected members of the population breed to produce new variants, which are not surprisingly referred to as offspring or children. During the selection function just discussed, highly fit individuals are chosen to be parents on the basis of their objective values. Since the evaluator is essentially a function relating input parameters to objective values, and objective values are a GA's only criteria for selection, the chromosomes of selected parents will on average contain relatively high quality schemata. It is the job of the reproduction process to mix and match these schemata so as to produce more highly fit children.

Coding

The code to be used for converting design parameters to bit strings for mating is one of several controversial aspects of GA research. Holland's work was done with binary coding, and this remains the most common method in the literature. However, higher cardinality codes have been used with success and there has recently been considerable interest in "real-coded" chromosomes, where the real design parameters themselves are used with no encoding at all [51, 94]. Any cardinality greater than two—including real code—complicates the implementation of the mutation operator somewhat. The complication is not prohibitive, however, and is offset by shorter chromosomes and a more rational parameter-to-gene relation.

The fact that such methods have been known to perform well has prompted researchers to re-examine Holland's schema theorem, which seems to favor low cardinality alphabets as discussed in Section 4.2. Goldberg [34] attempted to explain the paradox by developing a theory of real-coded GA operation, which he called the theory of virtual alphabets. He concluded that real-coded GAs are successful due to implicit internal reduction of high-cardinality alphabets to low-cardinality virtual

alphabets by the selection operator. This explained the successes of real coding while preserving the validity of the schema theorem. In the same paper, he also showed that real-coded GAs are subject to an objective function-dependent phenomenon called blocking, which in certain cases can prevent them from locating optima that are accessible to lower cardinality codes. In keeping with the opinions of the majority of GA researchers and Goldberg's results, binary coding is used exclusively in this research.

Even among binary codings, there are alternatives. Some authors advocate the use of Gray code as opposed to the more traditional base 2 [3, 11]. In the Gray code, any two adjacent base 10 integers differ at only one bit location. Adjacent base 10 integers represented in base 2, on the other hand, may differ at *all* bit locations. For example, $3_2 = 011$ and $4_2 = 100$. Gray code is therefore thought to be more compatible with the mutation operator, as the "flipping" of any one Gray-coded bit is less likely to result in a decoded parameter far removed from the original. Slight performance advantages due to the use of Gray code have been reported in some cases [79, 44]. Regardless, the coding used in this research is exclusively base 2 in keeping with convention. The investigation of Gray coding is left to future research.

Crossover

Crossover is by far the most common mechanism for mixing parental characteristics. It has several variations; the simplest involves the production of two children from two parents using what is called "single point" crossover. A crossing, or cut, location is randomly selected on the chromosomes of the parents. The first child is then defined as the portion of the first parent's chromosome prior to the cut concatenated with the portion of the second parent's chromosome after the cut. The second child is defined by reversing the order of the parents' contributions. Variations include increasing the number of cut points (resulting in multiple, or n-point crossover [33]) and "shuffle-cut-unshuffle" schemes, which are intended to allow any two genes the same probability of being passed together, regardless of the chromosomal distance separating them [70]. The latter process eliminates bias due to gene loci, which in most cases are arbitrary specifications by the programmer.

The advantages, if any, of multiple point crossover have been shown to be small [79]. Holland's schema theorem certainly indicates that schemata disruption should be minimized, and as Goldberg [36] puts it, " ... no convincing evidence of high-order mixing success—empirical or otherwise—has yet been offered." Shuffle-cut-unshuffle, as with many similar proposals in the literature, is intuitively attractive but has little supporting data. In the interest of limiting the departures from standards in

this research to the areas of primary investigation (selection, competitive species and steady-state processing), standard single point crossover is used exclusively here. The only non-standard aspect, made necessary by the species concept, is to disallow a cut point location within or immediately following the configuration gene. Since parents must be of the same configuration, such a crossover point would result in useless clones.

Mutation

Mutation is the simplest of all GA functions, particularly when binary coding is used. When children emerge from the crossover process, but prior to decoding their bit strings back into real design parameters, each bit in their chromosome is subjected to inversion with some (normally very small) probability. This is to maintain some measure of diversity in the population and prevent the permanent loss of any particular schema. In this research, the probability of mutation is set to 0.005; suggested rates in the literature vary from 0.001 to 0.01 [79]. Mutation is prevented from occurring within the configuration gene, which determines an individual's species. Mutating this gene would be the equivalent of biological saltation and has no counterpart in natural processes. In the Darwinian sense, the species here are immutable.[24]

4.3.5 Replacement

Typical GAs maintain a constant number of members in the population. Exceptions do exist in the literature, but variable population size, like variable chromosome length, makes analysis and implementation much more difficult without obvious offsetting benefits [80]. If population size is to remain constant, offspring must replace existing members. In a typical GA, the replacement function is autonomous because the entire population is simultaneously replaced by an equal number of offspring; each new population becomes the next generation. The predominance of generational GAs in the literature is such that the term itself is used only in the few papers that propose alternatives [92, 80].

During the course of one generation, a simple generational GA of population size P calls the selection and reproduction routines $\frac{P}{2}$ times, resulting in $\frac{P}{2}$ pairs of parents which produce two children each. The resulting P offspring are not immediately evaluated and are not considered to be members of the population until all selections and matings are complete, after which *all* new offspring are evaluated and they then

[24] Charles Darwin formulated his theory of natural selection with no knowledge of genetic mutation, and had very little to say about the evolution of full stern submarines. Nevertheless, given the nature of this thesis, it seemed a shame not to cite him somewhere [17].

become the population. A common variation on the generational process is some form of "elitist" strategy, which ensures the survival of highly fit members of the former population at the expense of an equal number of poorly fit children. Otherwise, no continuity necessarily exists from one generation to the next.

In contrast, a non-generational or incremental GA calls the selection and reproduction routines p times per increment, where $1 \leq p \leq \frac{P}{2}$. The children are then evaluated and inserted into the population; the subsequent selection is performed based on the modified population. Obviously, a generational GA may be viewed as simply the special case where $p = \frac{P}{2}$, but there is an important distinction: For any $p < \frac{P}{2}$, the members to be replaced by the offspring (or equivalently, the members which are allowed to survive) must be selected. Thus incremental GAs require an additional function; perhaps this explains their relative scarcity. Given that the processing time for GA functions is usually quite short compared to that of evaluation, however, this requirement should not exclude incremental GAs from consideration. Parent selection routines, which must be present anyway, can be easily adapted to select unfit members for replacement, possibly even concurrently. Also, for purists, an incremental GA is more in keeping with natural population dynamics, as natural populations are not known for mass simultaneous births and deaths.[25]

Intuitively, an incremental process would seem to provide higher quality feedback to the algorithm, as each new selection is made using the latest data available. This observation has been made by Whitley [91], and an incremental process is used in his GENITOR algorithm. Schott also applied incremental strategy in Pareto optimization of fault tolerant systems [80]. There are, of course, critics of this approach. Hancock [41] claims that incremental reproduction inevitably involves the same kind of sampling errors as roulette wheel selection, and a study of selection methods by Goldberg and Deb [35] concludes that the GENITOR algorithm produces very high selective pressure. According to Hancock, however, this pressure is primarily due to perpetual replacement of the worst individual in the population by new children, a feature which is obviously not inherent in incremental strategy.

The GA implementations written for this research investigate the case where $p = 1$; that is, where children are immediately evaluated and inserted into the population. This variation on incremental strategy is known as *steady-state* processing. In addition to being intuitively attractive, this method is chosen to avoid instability when tournament selection is combined with niching. Such instability has been observed and analyzed by Oei, Goldberg and Chang [68]. The remedy recommended in that

[25] The Cambrian explosion and various pre-historic mass extinctions notwithstanding [39].

report, termed *tournament selection with continuously updated sharing*, is very similar in principle to the notion of a steady-state process.

The replacement mechanism chosen here utilizes the three selection sub-routines already in place. For the two ranking methods (Goldberg and Fonseca-Fleming), the two individuals to be replaced are simply chosen by roulette wheel selection based on inverse ranks. For the tournament method, the losers of two independent dominance tournaments are chosen for replacement (ties are again resolved by niche count). Unlike the parent selection process, replacement does not require the two selected individuals to be of the same species. Such stochastic selection of lethals should alleviate some of the high selective pressure associated with GENITOR-type algorithms.

The use of steady-state processing combined with clone rejection makes the probability of crossover term in the schema theorem (4.1) inapplicable here. In typical generational GAs, parents are selected one at a time (by roulette wheel, say) and temporarily placed in a mating pool until all selections are complete. It is therefore likely that highly fit individuals will be represented several times in the mating pool. Those in the mating pool are then paired off randomly, and mated with probability p_c (the pre-defined crossover probability). If crossover does not occur, the parents are simply copied into the next generation after being subjected to mutation. In the steady-state process used here, every mating involves crossover and the parents then compete with all other members of the population to avoid being replaced by the children. All parents are in a sense copied to the following generation, so whether this is equivalent to a p_c of zero or unity is a matter of interpretation but essentially irrelevant.

4.4 Competitive Species

The idea of allowing multiple, non-interbreeding species to compete in a genetic algorithm is an original component of this research. It is intended to allow simultaneous optimization of independent and mutually exclusive options. This is accomplished by simply placing an equal number of representatives from each species (i.e., each option) in the initial population and preventing subsequent inter-species mating. Each species will then multiply or diminish relative to the others based on how well it can adapt to optimality as defined by the selection method and the evaluator.

The need for such a mechanism in the current research is due to the incompatibility of the four propulsor configurations considered. During the initialization process,

the four configurations—rotor-only (one stage), rotor-stator (two stages), stator-rotor (two stages), or stator-rotor-stator (three stages)—are all equally likely, and all active propellers must have their circulation distribution specified. Each specification requires two inputs—a value at the hub and a value at the tip. Thus the two-stage configurations require two more input values than the single-stage configuration, and the three-stage configuration requires four more. Chromosomes for individuals of different configurations must therefore be of different lengths, or must contain unused place-holders.

