Using Single Propeller Performance Data to Predict Counter-Rotating Propeller Performance for a High Speed Autonomous Underwater Vehicle

Jessica A. Jacobson

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Aerospace and Ocean Engineering

Dr. Wayne Neu
Dr. Daniel Stilwell
Dr. Craig Woolsey

May 2, 2007
Blacksburg, Virginia

Keywords: Propeller, Counter-Rotating, AUV

Using Single Propeller Performance Data to Predict Counter-Rotating Propeller Performance for a High Speed Autonomous Underwater Vehicle

Jessica A. Jacobson

Abstract

The use of counter-rotating propellers is often desirable for aerospace and ocean engineering applications. Counter-rotating propellers offer higher peak efficiencies, better off-design performance, and roll control capabilities. But counter-rotating propeller matching is a difficult and complex procedure. Although much research has been done on the design of optimal counter-rotating propeller sets, there has been less focus on predicting the performance of unmatched counter-rotating sets. In this study, it was desired to use off-the-shelf marine propellers to make a counter-rotating pair for a high speed autonomous underwater vehicle (AUV). Counter-rotating propellers were needed to provide roll control for the AUV. Pre-existing counter-rotating propeller design methods were not applicable because they all require inputs of complex propeller blade geometries. These geometries are rarely known for off-the-shelf propellers.

This study proposes a new method for predicting the counter-rotating performance of unmatched propeller sets. It is suggested here that propeller performance curves can be used to predict counter-rotating thrust and torque performance.

Propeller performance tests were run in the Virginia Tech Water Tunnel for a variety of small, off-the shelf propellers. The collected data was used to generate the propeller performance curves. The propellers were then paired up and tested as counter-rotating sets. A momentum theory based model was formulated that predicted counter-rotating performance using the propeller performance data. The counter-rotating data was used to determine the effectiveness of the method.

A solution was found that successfully predicted the counter-rotating performance of all of the tested propeller sets using six interaction coefficients. The optimal values of these coefficients were used to write two counter-rotating performance prediction programs. The first program takes the forward and aft RPMs and the flow speed as inputs, and predicts the generated thrust and torque. The second program takes the flow speed and the desired thrust as inputs and

calculates the forward and aft RPM values that will generate the desired thrust while producing zero torque. The second program was used to determine the optimal counter-rotating set for the HSAUV.

Acknowledgments

I would first like to thank my parents for their unwavering support throughout my entire education, and especially for their support over the past two years. I am not embarrassed to admit that this project sometimes proved to be almost too stressful to handle, but my parents were always just a phone call away, encouraging me to continue and listening to my concerns as only parents can. Thank you Mom and Dad, I do not know how I would have done this without you.

I would also like to thank my fiancé, Jesse. There are no words to convey how grateful I am for his love and support. Even while working on his own thesis, he was always there for me - motivating and encouraging me when I felt as though I would never find a solution, and celebrating small victories with me even when I wouldn't stop to explain why we were celebrating. He is my best friend and I hope that one day I can repay the kindness he has shown me.

I would also like to thank my advisor, Dr. Neu. He patiently corrected me every time I spoke before I thought and sat through the, what I can only imagine as unbearable, meetings in which I had drank too much caffeine. He has taught me so much about propellers and about the importance of thinking 'physically' and not just 'mathematically'. Without his attention to detail and knowledge of everthing from Fortran to momentum theory, this project would not be half of what it is today.

Thanks are also definitely deserved by my other committee members, Dr. Stilwell and Dr. Woolsey. In their own way, each has been supportive and helpful throughout this project. I am so grateful for both their advice and their time.

I would also like to thank John Hennage, Ryan Williams, Clint Jones and Jan Petrich for all of their help throughout this project. Without them, this project literally would not have been possible. John and Ryan deserve special thanks for designing and building both the single propeller and counter-rotating rigs and for designing and setting up the single and counter-rotating tests. Clint also deserves special thanks for always being willing to help me with the

electronics. John, Ryan, Clint and Jan have all been ‘go-to guys’ for me whenever I had a problem (and I had plenty) and they were always friendly and helpful. Again, without them, this project literally would not have been possible.

I would also like to thank all of my friends here in Blacksburg as well as those who have already moved away in pursuit of other goals. Laura, Ashley, Adam, Ryan, all of you have helped me in one way or another throughout the six years I have spent at Virginia Tech and I am truly grateful for your friendship.

Lastly, I would like to dedicate this thesis to the thirty-two innocent people who lost their lives during the Virginia Tech shootings on 4/16/07. No words can express the sadness I feel over the tragedy that occurred here on that day. I can only say that my thoughts and prayers have gone out to the family and friends of all of the victims – your loved ones will live in my memory forever.

Table of Contents

Nomenclature

a	percent velocity increase over the propeller
avf	axial velocity factor
A	propeller area, in^2
C_T	thrust loading coefficient
D	propeller diameter, in
$F(x)$	optimization error
J	advance ratio
K_T	open water thrust coefficient
K_Q	open water torque coefficient
η	open water efficiency
η_I	ideal efficiency
n	revolutions per second
NP	number of points
ρ	water density, $slug/ft^3$
P/D	pitch to diameter ratio
Q	torque, in-lbs
Qf	torque factor
Q_{off}	torque error, in-lbs
$rpmf$	RPM factor
R	propeller radius, in
RPM	revolutions per minute
T	thrust, lbs
Tf	thrust factor
T_{off}	thrust error, lbs
V	flow velocity, ft/s
V_A	speed of advance
ω_{swirl}	angular swirl velocity, rad/s
w	wake fraction
x, y, z	water tunnel coordinate system

Subscripts

1	forward propeller
2	aft propeller

List of Figures

List of Tables

Chapter 1 - Introduction

Counter-rotating propellers have long been a topic of interest in the aerospace and ocean engineering fields. The use of counter-rotating propellers offers many benefits when compared to using only one propeller. Counter-rotating propellers offer higher peak efficiencies and better off-design performance. This is mainly because swirl velocity losses of the forward propeller are recovered by the aft propeller as additional angular velocity. The use of counter-rotating propellers can also provide roll control for torpedoes or autonomous underwater vehicles (AUVs). With all of the benefits of counter-rotating propellers, there have been no large scale naval applications to date. This is mainly because the benefits offered by counter-rotating propellers are overshadowed by shafting and gearing complexities. The main naval application of counter-rotating propellers has been for roll controlled torpedoes.

1.1 Previous Research

There have been many studies done on the design of counter-rotating propeller sets. In 1943, Ginzel [1] created a theoretical model for the design of counter-rotating propellers. Ginzel focused his study on two propellers with the same diameter, number of blades, and blade shape, but with different distributions of the blade angle over the radius. The propellers were modeled with very little axial spacing and operating at opposite values of the same RPM.

Playle, Korkan and von Lavante [2] developed a numerical counter-rotating design method that calculated the optimal propeller geometries by determining the planform and twist distribution for each propeller. Their method required specific inputs of engine shaft horsepower, diameter, and disc spacing.

The most notable design method was developed in 1988 by Cox and Reed [3]. Their design improved on Ginzel's method by including a more accurate solution for the optimal circulation distribution. Their method was also the first to include a prediction of the slipstream contraction and boasted a more accurate method for computing the mutual interactions between the forward and aft propellers.

In 1987 design methods were so well developed that studies on the off-design performance of counter-rotating propellers were being undertaken. Korkan and Gazzaniga [4] did a study to determine whether counter-rotating propellers maintain superior off-design performance for constant speed and variable pitch cases.

Design methods had been so thoroughly developed that a code for counter-rotating propeller design was published by Reed [5] in 1985. This program designed the optimal aft propeller given the forward propeller of the counter-rotating set.

Although there has been much work done on the design of optimal counter-rotating sets, there has been far less research done on predicting the performance of two unmatched propellers. Hecker and McDonald [6] touched on the subject of the performance of an unmatched counter-rotating set in their 1960 study on the effect of axial spacing and forward propeller diameter.

Being able to predict the counter-rotating performance of any two propellers is an important research topic. Design methods often call for the use of complex, unknown blade geometries. When these geometries are not available, pre-existing design methods are virtually useless.

1.2 Motivation

This study was undertaken as a side project to Virginia Tech's HSAUV (High Speed Autonomous Underwater Vehicle) project. The HSAUV project objective was to design a high speed AUV for the U.S. Navy. The AUV had to be 3 inches in diameter and 3 ft long. It had to be able to travel at a minimum of 10 knots and have hover capabilities. The AUV also had to have roll control. The roll control requirement is what made the use of counter-rotating propellers desirable.

The topic of propeller matching for the AUV became of interest when it was realized that there are no current methods for matching off-the-shelf propellers. The HSAUV project did not have the luxury of being able to custom-design propellers, so some method of performance evaluation for the counter-rotating sets was needed.

1.3 Thesis Outline

This paper will first outline the modifications that were made to the Virginia Tech Water Tunnel in preparation for propeller testing. The single propeller tests and propeller data reduction results will then be discussed. Counter-rotating propeller tests and the subsequent data reduction results will follow the single propeller discussion. The calculations that are the basis of the performance prediction models will be outlined followed by a summary of the different model versions and results. A final prediction method and optimal counter-rotating set will be presented as a solution to the HSAUV counter-rotating matching problem.

Chapter 2 - Virginia Tech Water Tunnel

The HSAUV project needed a test facility that could accommodate a 3 inch diameter model, and provide a 10 knot flow. The facility was needed for preliminary drag tests, single propeller tests and counter-rotating propeller tests.

The Virginia Tech Water Tunnel was originally intended as a flow facility for calibrating and testing oceanographic water current meters. The original flow range was 0.01 to 5 knots. Three pump and motor sets were used to achieve this range. The “A pump” produced a flow range of 0.01 to 0.2 knots, the “B pump”, a range of 0.2 to 1 knots, and the “C pump”, 1 to 4 knots. The pumps could be operated with their valves partially open, which made it possible to achieve any flow speed within a pump’s respective range. Using the pumps in combination with one another made it possible to achieve any flow speed between 0.01 and 5 knots.

The tunnel is comprised of three chambers: the settling chamber, the test chamber and the receiving chamber. Water flows into the settling chamber through three vertical perforated standpipes. It moves through two large screens and is pushed through the nozzle and into the test chamber. The screens serve to reduce vorticity within the flow as well as filter the water. Once in the test chamber, the water moves into the receiving chamber through an opening in the chamber wall. Pumps pull the water out of the receiving chamber and through another set of three vertical perforated standpipes. The standpipes deliver the water to the A, B and C pumps, which then send the water to the settling chamber to restart the cycle. The tunnel dimensions are shown in Figure 2-1 **[7**]. Modifications had been made to the tunnel before it was obtained by Virginia Tech.

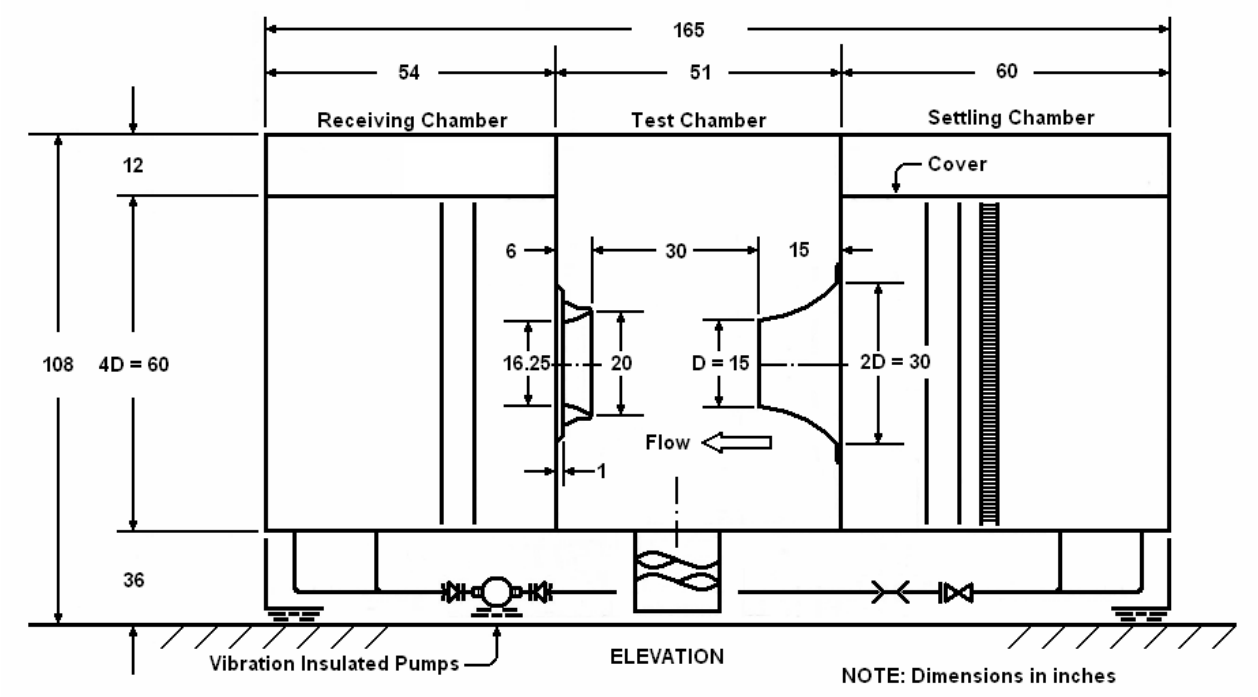

Figure 2-1. Diagram of the original calibration flow tank.