The chromosome length problem alone does not motivate the species concept. As mentioned in the discussion on initialization in Section 4.3.1, neither variable-length chromosomes nor dummy values cause insurmountable difficulties, although they can be troublesome and inefficient. Rather, the primary motivation for non-interbreeding species in this research is the prevention of invalid selection. The problem, as illustrated in Figure 4-3, occurs when a parent passes on dummy (latent) genes which then become active in the offspring. The parents in this example are of different configurations, and both are assumed to have been selected by some valid, stochastic

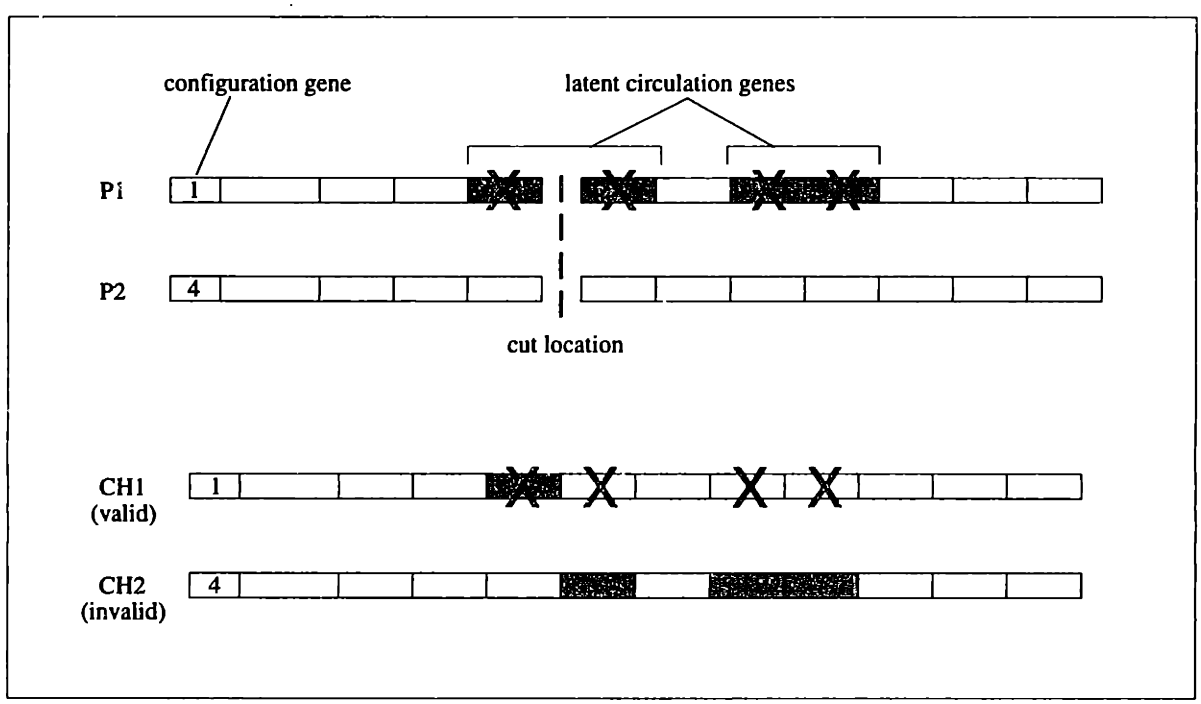

Figure 4-3: Latent genes activated by crossover.

method based on fitness. The first parent (P1) is of configuration 1 (rotor-only), and therefore has arbitrary values of hub and tip circulation for its non-existent second and third propeller stages. These genes are shaded in the figure, indicating that they

are latent, and are marked by an X, indicating that they were ignored by DPLL during evaluation and thus had no impact on P1's fitness value or its subsequent selection. The second parent (P2) is of configuration 4 (stator-rotor-stator) and therefore has active values of all six circulation genes. If P1 and P2 are mated, and the random crossover point falls between the configuration gene and the final circulation gene as shown, the result is one child of each configuration. When these children are evaluated, DPLL will ignore the valid but extraneous circulation genes which the rotor-only child (CH1) has inherited from P2. This is, at worst, a bit inefficient; CH1 is a valid mixing of the parents' parameters. What makes the mating invalid is the passing of latent, arbitrary genes from P1 to CH2 where they then become active, since CH2 is of configuration 4. The problem is worse if the cut location happens to fall within the configuration gene itself, as the children will then be of different configurations than either parent.[26]

This problem is resolved by the multiple species concept, which is intended to allow the four incompatible configurations to be optimized simultaneously in less time than would be required for separate optimizations. A first parent is selected from the general population without regard to species, using some standard stochastic method based on fitness. Selection of its mate, however, is restricted to members of the same species. The same selection *method* is used for the second parent, but only individuals of the same species as the first parent are eligible—the potential mating pool is segregated. A minor difficulty with this process, namely what to do if only one member of a species remains in the population and happens to be selected, is easily overcome by implementing an "endangered species" strategy. This simply spares the last *two* representatives of any species from extinction; however, it is probably not crucial to overall results. Any species which has been reduced to only two remaining representatives is obviously not well-adapted and is unlikely to play a role in further convergence of the algorithm.

Certainly the four propulsor configurations considered here could be optimized independently using custom chromosome lengths and a modified algorithm for each. This would result in a unique Pareto frontier for each configuration. These frontiers could then be overlaid, if desired, to obtain a "meta-frontier" having configuration as a location-dependent characteristic rather than a fixed, global parameter (see Figure 4-4). This "frontier of frontiers" would present all the information needed—in condensed form—for making a final decision, assuming that no external reasons exist

[26] It is certainly possible that, by accident, such a child would turn out to be highly fit. On average, however, it would be inefficient to spend time evaluating such "illegitimates." Injecting arbitrary bits into the selection process is the job of the mutation operator.

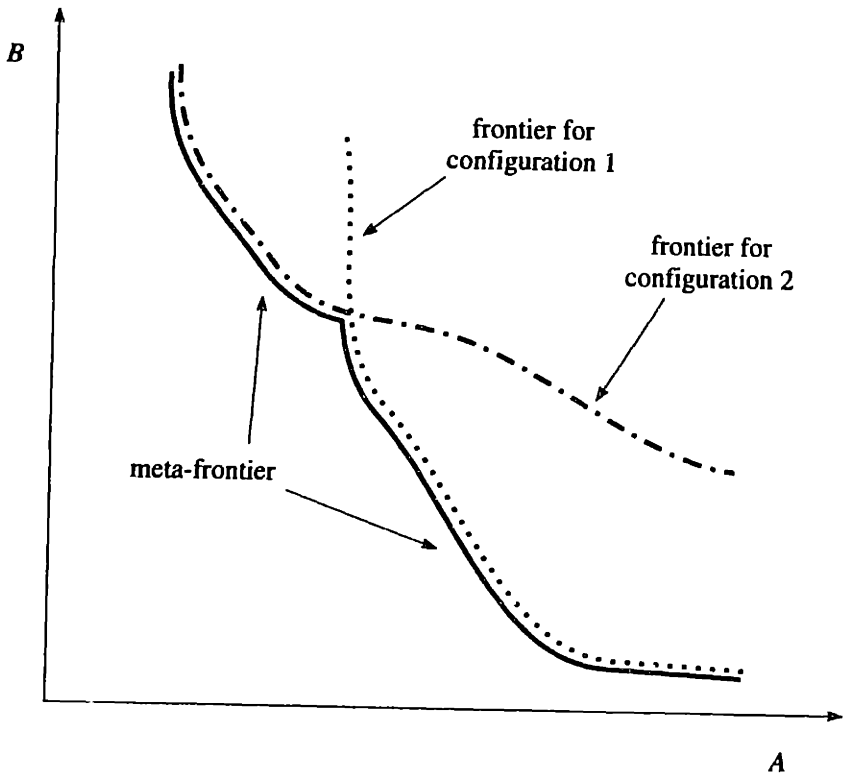

Figure 4-4: Notional Pareto meta-frontier, with objectives A and B to be minimized.

for preferring one configuration over another.

Production of the meta-frontier while simultaneously solving the latent gene problem is the motivation behind the competitive species concept; elimination of inferior species from consideration as the population evolves should make this a more efficient approach overall than performing separate optimizations. The concept may be generalizable to include any design or decision where mutually exclusive and GA-incompatible options exist.

Chapter 5

Results

5.1 Comparison of Selection Methods

Three versions of the Pareto GA described throughout Chapter 4 are run to the completion of 3500 cost function evaluations (CFE). These versions differ only in the selection method they employ; the selection methods themselves are described in Section 4.3.3, page 107. The cost function is DPLL v2.0; the Pareto objectives are maximum internal volume and minimum power coefficient for ducted propulsor submarines. Internal volume corresponds to stern fullness, as all variants are of identical length and diameter and use a common bow profile. All results documented here, including the three-objective results of Section 5.3, are obtained using a population size of 200 and all are started from the same randomly generated population.

5.1.1 Final Populations

Figures 5-1, 5-2 and 5-3 show objective space plots of the final populations produced by the three versions. The horizontal axis is normalized internal volume (as defined in Appendix B) converted to a minimization. Greater volume (increasing stern fullness) is toward the origin. The vertical axis is power coefficient, with a smaller value indicating that less power is required to maintain the given forward speed. Variants identified as infeasible in the plots are those for which flow separation occurs forward of the rotor.

A prominent feature of these plots is the tendency of the solutions to align vertically at discrete values of volume. This is a result of parameter discretization and is not a characteristic of the GA. Due to the way this particular optimization is set up, the volume objective happens to be strongly dependent on one input parameter—fullness factor—and weakly dependent on all others (via the gear and shaft volume penalties described in Appendix B). Since fullness factor, like all input parameters,

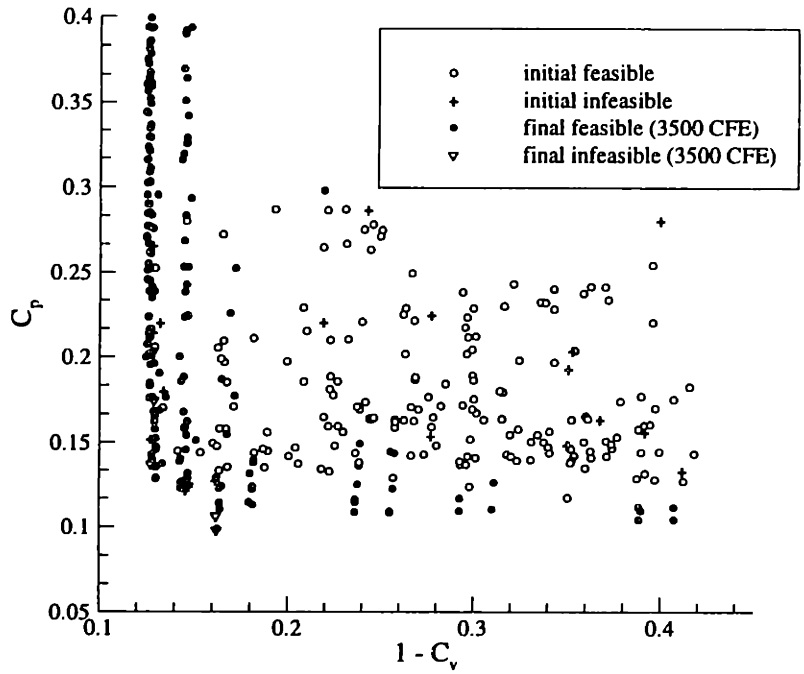

Figure 5-1: Initial and final populations from Goldberg ranking method.