2.1 Modifications

Since the HSAUV project required 10 knot flow, the Virginia Tech Water Tunnel needed modifications before it could be used. The main modification was decreasing the diameter of the nozzle, but this made many other modifications necessary.

Nozzle Insert

One of the major modifications to the tunnel was decreasing the diameter of the nozzle. Ideal conservation of mass calculations were done to determine the nozzle diameter necessary to produce 10 knot flow:

$$U_{old} A_{old} = U_{new} A_{new} \quad (2\text{-}1)$$

$$5knots \times (15in)^2 = 10knots \times (D_{new})^2 \quad (2\text{-}2)$$

$$D_{new} = 10.6in \quad (2\text{-}3)$$

Since the calculations did not take frictional losses into account, a new nozzle diameter of 10 inches was chosen to ensure that the 10 knot flow was reached. Instead of constructing a completely new nozzle, it was decided that a nozzle insert would be made that could fit inside of the existing nozzle. The nozzle was removed from the tunnel and diameters at different stations were measured. The insert was built from PVC and was secured to the inside of the nozzle using silicone. The profiles of the nozzle and insert are shown in Figure 2-2. The insert is 10 inches in length and has a contraction ratio of 9 (as compared to the original nozzle's length of 15 inches and contraction ratio of 4).

The nozzle insert runs linearly along the inside surface of nozzle for its first 1.5 inches (z = 0 to 1.5 in.) and then becomes slightly concave (z = 1.5 to 2.5 in.). It then smoothly necks down (z = 2.5 to 7 in.) to a parallel region (z = 7 to 10 in.). This final linear region is intended to help straighten out the flow before it is pushed into the test chamber.

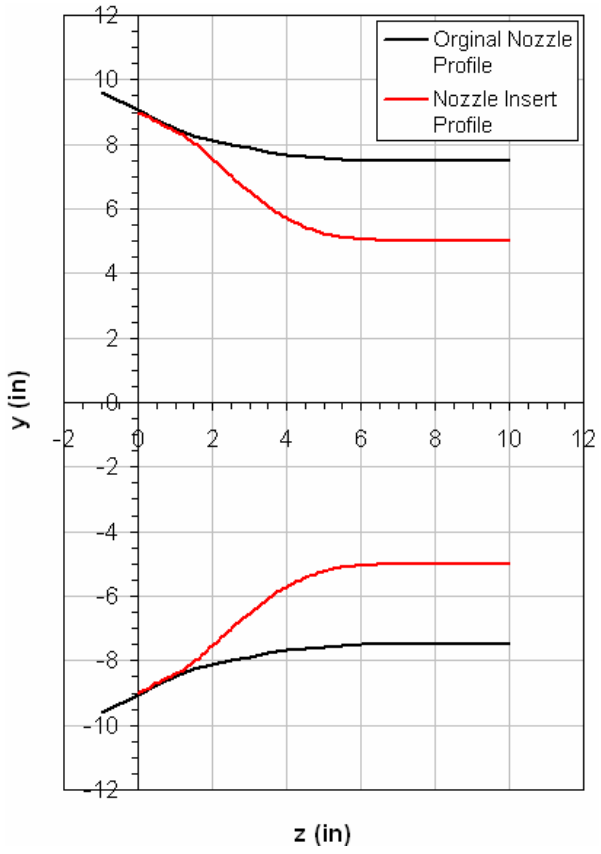

Figure 2-2. The nozzle insert.

Foam Catch

When the tunnel was run with the newly installed nozzle insert, it was found that the increased flow speed caused water to violently ricochet off of the wall between the test chamber and receiving chamber. Instead of flowing smoothly through the wall opening as was seen with the slower flows, the water entrained by the outside of the jet would hit the wall outside of the opening and circle backwards in the test chamber. It would then become reincorporated into the flow being propelled out of the nozzle. This circulation created an unwanted excess of turbulence in the test chamber.

A detachable foam catch that was part of the original tunnel but had been removed, proved to be the solution to this problem. The catch was bolted to the inside of the test section around the opening between the test and receiving chambers. The catch extended conically outward towards the nozzle and was 6 inches long. The sides of the catch were constructed with a wire netting and a mesh screen that slowed the circulating flow and funneled the water into the receiving chamber. The original catch extended too far into the test section so it was rebuilt to be only 3 inches long. The catch greatly reduced the circulation and turbulence seen in the test chamber. Although no velocity or turbulence measurements were made, the improvements to the flow quality could be seen visually. The catch is shown in Figure 2-3. In the figure, one of the three vertical standpipes in the receiving chamber can be seen through the catch.

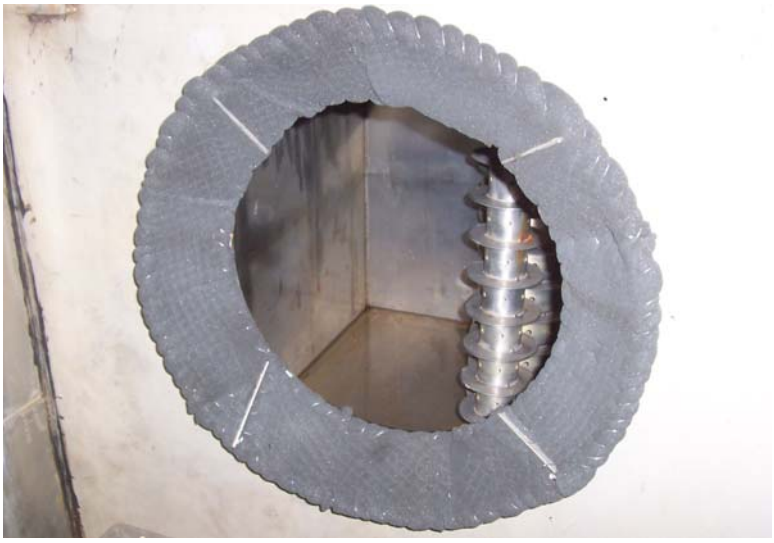

Figure 2-3. The foam catch.

Pressurized Settling Chamber Cover

Increasing the flow speed gave way to other problems with the water tunnel. When the nozzle insert was first installed, there were problems with the pressure differential between the chambers when the tunnel was running. Even below 10 knots, the pressure differential caused the water level in the settling chamber to become so high that it threatened to overflow. This made the water levels in the other two chambers too low. In the test chamber, the water level would become so low that the top of the nozzle would become exposed, and it was possible to actually see the jet coming out of the nozzle. In the receiving chamber, the low water level would expose the holes in the standpipes, causing them to suck in air instead of water.

In order to solve this problem a permanent cover was constructed to fit over the settling chamber. The cover is shown in Figure 2-4. A wooden lid was built and bolted to a metal lip that circled the top of the settling chamber. Wood was chosen for cost and manufacturing time savings. This lid was outfit with a pressure relief valve and pressure regulator. The pressure regulator was hooked up to the shop air line and allowed the settling chamber to be pressurized between 0 psi and 5 psi. Depending on the flow speed, the chamber was generally pressurized between 0.75 psi and 2.5 psi. When the pressure in the settling chamber was above 0.5 psi, the pressure relief valve would release 0.5 psi. This prevented the chamber from becoming over pressurized and became an unintentional indicator of whether or not the chamber was not pressurized enough. When the settling chamber was not pressurized enough, water would be pushed up into the pressure relief valve and would squirt out onto the settling chamber cover, indicating that the pressure regulator needed adjusting. This setup allowed the user to pressurize the settling chamber, thereby controlling the water level in all three chambers.

Receiving Chamber Modifications

Once the pressurized settling chamber cover was operational, it was discovered that there was still too much air entrained in the flow. The main cause of this was due to the setup of the receiving chamber. The vertical perforated standpipes sucked in air whenever the water level was not high enough to cover the top inlets. If the water level was above the top of the standpipes, the surface was extremely turbulent, allowing for more water and air mixing.

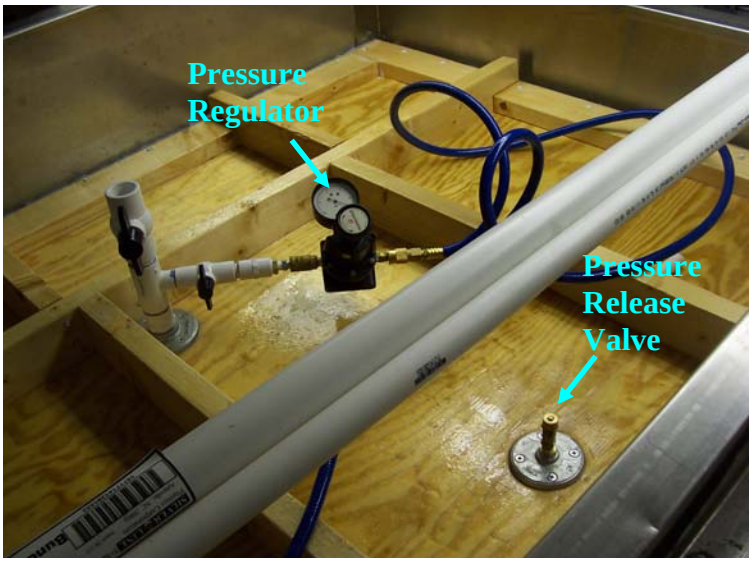

Figure 2-4. The settling chamber cover.

In order to prevent the standpipes from taking in air instead of water, rubber stoppers were inserted into the top three rows of holes. Since the water level never fell below this level, this ensured that suction from the pumps only pulled water into the pipes. In order to prevent air mixing into the water on the surface, a floating cover was installed. The cover was simply a piece of plywood that would rest on the standpipes when the water level was lower and would float on the water surface when the water level exceeded the standpipe height. These two modifications greatly reduced the amount of air in the flow.

2.2 Baseline Velocity Tests

After the modifications were made to the tunnel, baseline velocity tests were performed in order to ensure that the flow quality in the test section was still good.

In order to run the baseline tests, it was necessary to have a reliable flow speed measurement device. Cost and time constraints led to the decision that a pitot static tube would be used. A differential pressure transducer that was rated from 0 to 5 psi and was manufactured by Honeywell was used in conjunction with the pitot tube. Using a pressure transducer to measure mean velocities in water is often difficult. This is because the pressure transducer cannot be submerged with the pitot tube, therefore requiring that long tubes connect the pitot

tube to the static and dynamic ports on the pressure transducer. Managing the air / water interface in these long tubes is difficult and if the interfaces in the two tubes are not at the same height, the pressure transducer will not give accurate results. In order to solve this problem, the entire transducer was sealed inside of a potting compound (with the static and dynamic ports exposed). This allowed the pressure transducer to be submerged along with the pitot tube. The two exposed ports could then be attached to the pitot tube. This method of potting the pressure transducer made things much simpler by eliminating the need to run tubing all the way up the pitot tube strut, out of the tank, and to the pressure transducer and then constantly observing the air / water interfaces in the two tubes.

A traverse was constructed that allowed the pitot tube to move in the *x*, *y* and *z* directions. The traverse is shown in Figure 2-6. The tunnel axes are outlined in Figure 2-5. The origin was taken to be located at the center of the nozzle outlet.

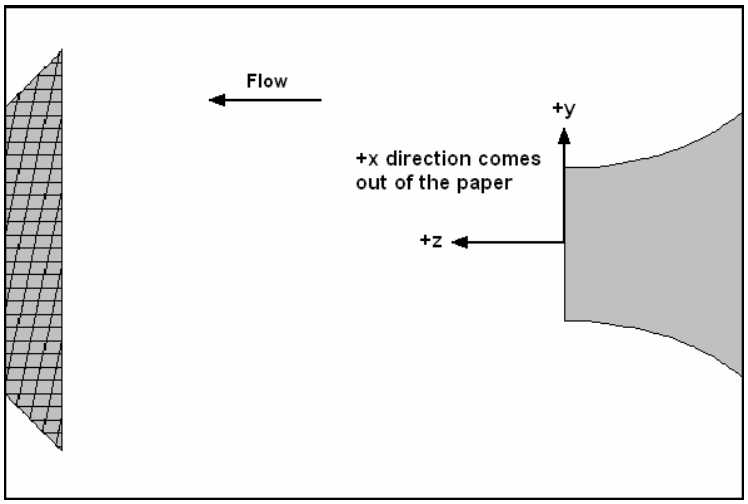

Figure 2-5. The tunnel axes.

Mean velocity measurements were taken at 3 stations: 4 inches past the nozzle (z = 4 inches), 16 inches past the nozzle (z = 16 inches), and 27 inches past the nozzle (z = 27 inches). A 9x9 grid of equally spaced points was taken at each station. The points were spaced 1.5 inches apart, so that the entire jet was encompassed as well as some area outside of the nozzle exit dimensions. The test grid is shown in Figure 2-7.

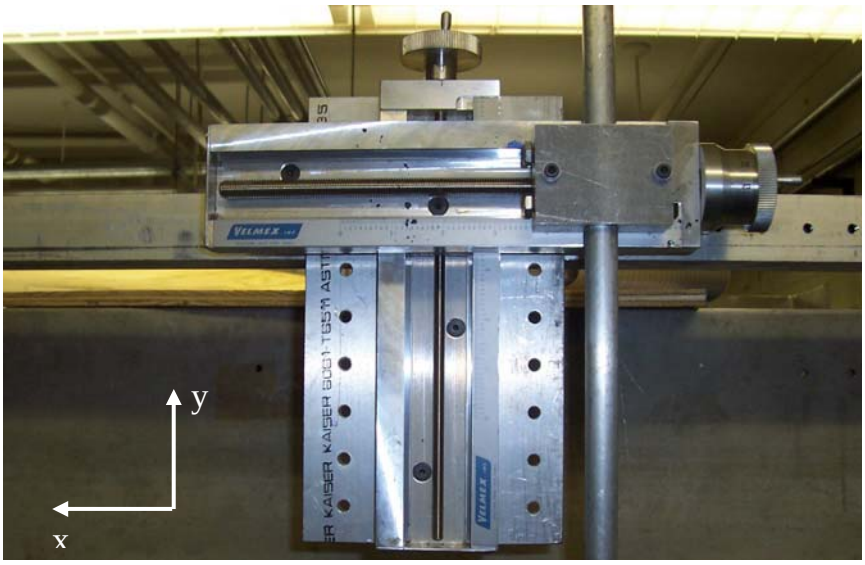

Figure 2-6. The traverse.

The test was run at three different speeds. The original ‘2 knot’ setting on the control panel, the original ‘4 knot’ setting on the control panel, and maximum speed (all three pumps on at maximum capacity). The test was intended not only to give the velocity contour plots, but also to see the speeds that now corresponded with the control panel labels.

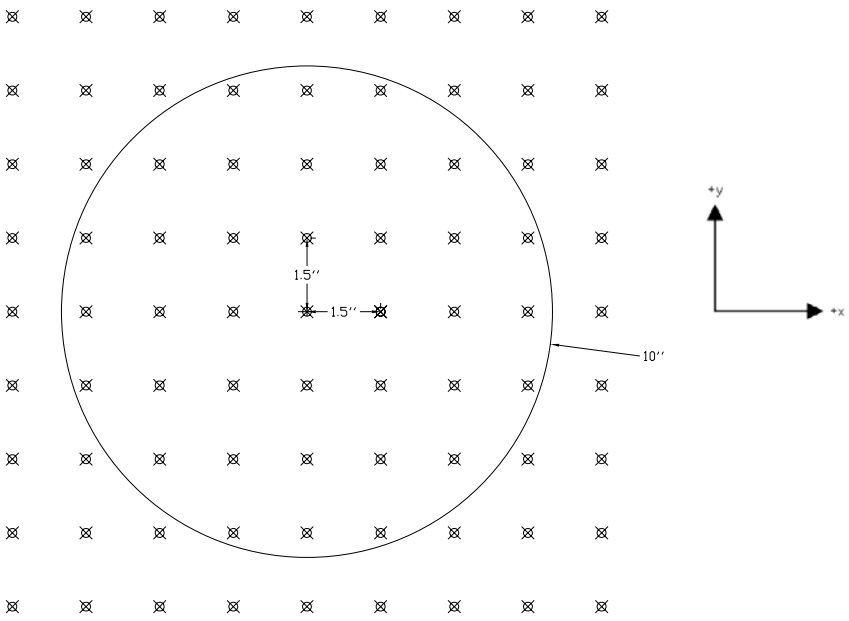

Figure 2-7. The test point grid for the baseline velocity tests.

Results

The results for the maximum speed run are shown in Figure 2-8, Figure 2-9, and Figure 2-10. Figure 2-8 shows the results at z = 4 inches, Figure 2-9 shows the results at z = 16 inches, and Figure 2-10 shows the results at z = 27 inches. In the figures, the black circle in the horizontal plane represents the nozzle exit. The flow is represented by the surface rising above the horizontal plane. The full results are shown in Appendix A.

The overall results of the analysis showed that the flow was very uniform as it exited the nozzle and did not deteriorate drastically, even at z = 27 inches. The flow appeared uniform over the center 5 inches of the jet at all three stations, making the testing of the 3 inch diameter AUV possible. The tests also revealed that the maximum tunnel speed was actually 10.4 knots, meeting the 10 knot requirement.

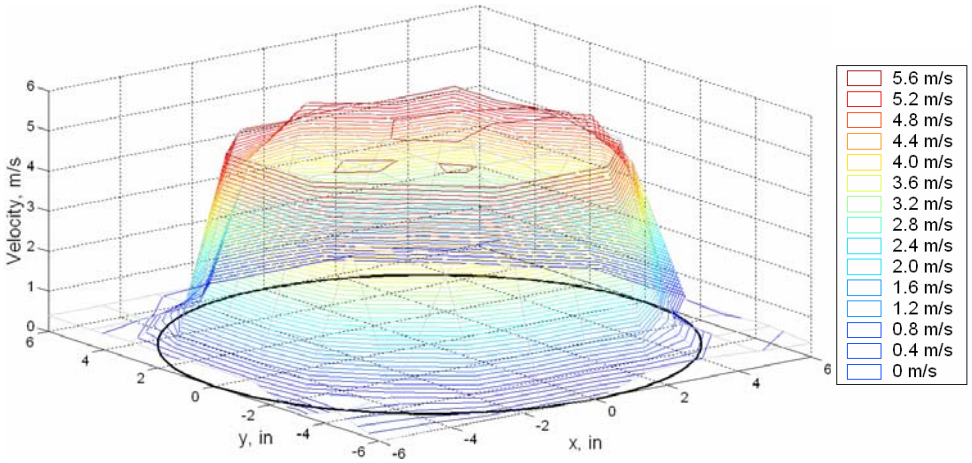

Figure 2-8. Baseline velocity results for maximum speed at z = 4 inches.

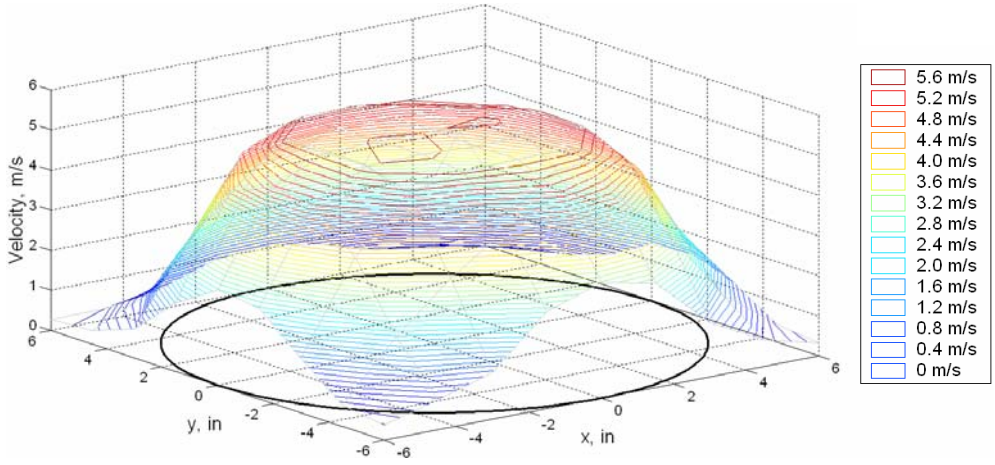

Figure 2-9. Baseline velocity results for maximum speed at z = 16 inches.

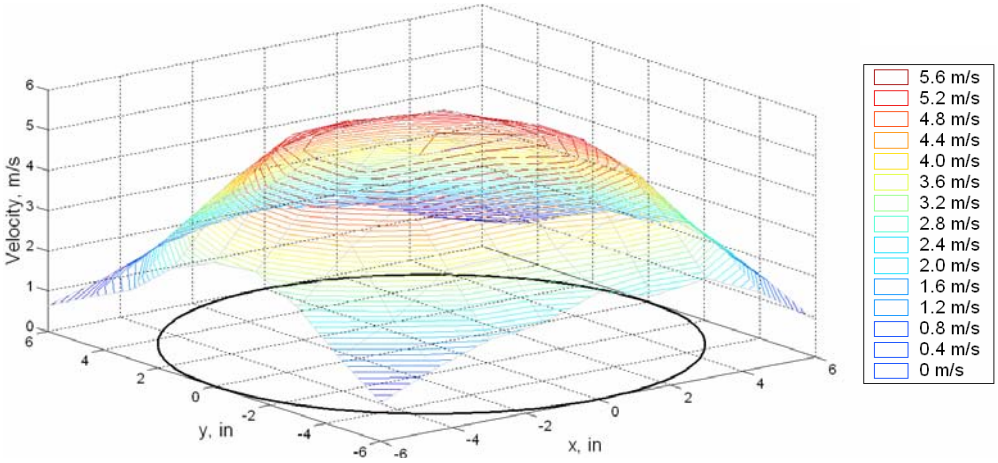

Figure 2-10. Baseline velocity results for maximum speed at z = 27 inches.

Chapter 3 - Single Propeller Tests

Once the water tunnel had been renovated, single propeller tests were run. The purpose of these tests was to generate the propeller performance curves for each propeller.

3.1 Propellers

Nine different Octura propellers were tested. Octura is a brand that manufactures small propellers for hobby boats. Octura propellers were chosen for this study because they were in the right diameter range and were off the shelf. Because the AUV is 3 inches in diameter, the diameter of each tested propeller had to be less than 3 inches. Two other stipulations were made when choosing propellers to test. The first was that possible aft propellers were preferred to have smaller diameters than the possible forward propellers. This was to prevent the aft propeller from operating outside of the front propeller's streamtube. The second stipulation was that possible aft propellers had to have the same or larger values of the *P/D* (pitch to diameter ratio) than the possible forward propellers. Since the front propeller was already increasing the flow velocity, the aft propeller needed a larger *P/D* in order to operate at a comparable RPM. Two of the lower pitch propellers were tested as candidates for the forward propeller of the counter-rotating set. The other seven propellers were tested as possible aft propellers. Figure 3-1 shows the tested propellers while Table 3-1 gives their characteristics. In the table, Z represents the number of blades.

Figure 3-1. The tested propellers

Table 3-1. Tested propellers.

Propeller	Diameter (in)	P/D	Z	Material	Series
1457	2.24	1.4	2	Plastic	14
1462	2.44	1.4	2	Plastic	14
2.0	2	1.8	2	Plastic	2.0
2352	2.05	2.0	2	Copper	23
2.1/3	2.1	1.8	3	Copper	2.0
1455	2.17	1.4	2	Plastic	14
1755	2.17	1.7	2	Copper	17
2055	2.17	2.0	2	Plastic	20
Z55	2.17	1.6	2	Plastic	Z

In the counter-rotating set, the forward propeller is actually a right handed propeller. All of the propellers that were tested were left handed propellers, but it was assumed that a left handed propeller would generate the same performance curves as its right handed counterpart. Not all propellers were available as both right and left handed propellers.