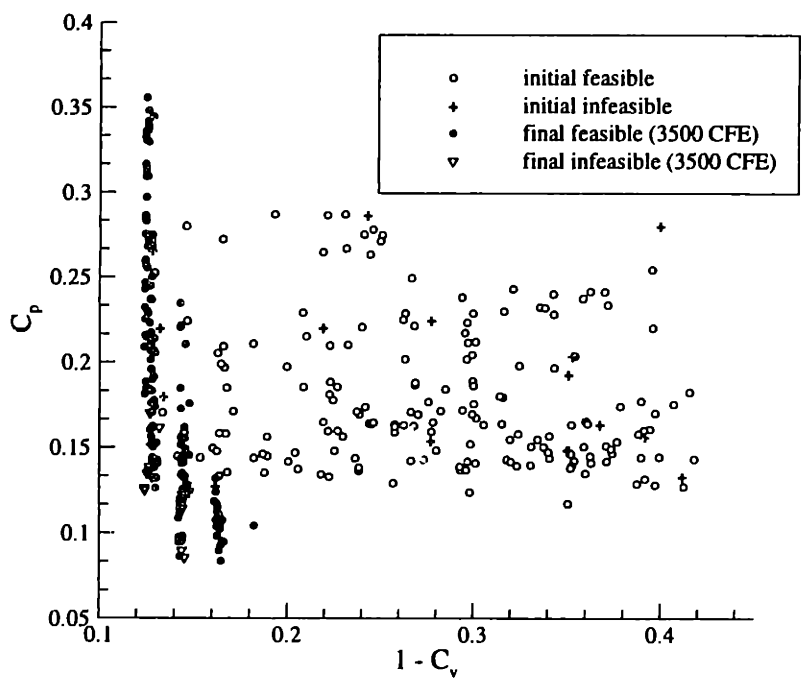

Figure 5-2: Initial and final populations from Fonseca-Fleming ranking method.

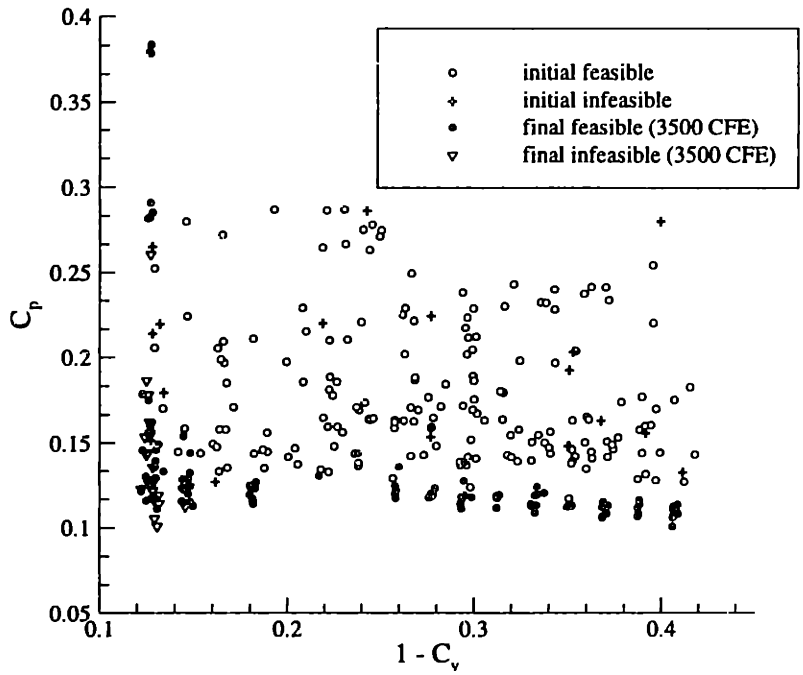

Figure 5-3: Initial and final populations from tournament selection method.

is necessarily discretized, the evaluated volumes are also somewhat discretized. This observation prompted the higher resolution of fullness factor relative to the other inputs as mentioned in the introduction to Section 4.3.1.

It is also apparent that DPLL's power-volume frontier is constrained by the maximum stern fullness allowed in the input. Solutions lying on the constant-volume border on the left side of the plots are "weakly" Pareto optimal—they all have nearly the same value of one objective. Weak crowding is not a desirable condition in a Pareto GA, as it reduces population diversity without supplying any additional information.[1] Niching penalties alone, at least as they are used in this research, do little to alleviate weak crowding; apparently, the non-dominated status of these solutions overcomes their penalties due to niching and they remain likely candidates for selection. Tolerances on objective values during dominance checking, intended to make some weakly optimal solutions appear dominated, are incorporated in the algorithm but are evidently unsuccessful at reducing the effect.

Generally, a frontier constrained by parameter ranges would indicate that these ranges should be increased or adjusted. In retrospect, it is possible that increasing

[1] This observation is the author's own; none of the Pareto GA papers reviewed mention weak crowding.

the upper limit on fullness factor in this research would have reduced the weak optimality stacking effect and provided a more complete frontier. On the other hand, the maximum fullness allowed here is already quite extreme compared to operational designs (see the outer envelope in Figure 4-2) and, since it appears that the minimum power coefficient increases rapidly with stern fullness when $1 - C_V < 0.14$, sterns of greater fullness than the maximum allowed here would probably be very inefficient.

Comparison of the final population plots from the three methods reveals some disagreement regarding the form and location of the non-dominated frontier. Both of the ranking methods produce a frontier of very limited span consisting exclusively of high-volume variants. The Fonseca-Fleming method appears to have located variants of greater efficiency than the Goldberg method, while the Goldberg method retains greater volume-diversity in its final population. The general location of the frontier, however, is the same for both. Tournament selection maintains the best volume-diversity of the three, but this is probably a consequence of the fact that it fails to locate the global power minimum at $1 - C_V \approx 0.16$, which would otherwise dominate most of the objective space. As will be seen when the frontier's composition is presented in Section 5.2, this is an isolated minimum in terms of propulsor configuration, making it difficult for a GA to locate (and, incidentally, *very* unlikely to be located by a hill-climbing method, without some improbably accurate prior assumptions).

The fact that the true frontier—taken hereafter to resemble that produced by the Fonseca-Fleming version—seems limited to very full sterns is hydrodynamically interesting and will be discussed in Chapter 6, but leaves much of the (presumably dominated) feasibility boundary unresolved. The form of the *entire* boundary, including the dominated regions, would be useful in supporting the validity of the frontier and in learning how minimum attainable power varies across the entire range of stern fullness. Fortunately, some indication of the dominated boundary's location is available from the cumulative information produced throughout the GA's iterations.

Figures 5-4, 5-5, and 5-6 are the results of overlaying all feasible variants generated throughout 3500 reproduction and evaluation cycles and tracing the boundaries of the resulting scatter in objective space. This procedure reveals the Pareto frontier in each case as well as an approximation of the dominated feasibility boundary. Resolution of the region to the right of the minimum at $1 - C_V \approx 0.16$ is relatively poor for both of the ranking methods because it is dominated and the GA allocates most of its computation time to the non-dominated frontier. The tournament method, however, in failing to locate the minimum, provides corroborating evidence of the lower-volume feasibility boundary as it conducts a much more extensive search in

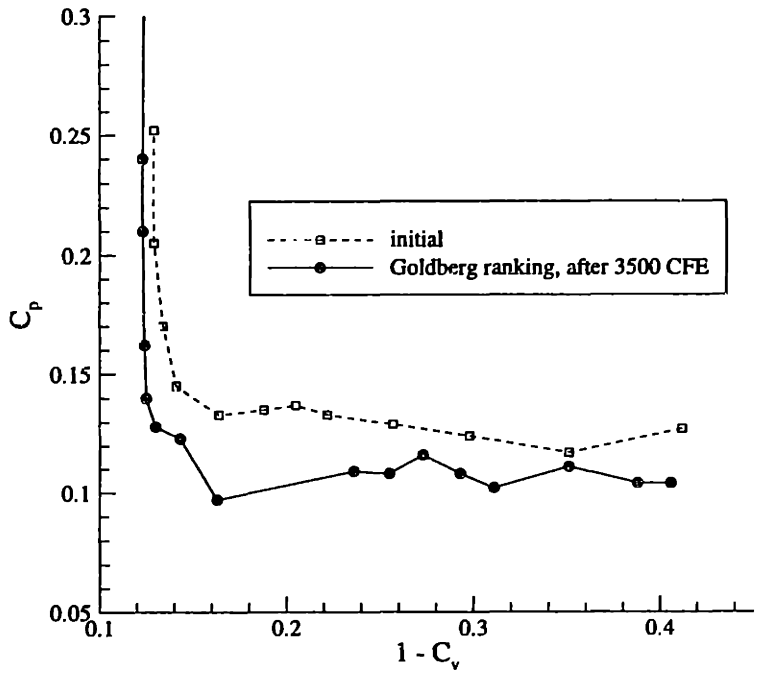

Figure 5-4: Feasibility boundary from Goldberg ranking method.

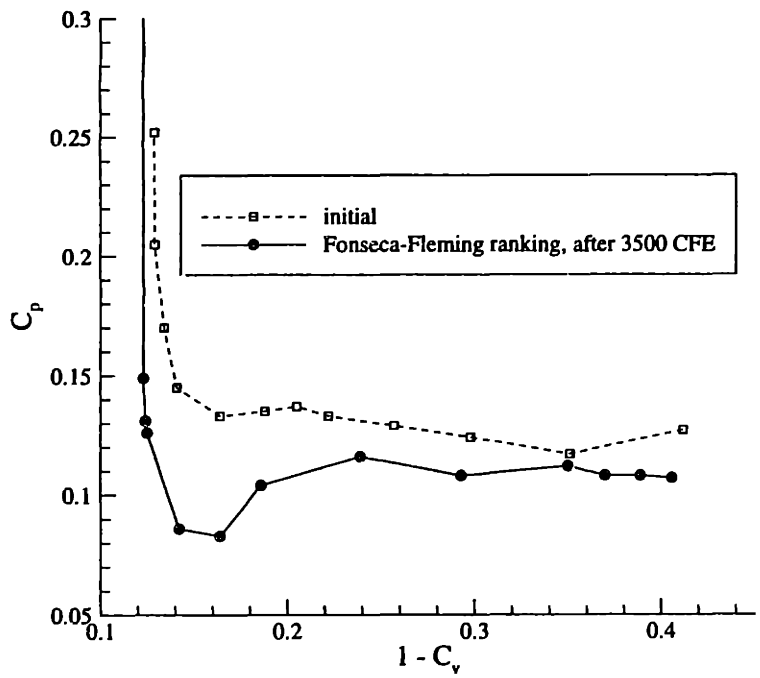

Figure 5-5: Feasibility boundary from Fonseca-Fleming ranking method.