Each propeller's name indicates the manufacturing series to which it belongs. Different series have different characteristics. The series characteristics are given in Table 3-2.

Table 3-2. Octura propeller series characteristics.

Series	Characteristics
2.0	Highest levels of lift in the Octura range. No rake on the trailing edge.
14	Medium levels of lift. Blade area weighted toward the tip of the blade.
17	Highest levels of lift in the Octura range. No rake on the trailing edge.
20	High levels of lift. Originally designed for geared electrics.
23	Very high pitch to diameter ratios
Z	True pitch propellers (pitch does not change over the blade)

Since counter-rotating propeller matching is a complex problem with very few knowns, propellers from different series were chosen to try to determine which characteristics are important in propeller matching.

Some of the tested propellers were plastic and some were beryllium copper. The copper propellers were balanced before they were tested as recommended by the manufacturer. Most propellers were available in plastic, beryllium copper and aluminum. Plastic propellers were chosen when available because of cost and simplicity. When plastic was not available, copper was chosen simply because most of the propellers were not available in aluminum.

3.2 Test Setup

A rig was built to be used in the water tunnel specifically for this test. A picture of the rig is shown in Figure 3-2. The inside of the electronics section was kept completely dry. A seal in the nose of the rig prevented water from leaking in at the shaft. The balance section housed the six degree of freedom balance that was used to measure the forces and moments for these tests. This section was open to the flow and was therefore flooded during all of the tests.

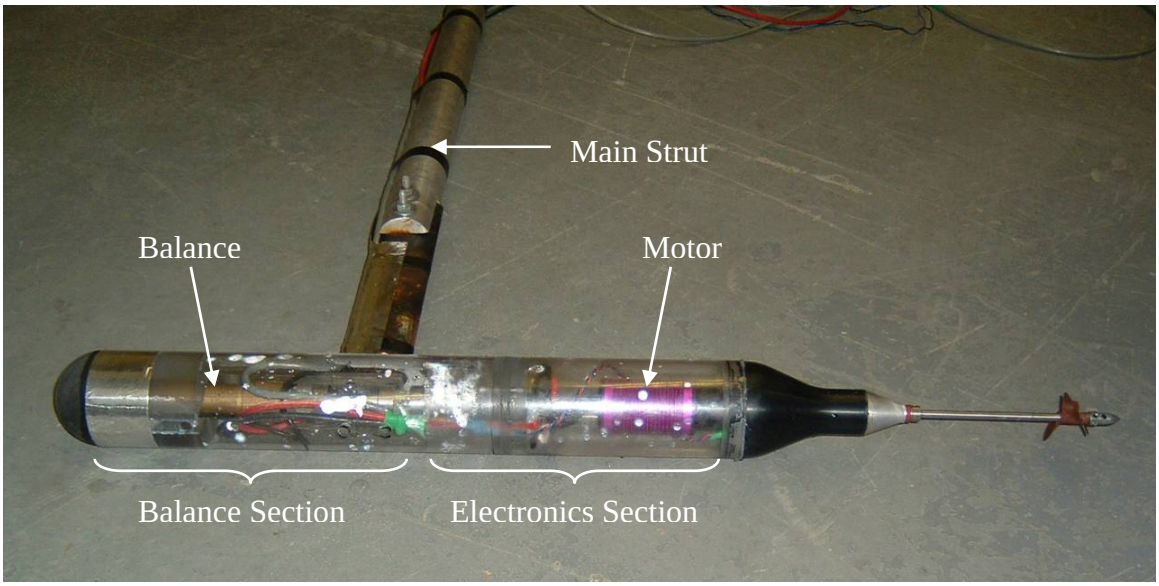

Figure 3-2. A picture of the single propeller test rig.

A diagram of the rig and its components is shown in Figure 3-3. The shroud and supporting struts are not shown in Figure 3-2. In Figure 3-3, the top diagram shows how the shroud fits over the rig. The bottom diagram shows the rig as if it had been cut through the center.

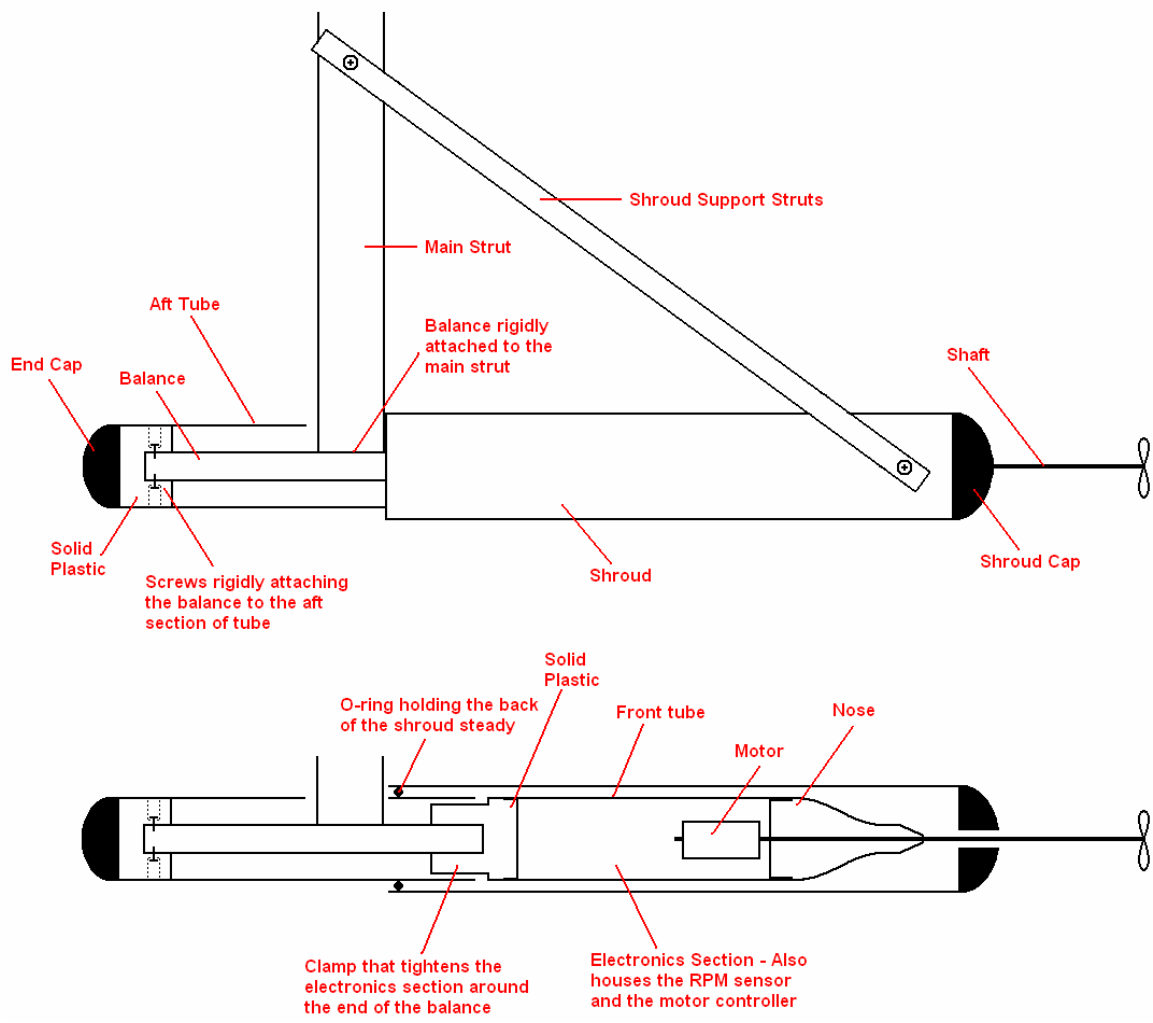

Figure 3-3. A diagram of the single propeller test rig.

Hardware

The test rig was designed to isolate the forces on the propeller. The balance was attached to both the main strut and aft tube by screws. By rigidly attaching the balance to both the strut and the aft tube, it was ensured that any drag forces on the strut or tube would not be measured by the balance. The electronics section was clamped to the forward end of the balance, so the

balance measured any forces or moments felt by the electronics section. The motor was used to rigidly attach the propeller shaft (and therefore the propeller) to the electronics section. The electronics section, therefore, transmitted the forces and moments generated by the propeller back to the balance. The shroud was intended to divert the flow and prevent the electronics section from transmitting any additional drag forces to the balance.

Wires from the electronics section were sent through the balance section and then taped to the back of the strut. It was crucial to ensure that the wires did not touch the balance as even the slightest disturbance would have affected the results. Once the balance and electronics sections were in place, the shroud slid over the shaft and over an o-ring on the balance section. This o-ring secured the back of the shroud and kept it from bouncing around. Two smaller support struts were also used to hold the shroud in place. The struts were attached to opposite sides of the main strut and then attached to the sides of the shroud. The purpose of these struts was to secure the front of the shroud. Ideally, the shaft would never touch the sides of the opening at the front of the shroud.

The main strut was hung from a crossbar that sat above the tank. The model was positioned so that it was directly in the center of the flow field.

The pitot tube and pressure transducer were used to measure the mean velocity for these tests. The pitot tube was placed ahead and to the side of the propeller. This was to ensure that the pitot tube would measure the flow outside of the stream-tube defined by the propeller disc. The baseline velocity results showed that center 8 inches of the jet was uniform for all speeds at $z = 4$ inches. Therefore, the pitot tube could be placed in the side of the jet as long as it remained within the center 8 inches.

Electronics

A six degree of freedom balance was used to measure the forces and moments seen by the propeller. Each balance signal was sent through a 2310 amplifier and then recorded using LabView software. A Hyperion brushless motor that was rated up to 40 amps was used to drive the propeller. The motor was driven by a Hyperion motor controller that was rated up to 30 amps. A Hall Effect Sensor (RPM sensor) that was manufactured by Allegro Microsystems was mounted next to the motor and measured the shaft RPM. The RPM signal was also recorded in

the LabView program. The propeller's speed could be increased or decreased by stepping the motor up or down. The motor was driven by a standard 12 volt automotive battery.

Since the electronics section was not ventilated in any way, there were initial problems with the electronics overheating. The brushless motors that were used to drive the propeller are off the shelf hobby motors. These motors are not generally used in sealed off spaces with no ventilation. The heat generated by the motor was not able to dissipate and the high temperatures would wreak havoc on the performance of the electronics. The main problem was that the motor controller would overheat and suddenly shut off. This problem was fixed by installing a water cooling system on the motor controller. Flexible, small diameter, plastic tubing was wrapped around the motor controller. A fish tank filter pump was used to pump water through the tubing, effectively cooling the motor controller. This greatly helped the overheating problems, although it did not solve them. Overheating was consistently a problem for both the single propeller and counter-rotating propeller tests.

3.3 Testing

All of the propellers were tested at tunnel speeds of 9 ft/s and 14 ft/s. When time permitted, propellers were also tested at 6 ft/s. For each test, the tunnel velocity was held constant while the propeller RPM was increased. The RPM was increased in steps and at each step the flow velocity (as seen by the pitot tube), three forces and three moments (as seen by the balance), the RPM (as seen by the RPM sensor), and the voltage and amperage (manual input) were recorded in LabView. A voltmeter and current meter were used to display the voltage of the battery and the current draw to the rig.

Results

The results were used to generate the propeller performance curves for each propeller. Propeller performance curves consist of a plot of the thrust coefficient (K_T), ten times the torque coefficient ($10K_Q$), and efficiency (η) all as a function of the advance ratio (J). These coefficients are also functions of n and D, the propeller RPM and diameter. The coefficients are defined in Equations (3-1) through (3-4).

$$K_T = \frac{T}{\rho n^2 D^4} \quad \textbf{(3-1)}$$

$$K_Q = \frac{Q}{\rho n^2 D^5} \quad \textbf{(3-2)}$$

$$\eta = \frac{K_T}{K_Q}\frac{J}{2\pi} \quad \textbf{(3-3)}$$

$$J = \frac{V_A}{nD} \quad \textbf{(3-4)}$$

The performance curves for propellers 2.0, Z55, and 1462 are shown in Figure 3-4, Figure 3-5 and Figure 3-6, respectively. The curves for all tested propellers are displayed in Appendix B.