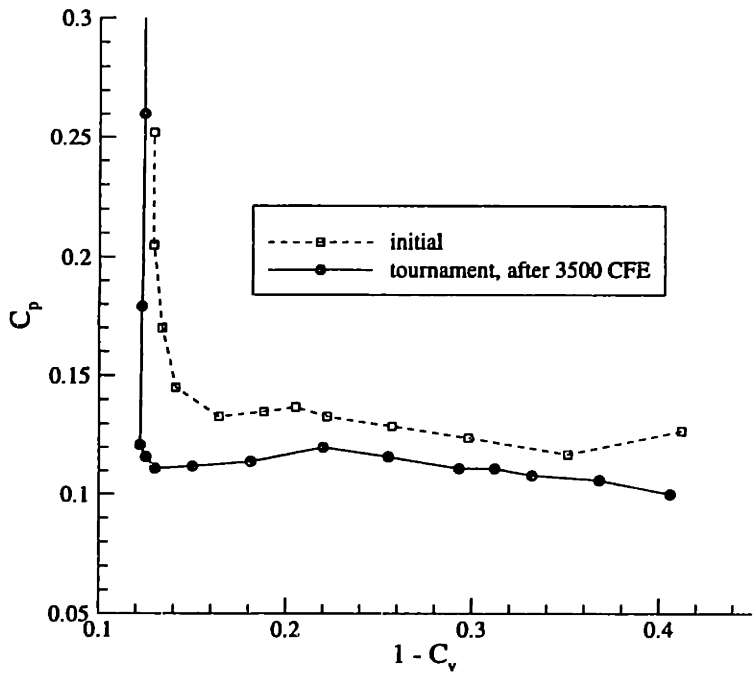

Figure 5-6: Feasibility boundary from tournament selection method.

this region. These incomplete but informative results confirm a slight downward trend in power coefficient for very slender sterns which is barely noticeable in the feasibility plots of the ranking methods.

5.1.2 Convergence Analysis

Defining a measure of performance for a Pareto GA is a difficult matter. Simply counting the number of non-dominated solutions in the population after a given number of CFEs is reasonable, but does not account for the diversity or quality of the solutions. Even initial, randomly generated populations have several non-dominated members if domination is defined only in terms of contemporaries, which it must be since the location of the true frontier is never precisely known. The number of Pareto solutions found, therefore, should not be used as the sole criterion for judging performance. The quality of a Pareto GA's frontier actually depends on several criteria, including accuracy, resolution, range (i.e., how closely the true limits are approached), and density distribution.

There are few suggestions in the literature for quantifying the performance of Pareto methods in general. Schott [80] proposed the variance of the Euclidean dis-

tance (in objective space) between adjacent frontier solutions as a measure of a Pareto GA's ability to spread the population evenly. This is certainly a valid measure in general, but is not appropriate for this particular problem. Here, the discretized objective space combined with a very limited frontier span produces misleading variance values. Schott also proposed a "7-point distance measure," where points on the objective axes are pre-selected and the distances from each point to the nearest feasible member of the final population are averaged. As with variance, this is a measure more suited for continuous frontiers defined by a statistically significant number of solutions, and is not well-suited to this particular problem.

What *can* be used here to compare the three selection methods is qualitative analysis of the final population and feasibility boundary plots presented above, supported by some simple quantitative measures. Figure 5-7 shows characteristics of the pop-

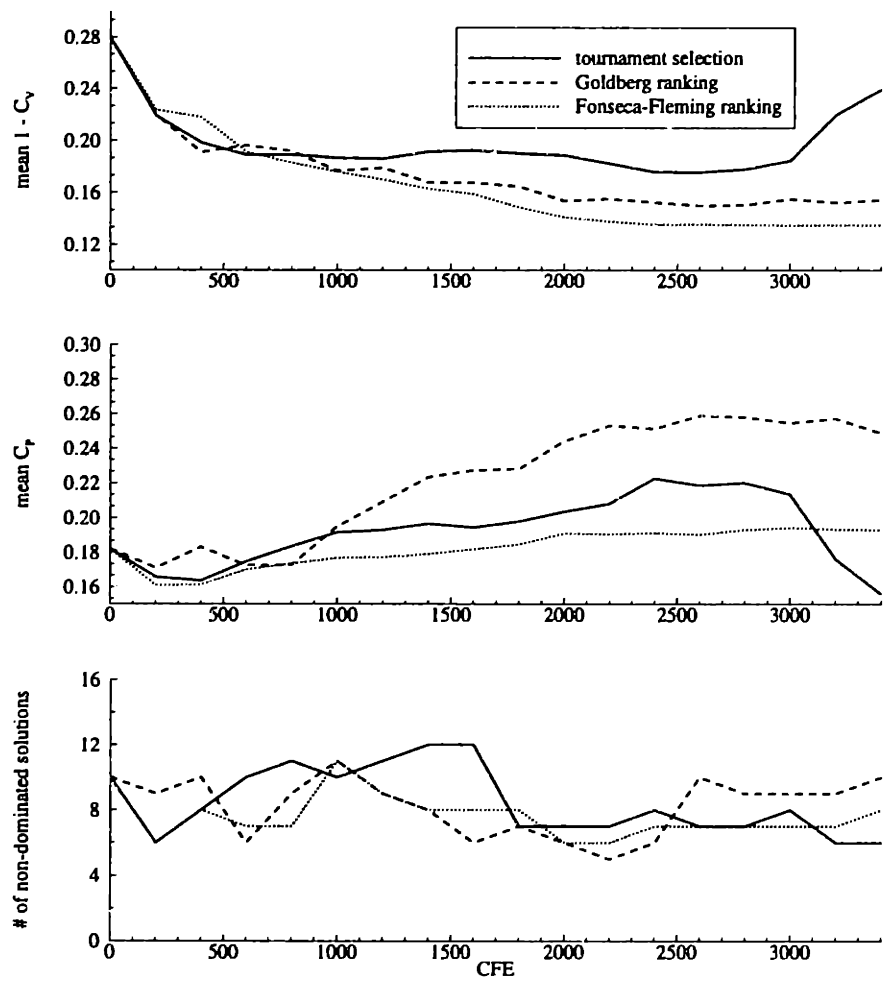

Figure 5-7: Comparison of frontier dynamics for the three selection methods. Mean volume, mean power coefficient, and number of non-dominated solutions found vs. cost function evaluations. All quantities shown involve feasible variants only.

ulation, as functions of the number of cost function evaluations, for each method. The upper plot tracks the mean volume of all feasible individuals in the population, the middle plot tracks mean power, and the lower plot tracks the number of non-dominated solutions as the population evolves.

Both ranking methods show steady trends in mean objective values; they appear to have reached a steady state after approximately 2300 CFE. The number of non-dominated solutions, as expected, are erratic and provide little means for distinguishing among the methods in this application. Tournament selection shows no real trend in mean values and its population is undergoing rapid change at the end of 3500 CFE. This is probably related to the proportion of infeasibles in the population, which for the tournament method begins to increase rapidly around 3000 CFE.[2] The number of non-dominated solutions, as with the ranking methods, is erratic.

Qualitatively, the ranking methods obviously outperform tournament selection. In addition to their steadier trends in population characteristics, they both identify an important region of the frontier which is missed by tournament selection. Between the two ranking methods, Fonseca-Fleming ranking appears to provide a fuller, smoother frontier than Goldberg ranking and, most importantly, identifies several feasible variants which would dominate those on the Goldberg frontier if the final populations from the two methods were overlaid.

Although Fonseca-Fleming ranking appears to out-perform the other selection methods in all respects, some caution is in order when drawing conclusions from these results. Due to the considerable processing time required by this evaluator (DPLL) and the time limitations on this research, the total number of CFEs for each method is quite small compared to most applications in the literature. It is possible, for instance, that tournament selection was close to locating the global power minimum when it was terminated, and that it would have soon settled there and produced a "better" frontier than either ranking method. The results of several undocumented preliminary runs, however, indicate that the tournament method does seem to have more difficulty in locating this inconspicuous optimal region.

5.2 Composition of the Feasibility Boundary

As seen in the previous section, the local position and shape of the final feasibility boundary, including the dominated region, varies among the three selection methods.

[2] It is possible that the dynamic constraint penalty is somehow incompatible with tournament selection as it is implemented here. The rapid increase in proportion of infeasibles was noted for several of the most recent revisions of the tournament selection subroutine.

Despite local differences, however, all methods generate very similar solutions—in terms of parameters, that is—at equivalent *positions* along the boundary. Figure 5-8 shows a boundary in its (assumed) continuous form, based on the combined results from Section 5.1.1. Analysis of interim and final populations indicates that the boundary consists of three zones defined by propulsor configuration.

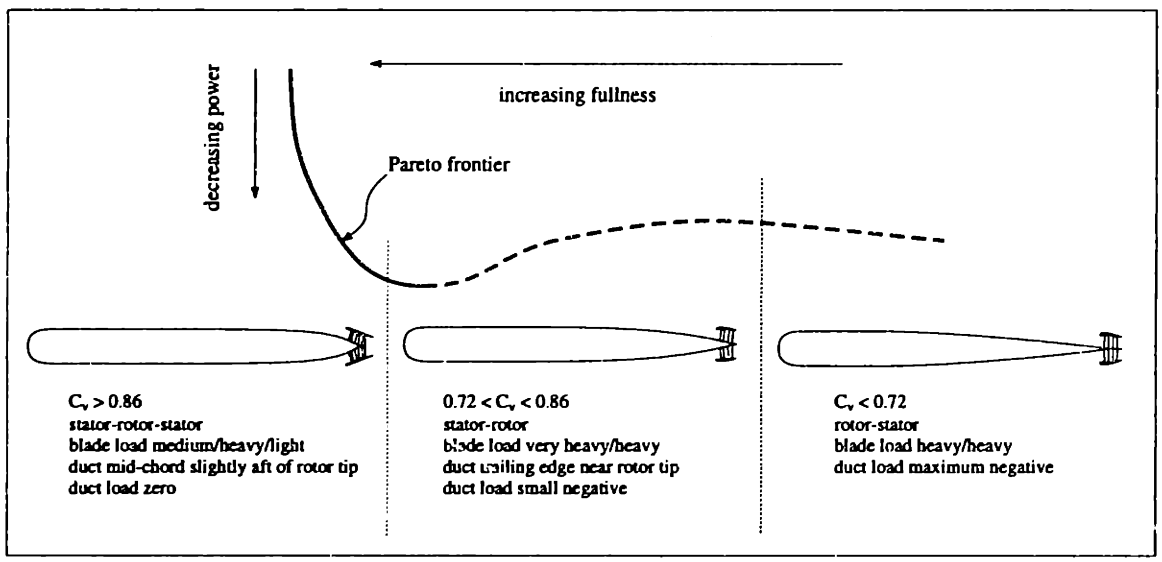

Figure 5-8: Feasibility boundary composition, from combined results of the three selection methods.

For gently tapered afterbodies with low volume coefficients ($1 - C_V > 0.28$), the rotor-stator configuration is predominant on the boundary—that is, all of the best (lowest power) solutions found in this region throughout the GA's iterations are rotor-stator. Circulation values on the blades of both stages are generally high, but do not appear to be limited by the input range. Ducts in this region have a very strong tendency toward mid-range chord length ($\approx 0.035L$) and toward the maximum negative load allowed ($G_{\text{duct}} = -0.04$).[3] Mean tip radius is around $0.033L$, with little variation. Duct axial position with respect to the rotor tip is variable.

For slightly fuller sterns (the middle zone in the figure), the best propulsor arrangement in terms of minimal power seems to be stator-rotor. Solutions in this region of the boundary tend to have stators with maximum allowed hub circulation and near-maximal tip circulation. Rotor hub and tip loads are also high, but slightly less, on average, than those of the stator. Duct position relative to the rotor tip tends strongly toward the most upstream location allowed (i.e., rotor tip at 80% of duct chord); however, duct chord length itself is variable in this zone. The mean duct load is negative but of lower magnitude ($G_{\text{duct}} = -0.02$) than in the first zone and the tip

[3]Negative load corresponds to decelerated flow through the duct and a lift component opposing rotor thrust.

radius is slightly decreased. This zone extends to the left past the global power minimum at $1 - C_V \approx 0.16$ and therefore makes up a small portion of the non-dominated frontier.

The third zone, where $1 - C_V < 0.14$, is composed entirely of stator-rotor-stator arrangements. This zone is entirely non-dominated, although much of its span is weakly optimal. Circulation values on the upstream stators are generally of mid-range, those of the rotor are somewhat higher (similar in magnitude to the rotor values in the middle zone), and those of the downstream stator are below mid-range and show very little variance. The duct tends to be positioned such that its mid-chord point is slightly downstream of the rotor tip. Mean duct chord length is $0.05L$, slightly above the median value allowed, with very little variance. Duct load shows essentially zero variance with a mean of $G_{\text{duct}} = 0$ (no net load, but possibly inducing velocities near the leading edge to prevent flow separation). Tip radius, as in the other zones, averages around $0.03L$ but shows a slight tendency to increase with stern fullness. Most of the non-dominated variants in this zone border on separation.

5.3 Three-objective Pareto Optimization

Pareto optimization is not limited to two-objective problems; the definition of Pareto optimality is applicable to any number of objectives greater than one. Two-objective spaces happen to be the easiest by far to display and interpret and are therefore seen in the majority of examples and test cases. In general, the Pareto frontier is a surface of dimension $n - 1$, where n is the number of objectives considered.

This section documents the performance of the Pareto GA in defining a two-dimensional frontier, with minimal cavitation index (defined in Appendix A) included as the third objective. The changes involved in converting the GA from two objectives to three are minor; only the selection subroutines and some minor input/output features are affected. The starting point is the same initial population used for all the two-objective optimizations presented above, with a population size of 200.

Figure 5-9 shows the resulting two-dimensional Pareto surface after 2900 CFE, when 24 non-dominated solutions exist in the population. These appear as the white squares in the figure; the surface is interpolated from these points.[4] Note that the axes of this plot are the same as those for the two-objective results, and that the locations of the frontier points correspond closely to the feasibility boundary plots of Section 5.1.1.

[4] This interpolation was done with Tecplot v7.0, using the "kriging" scheme.

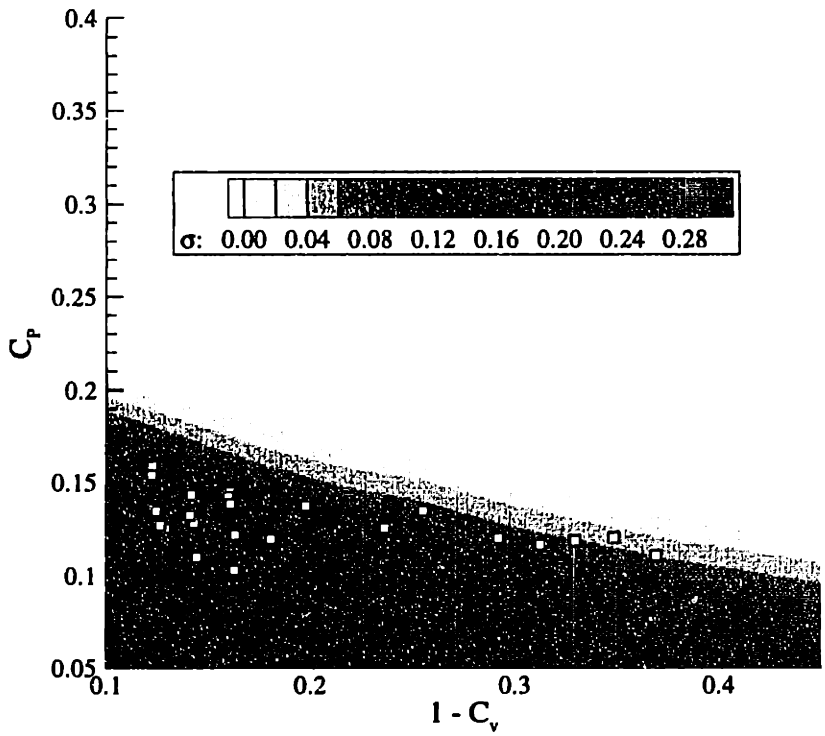

Figure 5-9: Non-dominated frontier for three objectives, using Fonseca-Fleming ranking method. The symbols are the non-dominated points after 2900 CFE. The cavitation surface is interpolated from these points.

Analysis of the final population shows that all of the non-dominated solutions in the region $1 - C_V > 0.18$ are rotor-stator configurations. These lower-volume variants account for 16 of the 24 solutions in this Pareto set, and include the two lowest cavitation indices found. Two stator-rotor configurations are present at $1 - C_V \approx 0.15$ and have the lowest power coefficients in the set, but not as low as those of the two-objective minima presented above. Presumably, further iterations would locate these points, as any solution which is non-dominated in two objectives must remain so when a third is added. For larger values of volume, stator-rotor-stator variants again predominate; however, at the extreme upper limit of volume are two solutions having a rotor-only configuration. In general, lower cavitation indices appear to correlate to rotor-stator arrangements and higher indices to stator-rotor-stator arrangements. Tip radii throughout the set are below mid-range ($\approx 0.025L$) and blade loading is relatively light compared to that of the two-objective set. Duct load shows a solid trend from slight negative for low-volume variants to mid-range positive ($G \approx 0.05$) for the fullest sterns with rotor-only propulsors.

Chapter 6

Conclusions

6.1 Hydrodynamic Issues

When drawing hydrodynamic conclusions from the results of Chapter 5, it must be kept in mind that what is being optimized here is the DPLL *model*. While much effort is made in this research to ensure and verify the revised DPLL's accuracy, it remains a preliminary, exploratory code and is subject to its assumptions and simplifications as put forth in Chapter 2.

6.1.1 Objective Relationships

The two-objective results presented in Chapter 5 strengthen the case for full stern submarines and support the first thesis of this research. Optimization of the DPLL model indicates that the power required to propel a submarine at steady speed does *not* increase with stern fullness if the propulsor configuration is matched to the hull profile. In fact, it may be that *less* power is required for fuller sterns, and there may be an optimal stern profile, quite full, which minimizes required shaft power.

The three-objective results indicate that the likelihood and severity of cavitation generally increases with stern fullness and propulsive efficiency. Based on interpolation of non-dominated solutions, it appears that a power trade-off is a more efficient means of reducing cavitation than a volume trade-off.

It may be possible in general to extract frontiers of lower dimension from surfaces of higher dimension; for example, the three-objective Pareto surface of Section 5.3 might allow determination of the power vs. cavitation frontier. This could be done by sorting the final non-dominated solutions in terms of two objectives, without regard to the third, and extrapolating the results. It is unclear how the accuracy of such a process would compare to that of an explicit, two-objective optimization. This potentially useful aspect of Pareto optimization is not investigated here.

6.1.2 Frontier Composition

The most interesting features of the frontier and its extension, the dominated feasibility boundary, are the changes in propeller arrangement and duct loading with stern fullness. Following are possible hydrodynamic explanations for the trends noted.

Flow near the stern should become more radial as the stern becomes fuller and the velocity magnitude should decrease due to the increasing effect of the wake fraction. Radial contraction intensifies any tangential velocity present; this is presumably the reason that compound propellers, which can take advantage of this effect, make up the entire frontier. The decelerated flow (wake fraction) caused by viscous effects and negative duct load should increase the rotor's efficiency, but also makes separation of the hull boundary layer more likely. The trend from negative duct loads for low-volume variants to zero load for full sterns evidently results from a tradeoff between these two effects. Since separation is not a problem for the low-volume, gently tapering sterns, a relatively large negative duct load can be used to reduce the rotor inflow velocity. As stern fullness increases, however, negative loads of the same magnitude result in separation and are therefore not feasible. This loss of potential rotor efficiency is compensated by moving the rotor downstream of the stator. In this location, it can benefit from the increase in stator pre-swirl provided by radial contraction. The trend toward zero duct load continues as the stern becomes very full, and zero load predominates among Pareto solutions at the upper limit of volume. In this zone, any attempt to decelerate the rotor inflow is likely to result in separation. This further loss of potential efficiency is compensated by an additional stator. Being limited in the magnitude of its circulation values, the rotor is unable to utilize all of the tangential velocity imparted by the forward stator and magnified by the radial contraction. The downstream, lightly loaded stator probably captures enough of this otherwise wasted energy to overcome its viscous penalty.

It should be noted that the high-volume Pareto solutions in all of these results are generally very near separation. In moving from an exploratory optimization such as this research provides to more detailed analysis, some margin of safety should be implemented to exclude borderline solutions which might separate when such things as surface roughness, appendage effects, and unsteady forces are accounted for.

6.1.3 Power Density on the Frontier

The two-objective results indicate that for given length and forward speed, a fuller stern (and therefore greater volume) does not require additional shaft power and may in fact require less, depending on the propulsor configuration. In practice, this

advantage would likely be used to shorten the overall length of the submarine while maintaining the *same* volume, rather than simply making the stern "roomier" inside. Warren has investigated such shortening and found it to be practical in terms of arrangements [88].