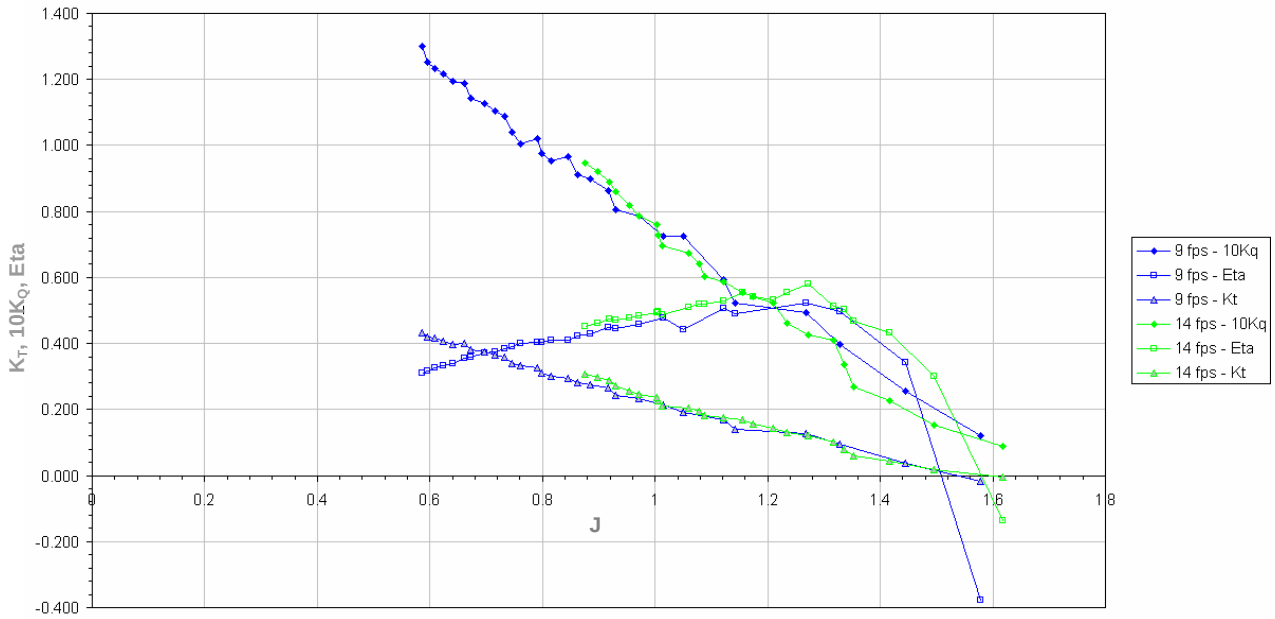

Figure 3-4. The propeller performance curves for propeller 2.0.

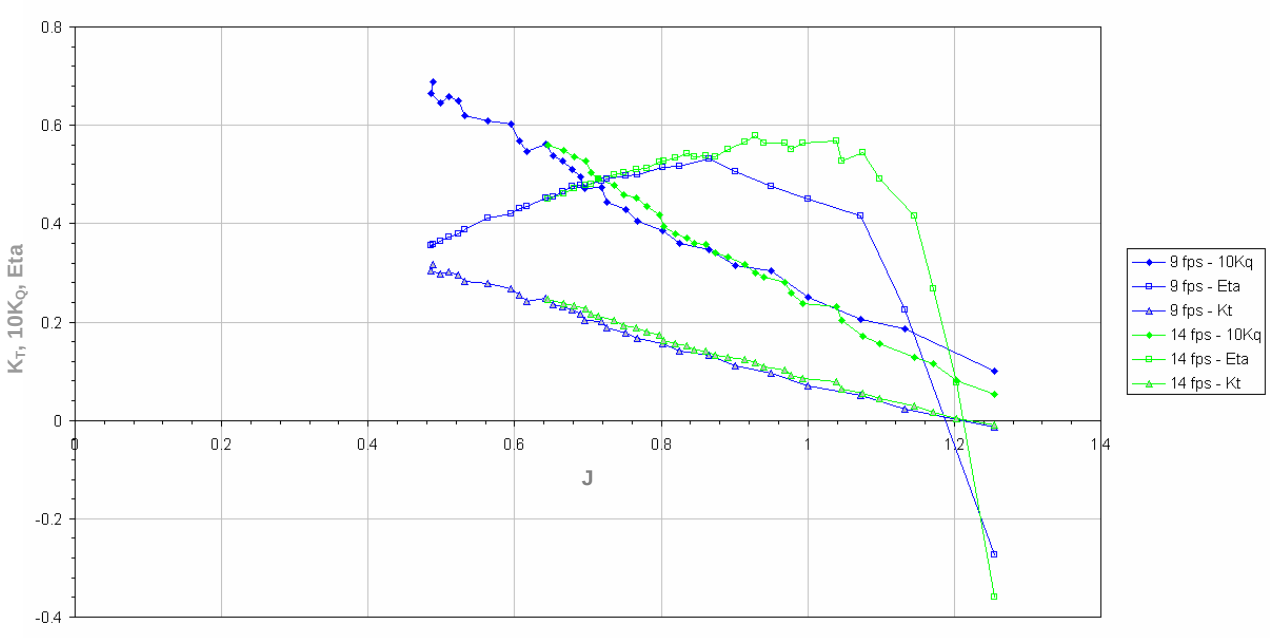

Figure 3-5. The propeller performance curves for propeller Z55.

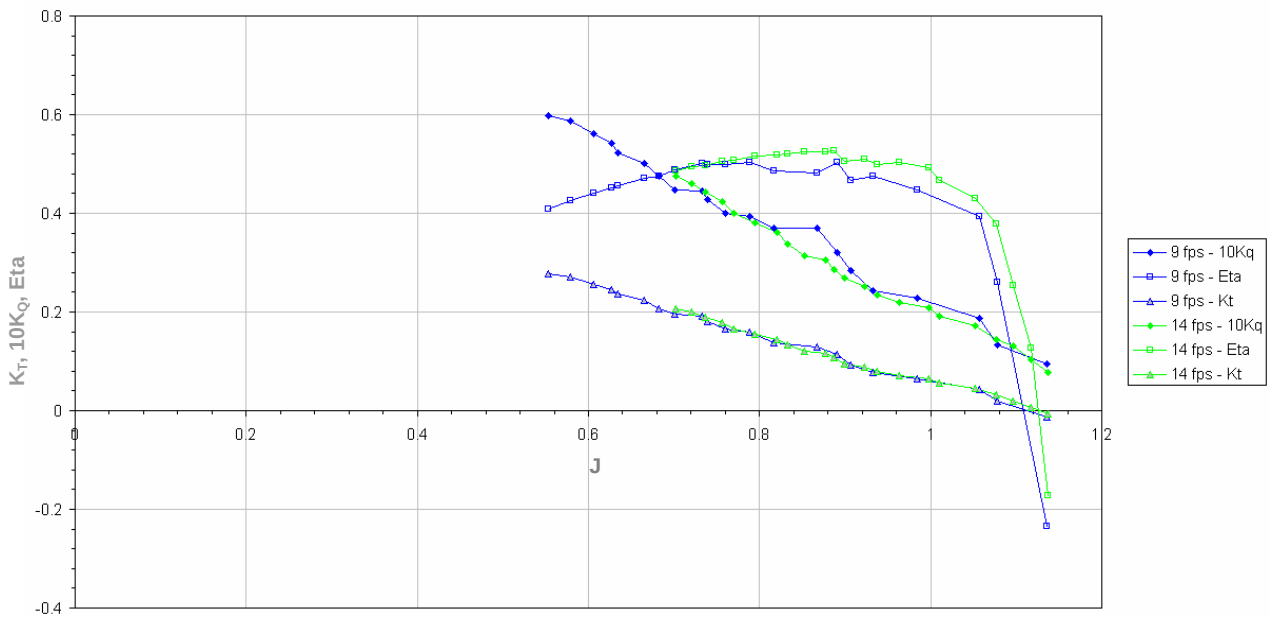

Figure 3-6. The propeller performance curves for propeller 1462.

The propeller performance curves, being non-dimensional, are independent of speed. After running the tests it was found that the data produced curves that were not independent of speed. In looking at the above figures, it is apparent that the curves do not fall on top of each

other for both the 9 ft/s and 14 ft/s tests. After examining the test rig and software for possible errors, it was discovered that the cause of the error was that the hole in the front of the shroud was too large. When the tests were first being run, the shaft kept rubbing against the hole in the front of the shroud, so the hole was drilled out to be slightly larger. Increasing the hole size effectively allowed water to flow into the shroud and caused a "pressurization effect" inside of the shroud. This pressurization effectively pushed the electronics section towards the balance section, giving the balance an unknown amount of negative thrust. Different hardware methods were applied in attempts to fix the problem, but with the amount of time available, an effective fix could not be found. Therefore, it was necessary to determine a method to correct the data.

Once the cause of the data discrepancy was identified, a method for correcting the data was implemented. The rig was run in the water tunnel without a propeller at 9 ft/s and 14 ft/s. The axial force seen by the rig was recorded. These measurements were repeated numerous times. It was found that depending on where the shaft settled in the hole made a difference in the recorded axial force. Unfortunately, it was not possible to control the location of the shaft after the rig was put into the tunnel. Although when the rig was assembled the shaft was always centered in the hole, the shroud would shift slightly when the tests were run. This meant that the shaft would often end up touching the sides of the hole. If the test setup was ideal, the balance should not have seen any axial force at all. Each time the test was run, however, the axial force seen by the balance at 9 ft/s was approximately 67% of the axial force seen at 14 ft/s. Therefore, the correction was applied so that the 9 ft/s data had 67% of the correction of the 14 ft/s data. It was possible to determine how much force was necessary to correct the data since, for each propeller, the 9 ft/s and 14 ft/s data sets should give the same performance curves. The axial force data for the single propeller runs was manually corrected until the K_T curves matched. The corrected propeller curves for propellers 2.0, 1457, and Z55 are shown in Figure 3-7, Figure 3-8 and Figure 3-9, respectively. The K_T and $10K_Q$ curves have been fit with third order trendlines in the figures. These equations were then used as inputs to the counter-rotating prediction programs. The K_T and $10K_Q$ curve equations for each tested propeller are shown in Table 3-3.

Table 3-3. The K_T and $10K_Q$ third order trendlines.

Propeller	K_T	$10K_Q$
1457	$-0.7475J^3 + 1.9283J^2 - 2.0641J + 0.9681$	$-0.9236J^3 + 2.4553J^2 - 3.0937J + 1.7767$
1462	$-0.0475J^3 + 0.2747J^2 - 0.834J + 0.6734$	$0.1821J^3 - 0.3066J^2 - 0.7631J + 1.0905$
2.0	$0.0328J^3 - 0.0108J^2 - 0.5155J + 0.7327$	$0.2391J^3 - 0.6129J^2 - 0.7371J + 1.8599$
2352	$-0.0291J^3 + 0.1371J^2 - 0.493J + 0.8437$	$0.0015J^3 - 0.0824J^2 - 0.4723J + 2.4157$
2.1/3	$0.1152J^3 - 0.4397J^2 + 0.0464J + 0.6971$	$0.4168J^3 - 1.7006J^2 + 0.9717J + 1.6318$
1455	$0.4103J^3 - 1.0053J^2 + 0.3261J + 0.3703$	$1.2519J^3 - 3.1696J^2 + 1.6308J + 0.5714$
1755	$0.0155J^3 - 0.0213J^2 - 0.3528J + 0.4955$	$0.088J^3 - 0.0844J^2 - 0.803J + 1.3008$
2055	$-0.0092J^3 + 0.0437J^2 - 0.4532J + 0.7347$	$0.0052J^3 - 0.0157J^2 - 1.1393J + 2.19$
Z55	$0.1486J^3 - 0.2762J^2 - 0.2836J + 0.5001$	$0.4171J^3 - 0.9068J^2 - 0.2116J + 0.9436$

The corrected curves for all tested propellers are shown in Appendix B.

The corrected figures show that each propeller reached a maximum efficiency of approximately 60%. This was judged to be reasonable.