Figure 6-1 shows the final population from the two-dimensional Fonseca-Fleming ranking method overlaid with lines of constant power per unit volume. Clearly, min-

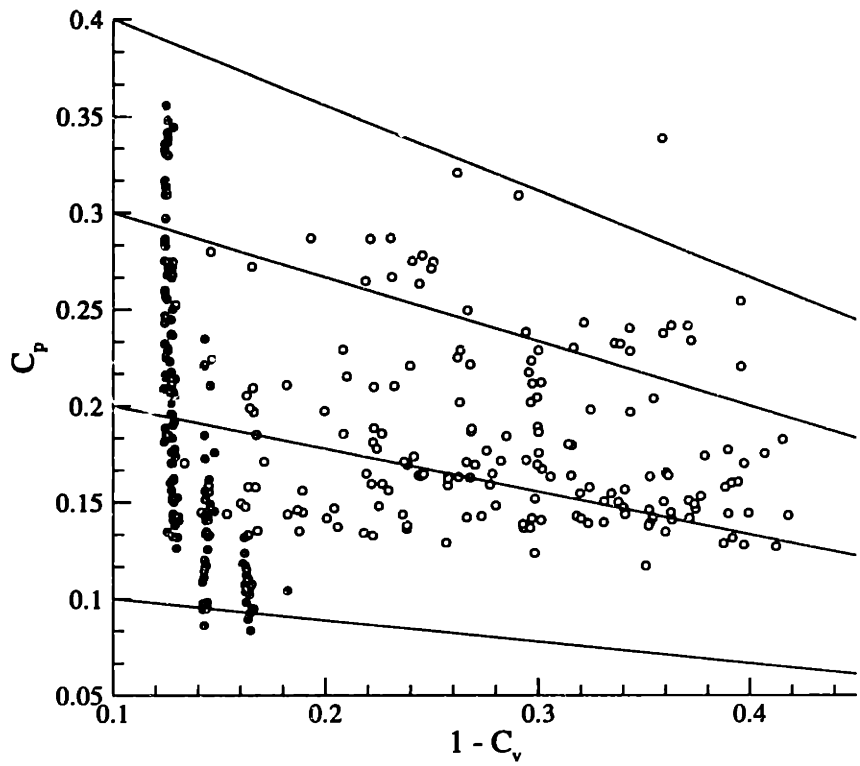

Figure 6-1: Initial and final feasible populations from Fonseca-Fleming ranking method, overlaid with lines of constant power density.

imum "power density"—or greatest volume propelled per unit power—of all feasible solutions is located near the global Pareto power minimum at $1 - C_V \approx 0.16$. It is tempting to conclude that this point represents *the* optimum, but note that minimization of power density is simply a scalarization and should not be used independently of a Pareto method. In this particular case, minimum power density simply provides a logical means of choosing from among the solutions on the frontier and eliminates the weakly optimal portion of the frontier from consideration.

6.2 Optimization and Decision-Making Issues

6.2.1 General Comments

The second thesis proposed in Chapter 1—that knowledge of the solution space is necessary for informed decision-making—is also validated; the dominance of fuller sterns over those which are more tapered is certainly non-intuitive. Such configurations would probably not be investigated without some indication of their potential, as is provided by the frontier.

It is notable that the ratio of cost function evaluations to design space size used here is quite small relative to most published applications. The two ranking methods are reasonably stable after 3500 CFE, which is less than one ten-millionth of the design space possibilities. This illustrates the power of implicit parallelism in exploring objective space and exploiting known solutions.

The process of defining the dominated feasibility boundary by the cumulative results of the Pareto GA is somewhat cumbersome and imprecise, but does provide supporting data for the final frontier at no additional computational cost. Definition of this boundary may be useful in general Pareto optimizations where the frontier itself is quite limited, as it is here.

The full stern power minimum discovered by the two ranking methods is isolated in terms of propulsor configuration. The majority of the Pareto frontier is composed of stator-rotor-stator variants; only a small portion, at the lower limit of power coefficient, is made up of stator-rotor variants. Such a point would be very difficult for a gradient-type method to locate, particularly if the four different configurations were optimized simultaneously as they are here.

The three-objective frontier produced by the GA shows how the Pareto concept can be extended into as many dimensions as necessary. The results indicate that the objective of minimal cavitation is incompatible with both efficiency and stern fullness in terms of the DPLL model. In fact, cavitation index is at a maximum in the region of the two-objective Pareto frontier. This is likely due to the predominance of high rotor blade loads on the two-objective frontier, which tends to result in high cavitation indices according to Equation A.1.

The algorithm would be perfectly capable of evaluating four or more Pareto objectives, but problems would be encountered in displaying and interpreting results. Problems of presentation could possibly be resolved by resorting to "slices" through the frontier where one or more objective values are held constant. For several objectives, however, interpretation becomes the primary obstacle. Each additional objective

provides the members of the population with another way of being non-dominated. Eventually the Pareto set becomes so large and diverse that it is of little use to a decision-maker. Trends and knees, two of the most useful results of Pareto optimization, become lost in the information overload. This research makes no particular attempt to resolve the problems of numerous objectives, other that to propose that in some cases the combining of low-level objectives into two, three or possibly four meta-objectives may be necessary. Analysis of the results may then indicate better ways of combining the objectives to expose trends and knees.

6.2.2 Selection in Pareto Genetic Algorithms

Based on the limited results of Chapter 5, the Fonseca-Fleming ranking method appears to give the best performance in this application. This conclusion is supported both quantitatively by the population's trend is objective values and qualitatively by inspection of the final populations as plotted in objective space. Tournament selection is somewhat erratic and is apparently prone to a runaway proportion of infeasibles, despite the dynamic penalty threshold implemented here.

The Pareto GA and its selection subroutines developed during this research went through two years of revisions and corrections. The results shown in Chapter 5 are from the most recent versions, which were not completed in time to allow numerous independent runs, with varying random number seeds, for each selection method. Average results over many such runs would be necessary to draw firm conclusions regarding the relative performance of the selection methods. However, the final results presented are very typical of those seen throughout the development of the GA. In other words, despite the limited performance sampling presented here, there is a good deal of cumulative, preliminary evidence that the Fonseca-Fleming ranking method outperforms the other two, and outperforms tournament selection by a considerable degree.

6.2.3 Competitive Species

The competitive species concept originated here is very promising. Its application reveals that the optimal number, type and relative positioning of propeller stages are dependent on stern fullness; the meta-frontier of the DPLL model has in fact been identified. Undoubtedly, compilation of the meta-frontier via separate optimizations of all configurations would be less time-efficient, although this assertion is not proven by direct comparison in this research. Such a comparison would require separate, tailored versions of the GA for operating on the different configurations' chromosome

lengths. This would be an interesting experiment, however, particularly if the results were overlaid (e.g., Figure 4-4) for comparison to the meta-frontier obtained here.

Analysis of the final populations from all three selection methods reveals that each propulsor configuration is present in approximate proportion to the percentage of the boundary it comprises. This indicates that the configuration zones are valid, insofar as the DPLL model is concerned, and that the solutions there are not sub-optimal accidents residing on a false boundary due to lack of sufficient exploration.

6.2.4 Miscellaneous

Penalties due to niching and constraint violation seem to be correctly applied, judging by the composition of the final two-objective populations of Chapter 5. The number of infeasible solutions in the population remains fairly constant for the two ranking methods, supporting the validity of the dynamic penalty scale described in Section 4.3.3, page 106. The tournament method typically experiences a sudden, uncontrolled rise in the proportion of infeasibles at some point during the iterations. This is probably due to some incompatibility between this type of selection and the dynamic constraint penalties applied, but the effect has not been investigated in detail. Crowding is most evident at the weak Pareto boundary, but is probably due in part to the somewhat discrete objective space of this problem. It is not clear that this phenomenon can be prevented in the general case without some application-specific tailoring.

6.3 Future Work

As with most time-limited projects, there are areas here which warrant further effort or investigation. These fall into two general categories: improvements and modifications to DPLL, and further research into the use of Pareto GAs for optimization and decision-making.

6.3.1 Evolution of DPLL

Following are possible modifications to DPLL which in the author's estimation could result in greater accuracy, speed, and robustness.

1. A front-end routine is needed to convert body and duct geometries to B-spline vertex files for DPLL input. Modeling an existing body and/or duct in the current version requires tedious trial and error to get the vertices correct. The user

should be given the option of providing vertices *or* offsets; conversion, if necessary, should take place internally. This will probably require some interaction, since the best possible fit using a given number of vertices varies according to the complexity of the prescribed curve.

2. The body model should be converted from submerged source rings to a paneled surface. In the current version, source ring strengths can be erratic depending on the number and spacing used. Converting to a panel method may result in better pressure drag calculations, which have been somewhat suspect throughout this research.

3. The duct and blade representations should be given thickness distributions, by adding sources to the system matrix which solves for singularity strengths. The additional computation load should not be significant.

4. The duct boundary layer should be modeled, in a manner similar to that of the body, to predict leading edge cavitation or separation.

5. Tip gap and blockage effects should be included. In keeping with the intended use of the code, a parametric estimate is probably sufficient.

6. Linear circulation distributions on lifting lines are restrictive. Goldstein calculations should be modified to allow specification of individual circulation values on all segments of the lifting lines.

7. The body B-spline should update at each iteration so that the radius of the sting equals the calculated radius of the hub vortex. This will prevent wake streamlines from entering the hub vortex region, a condition which currently must be monitored and corrected manually.

8. Submarine propeller blades are usually quite skewed; DPLL's lifting line model should be modified to allow more precise specification of skew and rake.

9. More validation against published data is needed.

6.3.2 Optimization

Population size was originally intended to be a variable in this research, but selection method was considered more important and there were insufficient resources of time and hardware to investigate both issues. It would be interesting to know the GA's convergence rate as a function of population size, and to determine the impact of the multiple species on optimal population size.

Two parameter ranges—duct load and stern fullness—apparently constrained the frontiers in this research. Although the allowed limits on both of these seem to already be quite extreme, a non-constrained frontier would be interesting. Properly adjusted parameter ranges might eliminate the weak crowding observed in the two-objective frontiers. If not, some mechanism should be devised for reducing this effect. One possibility is to apply negative dominance tolerances to solutions below the first dominance layer, thus increasing their apparent dominance count.

The results given by this exploratory model should be investigated with more sophisticated codes. In particular, the full stern, stator-rotor variants with slight negative duct load, which represent the apparent global power minimum, should be analyzed in greater detail.

The Pareto GA used here is not the only possible method for locating non-dominated solutions, although it seems to be the only way to locate several in a single run. This method should eventually be compared in terms of efficiency and efficacy to some baseline, such as stochastic hillclimbing, and thereafter to other alternatives, such as systematically varied objective weighting.

Appendix A

Cavitation Index

Farrell and Billet [25] cite three types of cavitation which occur in the tip region of an axial-flow pump: gap, blade-end and leakage vortex cavitation. Gap cavitation occurs when flow separation occurs in the gap between the blade tip and the housing or casing.[1] The likelihood of gap cavitation may be reduced by rounding the intersection of the blade surfaces with the tip [29]. Blade-end cavitation is due to boundary layer vorticity stretching and occurs when tip clearance is small. DPLL does not model boundary layer profiles on the blades or the duct; therefore, this type of cavitation cannot be taken into account here. Leakage vortex cavitation is due to low pressure within vortices formed by the interaction of through-flow and gap flow.