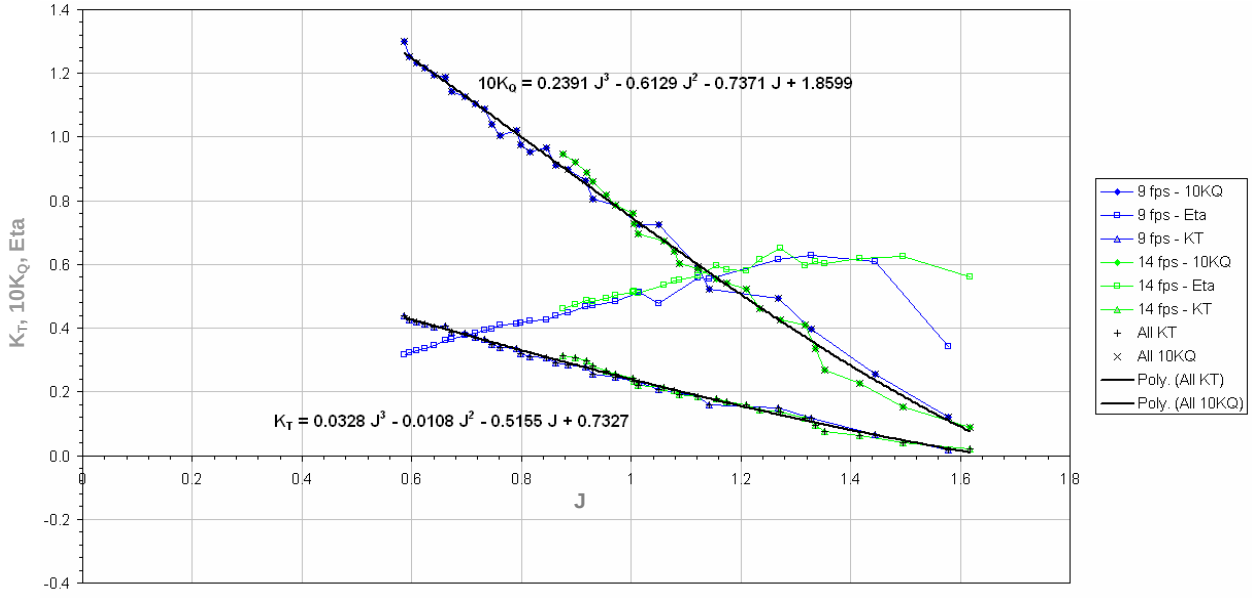

Figure 3-7. The corrected propeller performance curves for propeller 2.0.

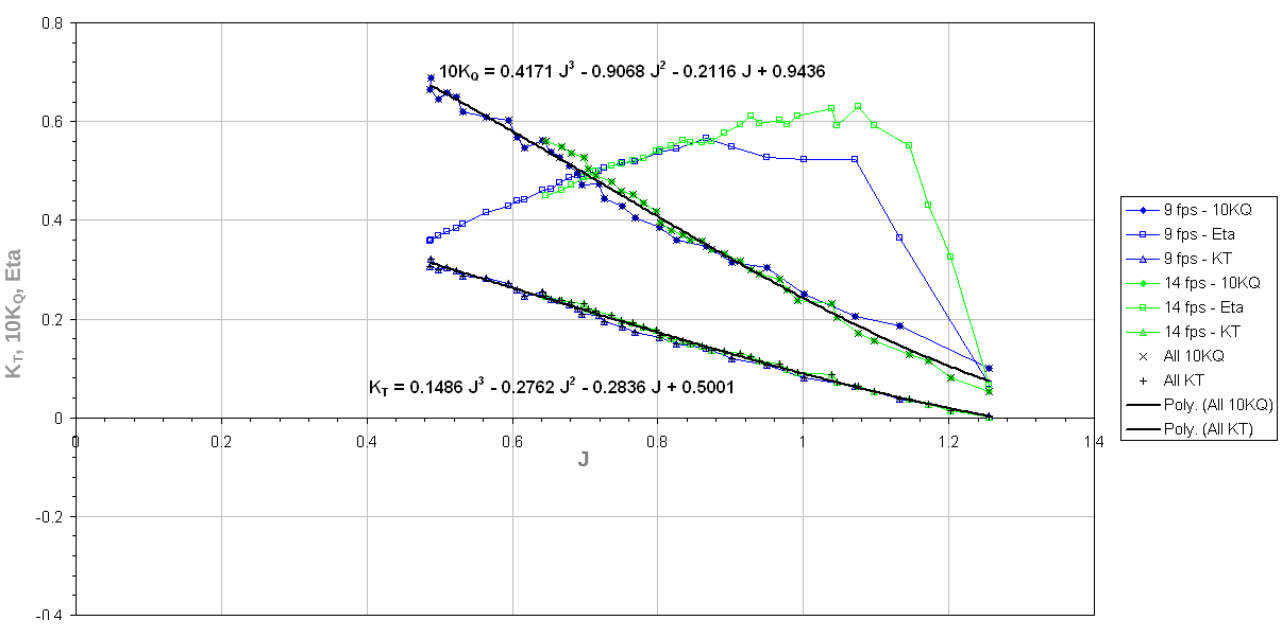

Figure 3-8. The corrected propeller performance curves for propeller Z55.

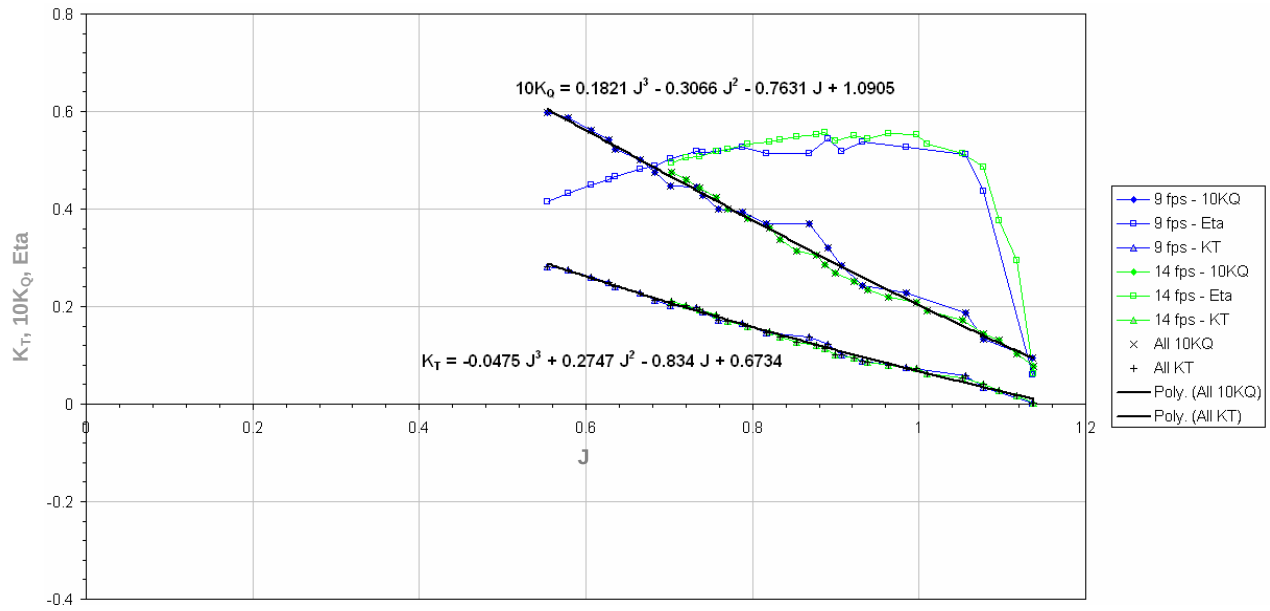

Figure 3-9. The corrected propeller performance curves for propeller 1462.

The value of J when K_T is zero is the observed P/D. It was found that the observed P/D was never the same as the advertised P/D for each propeller. Table 3-4 shows the predicted and observed P/D values.

Table 3-4. The advertised and observed P/Ds for the tested propellers

Propeller	Advertised P/D	Observed P/D
1457	1.4	1.15
1462	1.4	1.15
2.0	1.8	1.65
2352	2.3	2.55
2.1/3	1.8	1.85
1455	1.4	1.3
1755	1.7	1.4
2055	2.0	1.8
Z55	1.6	1.25

3.4 Wageningen B Matching

One of the interests of this study was to see if it is possible to predict the shape of unknown propeller performance curves using the Wageningen B series. The Wageningen B series is a series of propellers for which a set of regression equations for the K_T and $10K_Q$ curves have been determined. These equations are dependent on J, P/D, A_E/A_0 (blade area ratio), and Z (number of blades). The blade area ratio relates the size of the propeller blade to the propeller disc area. Although this parameter was not given for the Octura propellers, it could be calculated by determining the combined projected area of the propeller blades and dividing it by the disc area. The idea was that if it was possible to predict the propeller performance curves of the Octura propellers using only diameter, pitch, blade area ratio and number of blades as inputs, the propeller performance curves for all of the Octura propellers could be generated. If the propeller performance curves for all of the Octura propellers were known, the algorithm outlined in Chapter 5 could be used to predict the counter-rotating performance of any two Octura propellers.

Further examination, however, revealed that the Wageningen B series did not work well as a predictor for the Octura propellers. It was found that the Wageningen B series cannot predict the performance of propellers with large values of P/D. As this study required a high pitch aft propeller, the Wageningen B series was not helpful and this prediction method was abandoned.

Chapter 4 - Counter-Rotating Propeller Tests

Once the single propeller tests were finished, counter-rotating propeller tests were run in the Virginia Tech Water Tunnel. Since nine single propellers were tested (two forward propellers and seven aft propellers), there were fourteen possible counter-rotating sets. Due to time constraints, only nine counter-rotating sets were tested. The tested sets are shown in Table 4-1.

Table 4-1. Tested counter-rotating sets

	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Set 8	Set 9
Forward Propeller	1457	1457	1457	1457	1457	1462	1462	1462	1462
Aft Propeller	2.0	2.1/3	2055	2352	Z55	2.0	2.1/3	2055	Z55

4.1 Test Setup

The counter-rotating test setup was very similar to the single propeller setup. One of the major differences was that the rig was rotated 180 degrees. This put the propellers aft of the body with respect to the flow. Another major difference was that the shroud was not needed for the counter-rotating tests. A diagram of the counter-rotating test setup is shown in Figure 4-1. The figure shows the rig as if it had been cut through the center.

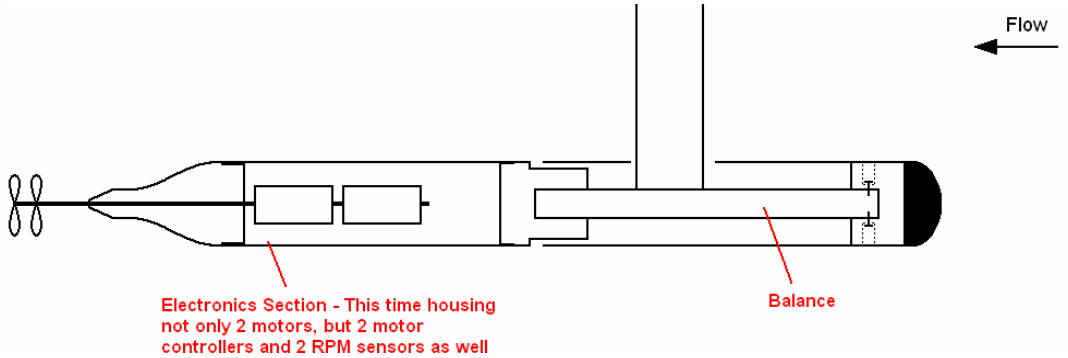

Figure 4-1. A view of the counter-rotating test setup.

The counter-rotating tests had two times the amount of electronics as the single propeller tests. Both the forward and aft propellers had a corresponding motor, motor controller and RPM sensor.

4.2 Testing

Since the propellers were located behind the body for the counter-rotating tests, preliminary tests were run to determine the drag on the rig. Once the drag data was known, the counter-rotating tests were run.

Before drag tests were begun, preliminary tests revealed that the strut was slightly deflecting the flow. Measurements of the wake behind the strut were made to ensure that the rig was aligned with the flow.

Drag Tests

To perform the drag tests, the counter-rotating rig was put in the water tunnel with no propellers. The tunnel velocity was stepped up from 0 ft/s to 16 ft/s in increments of approximately 2 ft/s. The axial force (drag) seen by the balance was recorded at each tunnel velocity. The drag seen by the balance was only the drag on the aft portion of the body. The drag results are shown in Figure 4-2.