According to Farrell and Billet, this is the "most problematic and least understood" of the three types. Based on experimental results[2] and comparison with published data sets, they suggest a correlation between relevant design parameters and the minimum leakage vortex pressure. This correlation is adapted here to predict relative likelihood of cavitation among variants in the optimization process:

$$\sigma = \frac{\kappa}{8\pi^2 A^2} C_{L0}^2 \left[\frac{W_1(1 - e^{-14\epsilon}) + W_\infty k_s}{(1 - e^{-6\lambda})U} \right]^2 Re_c^{\frac{2}{7}} \tag{A.1}$$

The correlation parameters are defined in Table A.1, along with their adaptations in the DPLL model. The usual definition of cavitation index involves the total far-field pressure p_∞, the vapor pressure at the operating temperature p_v, and the far-field velocity (assumed here, without loss of generality, to be ship speed V_s):

$$\sigma = \frac{p_\infty - p_v}{\frac{1}{2}\rho V_s^2} \tag{A.2}$$

This is an environment-dependent positive quantity which decreases with increasing reference velocity or decreasing p_∞. A lower value of σ indicates greater likelihood of

[1] Axial flow pumps are very similar in principle to ducted propellers. In adapting the Farrell and Billet correlation to this research, the housing is considered to be the inner surface of a duct.

[2] These experiments were performed at the HIREP facility discussed in Section 3.1.2.

Parameter	Farrell and Billet	DPLL
κ	correction factor for levels of air content in water	1.0
A	empirical constant (0.36)	0.36
W_1	relative velocity at blade tip	relative velocity at control point of outer lifting segment
W_∞	edge velocity of end-wall boundary layer at blade tip	W_1
C_{L_0}	tip lift coefficient at zero clearance condition	axial force produced by outer lifting segment normalized by $\frac{1}{2}\rho W_1^2 A$
ϵ	ratio of tip gap to blade chord length at tip	1% of tip radius divided by blade chord length, which is constant for all variants at 2% of body length
k_s	empirical constant (0.18)	0.18
λ	ratio of tip gap to maximum blade thickness at tip	1.0
U	tip speed	propeller angular velocity $\times$ radius of outer lifting segment control point
Re_c	blade tip chord relative Reynolds number, $\frac{W_\infty c}{\nu}$	$W_\infty = W_1$, blade chord length constant at 2% of body length, viscosity derived from input vehicle Reynolds number $\frac{V_s L}{\nu}$

Table A.1: Adaptation of cavitation parameters to DPLL model

cavitation. The local *pressure coefficient* is defined somewhat similarly as the difference between local total pressure and reference pressure, normalized by a convenient dynamic pressure:

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho V_s^2} \tag{A.3}$$

Obviously, local cavitation is dependent on local pressures. If the local pressure at some point in the flow field is less than vapor pressure, cavitation will occur. On the other hand, if the *minimum* pressure of the entire flow field is greater than vapor pressure, cavitation is unlikely. If this is the case, and if the global cavitation index is then somehow steadily lowered, cavitation will first appear in the flow when $\sigma = -C_{p_{\min}}$. In other words, the quantity $-C_{p_{\min}}$ is the maximum cavitation index for which cavitation will occur. When comparing propulsor configurations operating in the same global conditions, therefore, the one having a higher value of $C_{p_{\min}}$ (that is, a lower value of $-C_{p_{\min}}$ is the more desirable in terms of cavitation, as it requires a lower global cavitation index (i.e., lower static pressure or greater speed) to perform as poorly. This is why the σ of Equation (A.1), as calculated in the DPLL model, is to be minimized if minimal cavitation is desired.

In this research, it is assumed that a direct correlation exists between the calcu-

lated cavitation index and propeller noise level. There are several reasons, however, why this must be treated only as a relative, first order approximation. First, some parameters in Equation (A.1) are not calculated by the DPLL model and their values must either be ignored or assumed in the adaptation, as shown in Table A.1. Also, as mentioned above, the leakage vortex cavitation estimated by this correlation is only one of at least three types known to occur in the tip region. No account is taken of these other cavitation mechanisms. Finally, cavitation is certainly possible in regions other than the tip and the correlation makes no prediction in this regard.

Appendix B

Reduction Gear and Shafting Volumes

Reduction gear volume is a function of the diameter and face width of the gears. The maximum tangential load W_t per unit length on the gear teeth can be written as [42]:

$$(\frac{W_t}{F_e})_{max} = \kappa J d_g \tag{B.1}$$

where κ is a safety factor, J is an experimentally determined material constant in units of force per unit length of face per unit length of diameter, d_g is the gear diameter, and F_e is the effective face width of the gear. The actual working load is a function of power and shaft speed:

$$\frac{W_t}{F_e} = \frac{P}{\omega d_g F_e} \tag{B.2}$$

where P is power delivered by the prime mover and ω is the angular velocity of the gear. According to [42], the most "economical" reduction gear uses the smallest pinion diameter possible in relation to its working face. However, the face width-to-diameter ratio must be small enough to avoid excessive deflections; a typical value is 2.25. In any regard, the face width is of the same order as the diameter, so by setting the two expressions equal one may write gear diameter in terms of torque (Q):

$$d_g \propto Q^{\frac{1}{3}} \tag{B.3}$$

and the projected area of the gear is then

$$A_g = k_g Q^{\frac{2}{3}} \tag{B.4}$$

where k_g is constant for a given gear material, width-to-diameter ratio and safety factor.

It is necessary to approximate k_g in this research so that valid volume penalties may be applied to variants with high torque coefficients. To avoid the inclusion of

classified information, a published study of general submarine internal arrangements is used as a baseline [16]. The mechanical drive variant described there produces 25,000 hp at 100 rpm, resulting in a torque of 1.79×10^6 N· m. The cross-sectional area of the reduction gear for this variant, assumed to be representative of those currently in use, is approximately 50 m^2, giving a k_g of 0.0034. This constant, along with the assumption of a length-to-diameter ratio, allows approximation of reduction gear volume given torque.

The volume required for the shaft is calculated similarly, although shaft length is assumed constant for all variants and is thus included in the constant of proportionality. This reduces the torque contribution by an order of magnitude:

$$A_s = k_s Q^{\frac{1}{3}} \tag{B.5}$$

Using arrangements drawings from [16] to estimate the shaft cross-sectional area and the powering parameters from above, k_s is found to be 0.075.

The volume coefficient C_V used in this research is defined as the volume enclosed by the hull profile minus the estimated gear and shaft volumes, normalized to a right circular cylinder of radius R_B and length 1.0.

Bibliography

[1] T. Arslan, D. H. Horrocks, and E. Ozdemir. Structural synthesis of cell-based VLSI circuits using a multi-objective genetic algorithm. *Electronics Letters*, 32(7):651–652, March 1996.

[2] D. S. Barrett. *Propulsive Efficiency of a Flexible Hull Underwater Vehicle.* PhD thesis, Department of Ocean Engineering, Massachusetts Institute of Technology, May 1996.

[3] A. D. Bethke. *Genetic Algorithms as Function Optimizers.* PhD thesis, University of Michigan, 1981.

[4] C. F. Beyer, C. P. Chi, and M. D. Devine. User manual for ASSET/MONOSC: Advanced Surface Ship Evaluation Tool/Monohull Surface Combatant Program. Technical Report BCS 40530-10, David Taylor Research Center, May 1990.

[5] S. D. Black. An integrated lifting surface/Navier-Stokes propulsor design method. Master's thesis, Department of Ocean Engineering, Massachusetts Institute of Technology, June 1994.

[6] S. D. Black. *Integrated Lifting-Surface/Navier-Stokes Design and Analysis Methods for Marine Propulsors.* PhD thesis, Department of Ocean Engineering, Massachusetts Institute of Technology, June 1997.

[7] L. Booker. Improving search in genetic algorithms. In L. Davis, editor, *Genetic Algorithms and Simulated Annealing*, Research Notes in Artificial Intelligence, pages 61–73. Pitman Publishing, London, 1987.

[8] J. P. Breslin and P. Andersen. *Hydrodynamics of Ship Propellers*, volume 3 of *Cambridge Ocean Technology Series.* Cambridge University Press, 1993.

[9] F. Buchoux. Improved algorithms for the computation of induced velocities in propeller design. Master's thesis, Department of Ocean Engineering, Massachusetts Institute of Technology, September 1995.

[10] R. Burcher and L. Rydill. *Concepts in Submarine Design*, volume 2 of *Cambridge Ocean Technology Series*. Cambridge University Press, 1994.

[11] R. A. Caruana and J. D. Schaffer. Representation and hidden bias: Gray versus binary coding for genetic algorithms. In *Proceedings of the Fifth International Conference on Machine Learning*, pages 153–162, 1988.

[12] A. J. Chipperfield and P. J. Fleming. Multiobjective gas turbine engine controller design using genetic algorithms. *IEEE Transactions on Industrial Electronics*, 43(5), October 1996.

[13] A. J. Chipperfield, P. J. Fleming, and C. M. Fonseca. Genetic algorithm tools for control systems engineering. In *Proceedings of the First International Conference on Adaptive Computing in Engineering Design and Control*, Plymouth Engineering Design Centre, UK, September 1994.

[14] F. H. Clauser. Turbulent boundary layers in adverse pressure gradients. *Journal of Aeronautical Science*, 21:91–108, 1954.

[15] W. B. Coney. MIT-PLL User's Manual. Technical report, Department of Ocean Engineering, Massachusetts Institute of Technology, November 1988.

[16] T. B. Dade. Advanced electric propulsion, power generation, and power distribution. *Naval Engineers Journal*, 106(2):83–92, March 1994.

[17] C. Darwin. *The Origin of Species*. Norton, New York NY, 1st edition, 1975.

[18] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York NY, 1991.

[19] K. A. De Jong. *Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, Department of Computer and Communication Sciences, University of Michigan, 1975.

[20] D. C. Dlesk and J. S. Liebman. Multiple objective engineering design. *Engineering Optimization*, 6(3), March 1983.

[21] M. Drela. 16.13 Lecture Notes - Aerodynamics of Viscous Fluids. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, September 1997.

[22] M. Drela. Source code for boundary layer marching routine MRCHBL as of August 1996. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

[23] G. Dyne. The principles of propulsion optimization. *Transactions of the Royal Institute of Naval Architects*, 137:189–208, 1995.

[24] F. Y. Edgeworth. *Mathematical Psychics.* P. Keagan, London, England, 1881.

[25] K. J. Farrell and M. L. Billet. A correlation of leakage vortex cavitation in axial-flow pumps. *Journal of Fluids Engineering*, 116(3):551–557, September 1994.

[26] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, University of Illinois at Urbana-Champaign, July 1993.

[27] C. M. Fonseca and P. J. Fleming. Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction. In *Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, Sheffield UK, September 1995.