The trendline is of the form $Drag = \alpha V^2$ where V is the flow velocity. The value of α in $\frac{lb \times s^2}{ft^2}$ was determined by minimizing the error between the trendline and data points. The final drag equation for the counter-rotating rig is shown in Equation (4-1).

$$Drag = -0.007407 \frac{lb \times s^2}{ft^2} V^2 \qquad (4\text{-}1)$$

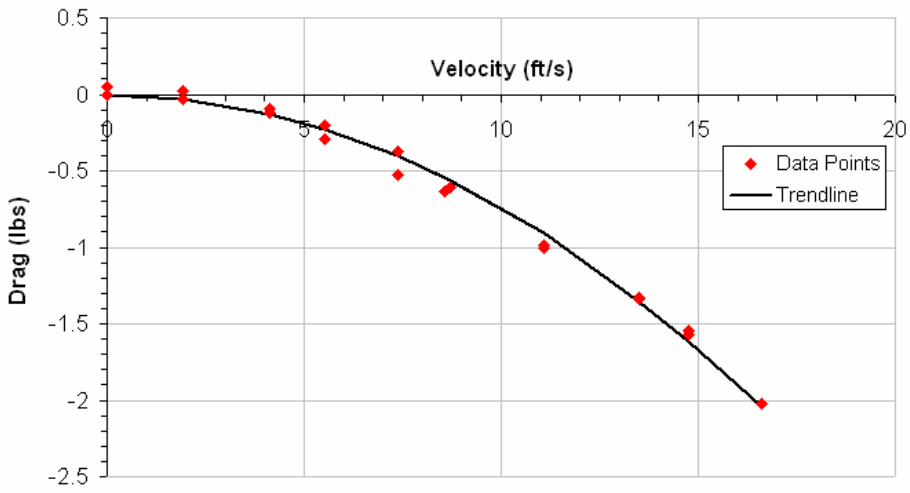

Figure 4-2. The drag data for the counter-rotating rig.

Counter-Rotating Testing

Once the drag data was known, the counter-rotating tests were run. Each counter-rotating set was tested at 4 ft/s, 9 ft/s and 14 ft/s. The tunnel velocity was held constant while the RPMs of the forward and aft propellers were changed (RPM_1 and RPM_2, respectively). Since the HSAUV was mainly interested in zero roll data, RPM_1 was stepped up in increments and RPM_2 was adjusted so that roll was near zero. For each test, velocity (as seen by the pitot tube), axial force and roll (as seen by the balance), and RPM_1 and RPM_2 (as seen by the RPM sensors) were recorded using LabView.

Results

Once the raw counter-rotating data was acquired, the drag force on the body was added to the measured counter-rotating axial force. This gave the total thrust produced by the counter-rotating propeller set. Since the drag data was calculated as a function of velocity, it was easy to determine the drag on the body at any speed. The most useful way to view the results is in a plot of thrust as a function of RPM_1 and RPM_2. The axial force results for propeller sets 1462R, 2.0 and 1462R, Z55 are shown in Figure 4-3 and Figure 4-4, respectively. In the figures, the left hand plots show the RPM_1 and RPM_2 paths that gave near zero roll for each propeller set. The right hand plots show the measured thrust as a function of RPM_1 and RPM_2 for those RPM values that gave near zero roll. The complete counter-rotating results can be seen in Appendix C.

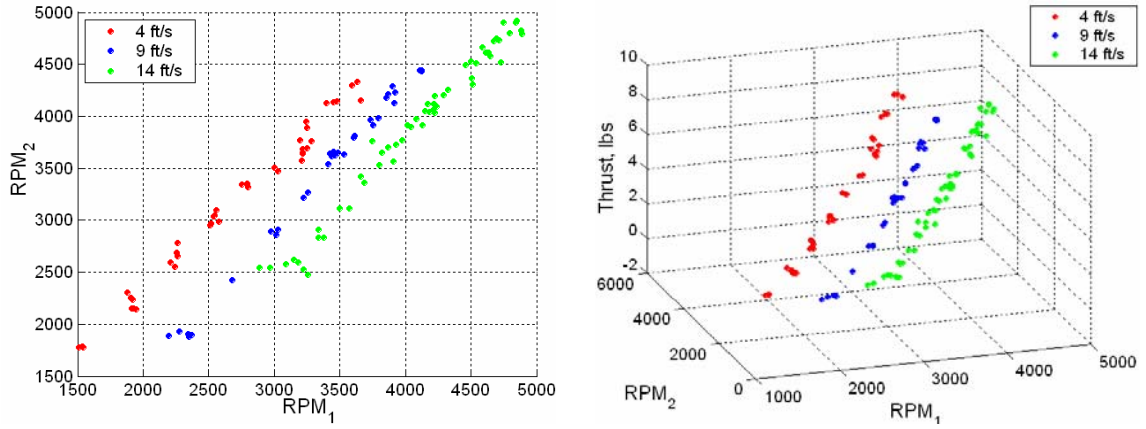

Figure 4-3. Counter-rotating test results for propellers 1462R and 2.0.

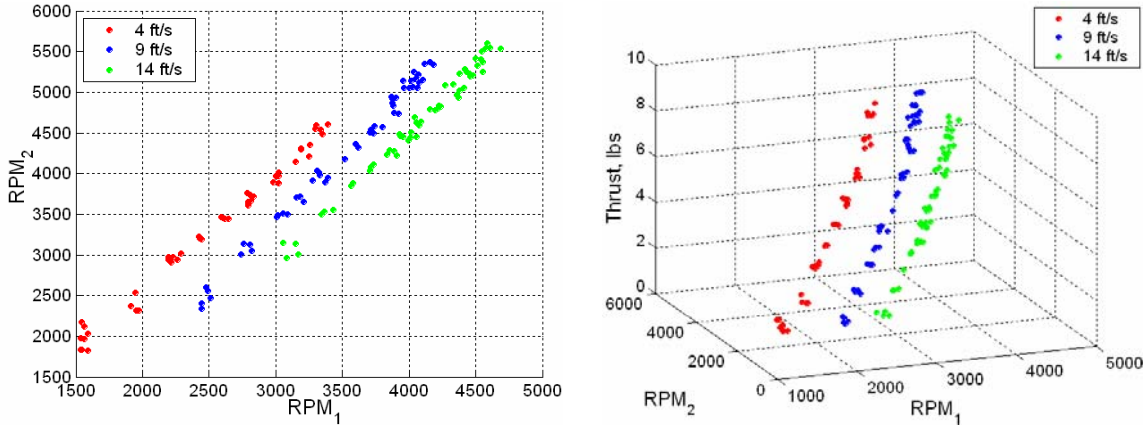

Figure 4-4. Counter-rotating test results for propellers 1462R and Z55.

Chapter 5 – Propeller Interaction Models

Once the single and counter-rotating propeller data was known, it was possible to begin building interaction models. The single propeller data was used to predict the counter-rotating performance. The effectiveness of the interaction models was determined by comparing the actual counter-rotating data with the predicted results.

Just as a note, the axial spacing between the forward and aft propellers was 1.5 inches (measured from the center of the hub of the front propeller to the center of the hub of the aft propeller). When the forward propeller of the counter-rotating set was the 1457R, the axial spacing was 70% of the forward propeller's diameter. When the forward propeller was the 1462R, the axial spacing was 61% of the forward propeller's diameter. This was the only axial spacing that was used for the tests. It would be expected that the propeller interaction coefficients would be dependent on the axial spacing.

5.1 Propeller Momentum Theory

Although a few different versions of the interaction model were tested before a solution was found, all models were based on propeller momentum theory. Momentum theory treats the propeller as a circular disc and assumes that the thrust is distributed uniformly over the area. The momentum equation is applied to a cylindrical control volume surrounding the propeller [8]. For this application, propeller momentum theory was first used to calculate the thrust and torque produced by the forward propeller. The thrust and torque produced by the aft propeller were then calculated using an increased value of the flow speed. The equations below outline the basic calculations that are the basis of all of the models [8].

The speed of advance, V_{A1}, of the forward propeller was calculated as:

$$V_{A1} = V(1 - w) \qquad (5\text{-}1)$$

where V is the flow speed and w is the Taylor wake fraction. The '1' subscript denotes that these calculations were done for the forward propeller.

The advance ratio, J_1, a non-dimensional velocity coefficient, was then calculated as:

$$J_1 = \frac{V_{A1}}{n_1 D_1} \quad (5\text{-}2)$$

where n_1 is the propeller RPM and D_1 is diameter.

K_{T1} and K_{Q1}, the thrust and torque coefficients were calculated from the propeller performance curves. The third order trendlines that were fit to the single propeller thrust and torque data were used to calculate values of K_{T1} and K_{Q1} at specific values of J_1.

The thrust and torque produced by the forward propeller were calculated using the values of K_{T1} and K_{Q1}:

$$T_1 = K_{T1} \rho n_1^{\,2} D_1^{\,4} \quad (5\text{-}3)$$

$$Q_1 = K_{Q1} \rho n_1^{\,2} D_1^{\,5} \quad (5\text{-}4)$$

Once the thrust and torque produced by the forward propeller were calculated, the flow speed increase induced by the forward propeller could be determined. It was necessary to determine this increase in flow speed in order to determine the speed of advance as seen by the aft propeller.

The thrust loading coefficient, C_{T1}, was calculated as:

$$C_{T1} = \frac{T_1}{1/2 \, \rho A_1 V_{A1}^{\,2}} \quad (5\text{-}5)$$

where A_1 is the area of the forward propeller and is calculated as:

$$A_1 = \pi \left(\frac{D_1}{2} \right)^2 \quad (5\text{-}6)$$

The ideal efficiency, η_{I1}, could then be calculated as:

$$\eta_{I1} = \frac{2}{1+\sqrt{C_{T1}+1}} \qquad (5\text{-}7)$$

Once η_{I1} was found, the increase in velocity over the forward propeller could be calculated. The percentage of velocity increase over the propeller is given the value a_1 and is calculated as:

$$a_1 = \frac{1}{\eta_{I1}} - 1 \qquad (5\text{-}8)$$

The value of a_1 is better defined by Figure 5-1 [8]. As the flow nears the propeller, it sees an increase in velocity. At the propeller, the velocity has increased by $a_1 V_{A1}$, or a_1%. Far downstream from the propeller, the velocity has increased by $2a_1 V_{A1}$.

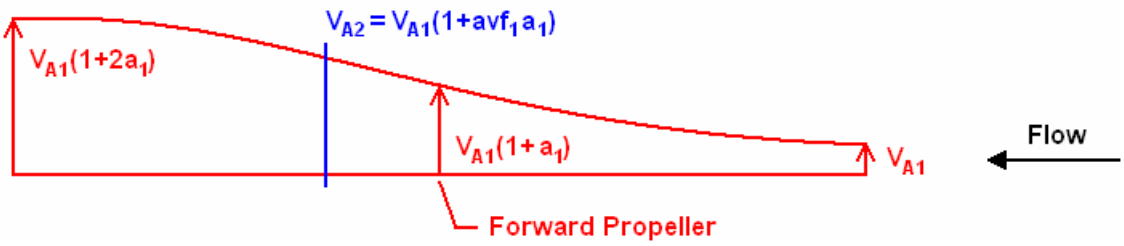

Figure 5-1. A visual representation of the value of a_1.

If the flow right behind the forward propeller has velocity $V_{A1}(1+a_1)$, then the flow at any distance behind the forward propeller will have velocity $V_{A1}(1+avf_1 a_1)$. The avf_1 is an axial velocity factor that takes into account the how far behind the forward propeller the flow velocity is being calculated. The flow velocity behind the forward propeller must be between $V_{A1}(1+a_1)$ and $V_{A1}(1+2a_1)$, so avf_1 must be between 1 and 2. Since there is little axial spacing between the forward and aft propellers, the velocity seen by the aft propeller is $V_{A1}(1+avf_1 a_1)$ with avf_1 being equal to or just above 1. The speed of advance for the aft propeller is shown in Equation (5-9):

$$V_{A2} = V_{A1}\left(1 + avf_1 a_1\right) \quad (5\text{-}9)$$

where the '2' subscript denotes values calculated for the aft propeller.

The thrust and torque produced by the aft propeller were calculated in the same way as the thrust and torque of the forward propeller (Equations (5-2) through (5-4)). K_{T2} and K_{Q2} were determined using the propeller performance curves for the aft propeller.

Although these calculations were the basis of all of the interaction models, each model had slightly different flow modeling characteristics.

5.2 DOT Optimizer

Fortran was used to build and run all of the interaction models. Each model had propeller interaction coefficients that were used to account for flow effects that could not be measured or calculated. The number and type of interaction coefficients varied from model to model. The DOT (Design Optimization Tools) program from Vanderplaats Research & Development was used to optimize for the values of the interaction coefficients.