[28] C. M. Fonseca and P. J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[29] W. S. Gearhart. Tip clearance cavitation in shrouded underwater propulsors. *AIAA Journal of Aircraft*, 3(2), 1966.

[30] T. C. Gillmer and B. Johnson. *Introduction to Naval Architecture.* Naval Institute Press, Annapolis, MD, 1982.

[31] D. E. Goldberg. Optimal initial population size for binary-coded genetic algorithms. Technical Report TCGA-850001, University of Alabama, November 1985.

[32] D. E. Goldberg. Simple genetic algorithms and the minimal deceptive problem. In L. Davis, editor, *Genetic Algorithms and Simulated Annealing*, Research Notes in Artificial Intelligence, pages 74–88. Pitman Publishing, London, 1987.

[33] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[34] D. E. Goldberg. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, 5(2):139–167, 1991.

[35] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, San Mateo, CA, 1991.

[36] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms, ɔise, and the sizing of populations. *Complex Systems*, 6(4):333–362, 1992.

[37] D. E. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3(5):493–530, 1989.

[38] S. Goldstein. On the vortex theory of screw propellers. In *Proceedings of the Royal Society of London*, volume 123A, London, 1929. The Royal Society.

[39] S. J. Gould. *Wonderful Life.* Norton, New York NY, 1989.

[40] J. J. Grefenstette. Optimization of control parameters for genetica algorithms. *IEEE Transactions on Systems, Man & Cybernetics*, 16(1), January-February 1986.

[41] P. J. B. Hancock. Selection methods for evolutionary algorithms. In L. Chambers, editor, *Practical Handbook of Genetic Algorithms*, volume 2, chapter 3. CRC Press, Boca Raton, FA, 1995.

[42] R. L. Harrington, editor. *Marine Engineering.* Society of Naval Architects and Marine Engineers, New York, NY, 1971.

[43] J. H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, 1975.

[44] R. B. Hollstein. *Artificial Genetic Adaptation in Computer Control Systems.* PhD thesis, Department of Computer and Communication Sciences, University of Michigan, 1971.

[45] J. Horn and N. Nafpliotis. Multiobjective optimization using the niched pareto genetic algorithm. Technical Report IlliGAL 93005, University of Illinois at Urbana-Champaign, July 1993.

[46] G. R. Hough and D. E. Ordway. The generalized actuator disk. Technical Report TAR-TR6401, THERM Advanced Research, January 1964.

[47] T. T. Huang, N. Santelli, and G. Belt. Stern boundary-layer flow on axisymmetric bodies. In *Twelfth Symposium on Naval Hydrodynamics*, pages 127–157, Washington, DC, 1979.

[48] T.T. Huang et al. Propeller/stern/boundary layer interaction on axisymmetric bodies: Theory and experiment. Technical Report DTNSRDC 76-0113, David W. Taylor Naval Ship Research and Development Center, December 1976.

[49] C. L. Hwang and A. S. M. Masud. *Multiple Objective Decision Making– Methods and Applications.* Springer-Verlag, New York, NY, 1979.

[50] H. A. Jackson. Submarine design trends. Lecture notes from MIT Professional Summer course, held at Draper Laboratory, Cambridge MA, June 1995.

[51] C. Z. Janikow and Z. Michalewicz. An experimental comparison of binary and floating point representations in genetic algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 31–36, University of California, San Diego, July 1991.

[52] A. Juels and M. Wattenberg. Stochastic hillclimbing as a baseline method for evaluating genetic algorithms. Technical Report CSD-94-834, University of California at Berkely, September 1994.

[53] C. L. Karr and L. M. Freeman. Genetic-algorithm-based fuzzy control of spacecraft autonomous rendezvous. *Engineering Applications of Artificial Intelligence*, 10(3):293–300, June 1997.

[54] J. E. Kerwin. 13.04 Lecture Notes - Hydrofoils and Propellers. Department of Ocean Engineering, Massachusetts Institute of Technology, January 1994.

[55] J. E. Kerwin. PLLH - Lifting-line circulation solution based on input tangential induced velocity distribution. Unpublished program and note, July 1995.

[56] J. E. Kerwin et al. A coupled viscous/potential flow design method for wake-adapted, multi-stage, ducted propulsors. In *Proceedings of the Society of Naval Architects and Marine Engineers*, 1994.

[57] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkely Symposium on Mathematical Statistics and Probability*, Berkely, California, 1951.

[58] W. P. A. Van Lammeren, L. Troost, and J. G. Koning. *Resistance, Propulsion, and Steering of Ships*, volume 2 of *Ships and Marine Engines.* H. Stam-Haarlem-Holland, 1948.

[59] H. W. Lerbs. Moderately loaded propellers with a finite number of blades and an arbitrary distribution of circulation. *SNAME Transactions*, vol. 60, 1952.

[60] J. Lis and A. E. Eiben. A multi-sexual genetic algorithm for multiobjective optimization. In *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, Indianapolis, IN, April 1997.

[61] K. R. MacCrimmon. An overview of multiple objective decision making. In J. L. Cochrane and M. Zeleny, editors, *Multiple Criteria Decision Making*, pages 18–44. University of South Carolina Press, Columbia, SC, 1973.

[62] K. F. Man, K. S. Tang, and S. Kwong. Genetic algorithms: Concepts and applications. *IEEE Transactions on Industrial Electronics*, 43(5):519–534, October 1996.

[63] G. P. McHugh. Advances in ducted propulsor analysis using vortex-lattice lifting-surface techniques. Naval Engineer thesis, Department of Ocean Engineering, Massachusetts Institute of Technology, September 1997.

[64] G. L. Mellor and D. M. Gibson. Equilibrium turbulent boundary layers. *Journal of Fluid Mechanics*, 24:225–253, 1966.

[65] K. Miettinen. *On the Methodology of Multiobjective Optimization with Applications.* PhD thesis, University of Jyväskylä, Finland, 1994.

[66] W. M. Milewski. *Three-Dimensional Viscous Flow Computations Using the Integral Boundary Layer Equations Simultaneously Coupled with a Low Order Panel Method.* PhD thesis, Department of Ocean Engincering, Massachusetts Institute of Technology, June 1997.

[67] J. N. Newman. *Marine Hydrodynamics.* The MIT Press, Cambridge, MA, 1977.

[68] C. K. Oei, D. E. Goldberg, and S.-J. Chang. Tournament selection, niching, and the preservation of diversity. Technical Report IlliGAL 91011, University of Illinois at Urbana-Champaign, December 1991.

[69] V. Pareto. *Manuale di Economia Politica.* Societa Editrice Libraria, Milano, Italy, 1906.

[70] M. A. Pawlowsky. Crossover operators. In L. Chambers, editor, *Practical Handbook of Genetic Algorithms*, volume 1, chapter 4. CRC Press, Boca Raton, FA, 1995.

[71] M. M. Pedram and H. Seifi. Extended algorithm of a fuzzy-set based power-system stabilizer with genetic algorithm tuning. *European Transactions on Electrical Power*, 7(3):205–210, May-June 1997.

[72] L. Prandtl. *Collected Works*, volume II. Springer, Berlin, 1961.

[73] M. Qiu. Prioritising and scheduling road projects by genetic algorithm. *Mathematics and Computers in Simulation*, 43:569–574, March 1997.

[74] M. A. Rosenman and J. S. Gero. Pareto optimal serial dynamic programming. *Engineering Optimization*, 6(4):177–183, 1983.

[75] R. H. Sabersky, A. J. Acosta, and E. G. Hauptmann. *Fluid Flow.* MacMillan, New York, NY, 3rd edition, 1989.

[76] J. D. Schaffer. *Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms.* PhD thesis, Vanderbilt University, December 1984.

[77] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the Ninth International Conference on Artificial Intelligence*, pages 93–100, Los Angeles CA, 1985.

[78] J. D. Schaffer. Some effects of selection procedures on hyperplane sampling by genetic algorithms. In L. Davis, editor, *Genetic Algorithms and Simulated Annealing*, Research Notes in Artificial Intelligence, pages 89–99. Pitman Publishing, London, 1987.

[79] J. D. Schaffer, R. A. Caruana, L. J. Eshelman, and R. Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, George Mason University, June 1989.

[80] J. R. Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, May 1995.

[81] K. J. Shaw and P. J. Fleming. Initial study of multi-objective genetic algorithms for scheduling the production of chilled ready meals. In *Proceedings of the Second International Conference on Genetic Algorithms*, Technical University of Brno, Czech Republic, 1996.

[82] T. E. Taylor. Source code for DPLL v1.0 as of August 1996. Department of Ocean Engineering, Massachusetts Institute of Technology.

[83] T. E. Taylor. *Preliminary Design and Analysis of Propulsors for Axisymmetric Underwater Vehicles.* PhD thesis, Department of Ocean Engineering, Massachusetts Institute of Technology, September 1996.

[84] M. W. Thomas. Evaluation and optimization of axial air gap propulsion motors for naval vessels. Naval Engineer thesis, Department of Ocean Engineering and Master's thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, June 1996.

[85] G. Tortora, B. Funke, and C. Case. *Microbiology: An Introduction.* Benjamin/Cummings, Menlo Park, CA, 2nd edition, 1986.

[86] J. D. van Manen. Effect of radial load distribution on the performance of shrouded propellers. In *Transactions of the Royal Institution of Naval Architects*, 1962.

[87] M.-H. Wang. Hub effects in propeller design and analysis. Technical Report 85-12, Department of Ocean Engineering, Massachusetts Institute of Technology, May 1985.

[88] C. L. Warren. Submarine design optimization using boundary layer control. Naval Engineer thesis, Department of Ocean Engineering and Master's thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, June 1997.

[89] D. S. Weile, E. Michielssen, and D. E. Goldberg. Multiobjective synthesis of electromagnetic devices using nondominated sorting genetic algorithms. In *Proceedings of the IEEE Antennas and Propagation International Symposium*, Baltimore, MD, 1996.

[90] F. M. White. *Viscous Fluid Flow.* McGraw-Hill, 2nd edition, 1991.

[91] D. Whitley. The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of the Third International Conference on Genetic Algorithms*, George Mason University, June 1989.

[92] D. Whitley and J. Kauth. GENITOR: A different genetic algorithm. In *Proceedings of the Rocky Mountain Conference on Artificial Intelligence*, Denver, CO, 1988.

[93] J. W. Wrench. The calculation of propeller induction factors. Technical Report 1116, David Taylor Model Basin, 1957.

[94] A. H. Wright. Genetic algorithms for real parameter optimization. In J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 205–218. Morgan Kaufmann, San Mateo, CA, 1991.

[95] W. C. Zierke, W. A. Straka, and P. D. Taylor. The high reynold's number flow through an axial-flow pump. Technical Report TR 93-12, Pennsylvania State University, November 1993.

Index