DOT optimized for the interaction coefficients by minimizing the error between the predicted counter-rotating performance and the actual counter-rotating data. Each model would calculate the total predicted thrust and torque produced by the propeller set at given values of RPM_1, RPM_2, and V. The RPM_1, RPM_2, and V values were given by the data points from the counter-rotating tests. Using the RPM_1, RPM_2, and V values from the data sets made it possible to compare the predicted thrust and torque values, $T_{predict}$ and $Q_{predict}$, with the measured thrust and torque values, T_{meas} and Q_{meas}. The difference between the predicted values and the measured values was calculated as:

$$T_{off} = T_{meas} - T_{predict} \quad (5\text{-}10)$$

$$Q_{off} = Q_{meas} - Q_{predict} \quad (5\text{-}11)$$

T_{off} and Q_{off} were calculated for each data point. The error function, *F(x)*, was defined as the sum of the squares of T_{off} and Q_{off}:

$$F(x) = \sum_{i=1}^{NP} \left[\left(T_{off} \right)_i^2 + \left(Q_{off} \right)_i^2 \right] \quad (5\text{-}12)$$

where *NP* represents the number of points in the data set. The DOT program optimized for the interaction coefficients by minimizing *F(x)*.

5.3 Propeller Interaction Model, Version 1

The first version of the counter-rotating propeller interaction model had four interaction coefficients. The first coefficient was *w*, the Taylor wake fraction. The wake fraction accounts for the fact that the propellers are operating in the wake of the model. As seen in Equation (5-1), the wake fraction decreases the flow velocity seen by the propeller. It was not possible to directly calculate the wake fraction from the counter-rotating tests, and pre-existing data on the wake fractions of AUVs and submarines was not available. Wake fractions are generally on the order of 0.1 – 0.4.

The second interaction coefficient was avf_1, the axial velocity factor.

The third and fourth interaction coefficients were thrust and torque factors for the aft propeller, Tf_2 and Qf_2, respectively. Since single propeller data was being used to predict counter-rotating performance, there had to be some sort of factor that allowed for thrust and torque corrections. Single propeller tests are open water tests that do not have wake effects or effects of another propeller in the flow field. Tf_2 and Qf_2 allow for the fact that the propellers will not operate as a counter-rotating pair as they will when they are in open water. Tf_2 and Qf_2 were incorporated into the model as:

$$T_2 = Tf_2 \left(K_{T2} \rho n_2^{\,2} D_2^{\,4} \right) \quad (5\text{-}13)$$

$$Q_2 = Qf_2 \left(K_{Q2} \rho n_2^{\,2} D_2^{\,5} \right) \quad (5\text{-}14)$$

As seen in Equations (5-13) and (5-14), Tf_2 and Qf_2 were simply used to boost or reduce the thrust and torque values.

Determining the Interaction Coefficients

The DOT optimizer was used to determine the optimal values of the four interaction coefficients. Interaction coefficients were fixed in the order that they appeared in the model. In this case, w was fixed first because its value effected every other calculation in the model. Avf_1 was the next coefficient to be fixed, followed by Tf_2 and Qf_2 (simultaneously fixed).

Table 5-1. The results of the first set of optimization runs for Version 1.

Forward Propeller	Aft Propeller	Speed (ft/s)	w	avf_1	Tf_2	Qf_2	$F(x)$	Number of Points	$F(x)$ per Point
1457R	2.0	4	0.0409	1	1.61	1.06	0.588	22	0.0267
		9	0.108	1	1.474	1.427	0.477	43	0.0111
		14	0.188	1	1.63	1.32	0.508	25	0.0203
	2.1/3	4	0.156	1	1.91	0.963	0.393	15	0.0262
		9	0.102	1	1.36	1.2	0.102	14	0.0073
		14	0.246	1	1.27	1.14	0.193	11	0.0175
	2055	4	0.274	1	1.94	0.617	1.2	27	0.0444
		9	0.081	1	1.34	1.42	0.877	37	0.0237
		14	0.182	1.08	1.43	1.18	1.58	43	0.0367
	Z55	4	0.04	1	1.56	1.11	0.29	19	0.0153
		9	0.165	1	1.36	1.14	0.21	27	0.0078
		14	0.218	1	1.31	1.123	0.567	33	0.0172
1462R	2.0	4	0	1	1.38	1.2	1.33	40	0.0333
		9	0.068	1	1.44	1.41	1.1	32	0.0344
		14	0.155	1	1.67	1.39	2.98	54	0.0552
	2.1/3	4	0	1	1.42	1.22	0.278	28	0.0099
		9	0.106	1.44	1.57	1.54	0.801	33	0.0243
		14	0.2	1	1.39	1.06	3.15	67	0.0470
	2055	4	0.023	1	1.27	1.14	0.55	32	0.0172
		9	0.0799	1	1.22	1.4	0.908	32	0.0284
		14	0.21	1.02	1.36	1.07	2.55	53	0.0481
	Z55	4	0	1	1.33	1.13	0.741	47	0.0158
		9	0.103	1	1.27	1.18	1.22	52	0.0235
		14	0.187	1.25	1.51	1.16	1.91	54	0.0354

The results of the first set of optimization runs are shown in Table 5-1. Since this is the first run, no interaction coefficients were fixed. Because each counter-rotating propeller pair had three different sets of data (the 4 ft/s data, the 9 ft/s data and the 14 ft/s data) Table 5-1 is actually

showing twenty-four different sets of optimization results. Each speed case of each counter-rotating pair was optimized separately in order to determine if any interaction coefficients had flow speed dependencies. The optimal w, avf_1, Tf_2 and Qf_2 values are displayed for each of the twenty-four data sets. $F(x)$ is the error function as defined in Equation (5-12). Since the number of data points varied with each data set, it was useful to calculate the '$F(x)$ per point' value as shown in the last column of the table.

When running these optimizations and comparing the effectiveness of different formulations, it was useful to have an overall error value to describe how well a particular optimization predicted the data. This value was taken to be the average of the $F(x)$/point values. For the case shown in Table 5-1, the average error value was 0.0261. This means that for this optimization, each point has an average square error of 0.0261.

The interaction coefficients were constrained in their optimization. Although there was a good idea as to what the interaction coefficients should be, determining the best constraints was simply done through trial and error. The constraints for this run are shown in Table 5-2.

Table 5-2. The constraints on the interaction coefficients.

Interaction Coefficient	Lower Bound	Upper Bound
w	0	0.4
avf_1	1	2
Tf_2	0.5	2
Qf_2	0.5	2

One phenomenon that was discovered with this first version of the model, and occurred consistently throughout other versions, was that the wake fraction showed a speed dependence. A speed dependent wake fraction would be expected because the size of the wake behind a body will vary with speed. As speed increases, the wake fraction should decrease because the wake will become more energetic and therefore more contained. This model, however, predicted that the wake fraction increased with speed. Intuitively, it does not make sense for the wake fraction

to increase with speed, but no data on wake fraction values behind AUVs or submarines could be found. All attempts to force the optimizer to yield a wake fraction that decreased with increasing speed were unsuccessful. Even in the later, more effective versions of the model, the wake fraction increased with increasing speed. Eventually, this paradoxical relationship between wake fraction and speed was simply accepted so that the optimization could continue. In the optimization displayed in Table 5-1, the wake fraction was monotonic with respect to speed.

In order to fix an interaction coefficient, the optimal values of the coefficient from the most recent optimization would be averaged. This average value would be the value that the coefficient was fixed at in the next optimization. For example, if the value of Tf_2 was going to be fixed after the run in Table 5-1, the average value of the optimal Tf_2 values in Table 5-1 would be calculated. For this case, the average Tf_2 was 1.46. Therefore, Tf_2 would be fixed at 1.46 in the next set of optimizations. In the case of speed dependent coefficients, the average value of the coefficient at each speed would first be found. A linear trendline would then be fit to the average values. This trendline would be used in the model to fix the interaction coefficient. Although the method of fixing interaction coefficients as overall averages may seem crude, it was a simple way to narrow down the interaction coefficients. If, once all of the coefficients were fixed, a model yielded good enough results to justify further examination, another program was built that could 'fine tune' the interaction coefficients. This version of the model did not warrant that in depth of an analysis, so it will not be discussed here.

Results

The final values of the interaction coefficients for the first version of the model are shown in Table 5-3. The wake fraction speed dependence is shown. The axial velocity factor had a final value of 1, which was expected since the forward and aft propellers had little axial spacing between them. The final value of the thrust factor was 1.44, meaning that the thrust values were all boosted by 44%. The torque factor also had a final value of 1.44.

As interaction coefficients were fixed and the optimization became more constrained, the average *F(x)*/point value increased. The average *F(x)*/point value associated with fixing each interaction coefficient is shown in Table 5-4. As seen in the table, there was not a significant increase in error associated with fixing either w or avf_1. There was, however, a significant increase in error when Tf_2 was fixed. When w and avf_1 were fixed, the error did not increase

significantly because Tf_2 and Qf_2 were still able to manipulate the calculated thrust and torque values to match the data. Once Tf_2 was fixed, however, the error jumped significantly because the model could no longer control the thrust values. The error did not see a large increase when Qf_2 was fixed because most of the torque values are very close to zero. Because all of the measured torque values are near zero, the dominant error for all data sets stems from the thrust calculations.

Table 5-3. The final values of the interaction coefficients for Version 1.

Interaction Coefficient	Final Value
w	$0.0121V + 0.0077$
avf_1	1
Tf_2	1.44
Qf_2	1.44

Table 5-4. The average *F(x)*/point values associated with fixing the interaction coefficients.

Fixed Interaction Coefficient(s)	Average *F(x)*/point
-	0.0261
w	0.0302
w, avf_1	0.0310
w, avf_1, Tf_2	0.2174
w, avf_1, Tf_2, Qf_2	0.2430

A fine tuning program would have been able to slightly reduce the final error, but it would not have been enough to make this version of the model acceptable.

A Matlab m-file was built that could plot the predicted thrust and roll surfaces as functions of RPM_1 and RPM_2. Plotting the data points with the surfaces gave a good visual

representation of how much error was associated with the final coefficients. For the final interaction coefficients of Version 1 (Table 5-3), the surfaces for three of the twenty-four cases have been included. Figure 5-2 is an example of a data set with a low *F(x)*/point value, Figure 5-3 shows a data set with a typical *F(x)*/point value, and Figure 5-4 shows the data set with the largest *F(x)*/point value.

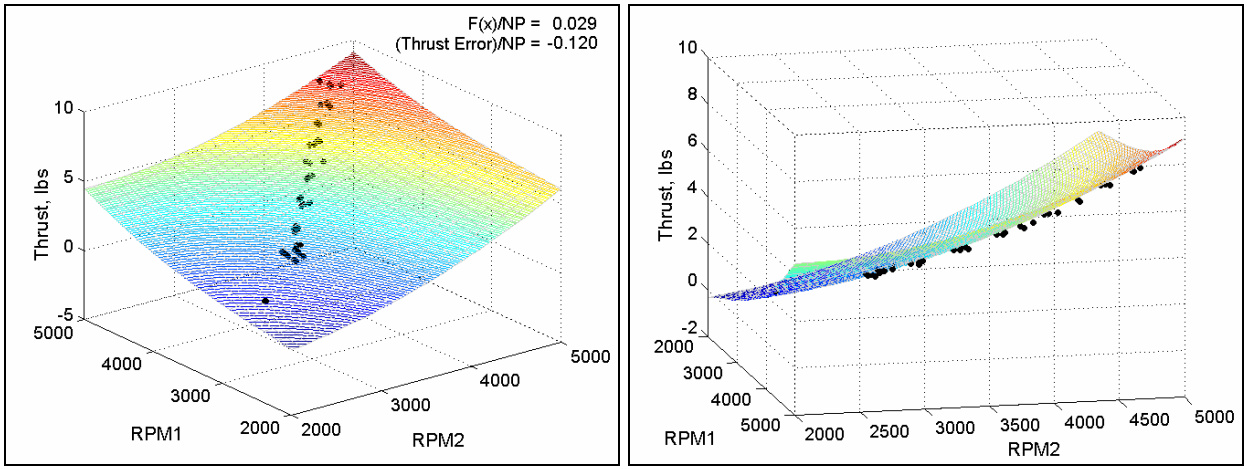

Figure 5-2. The thrust data points and predicted thrust surface for the 1457, 2.0 counter-rotating set at 9 ft/s.

In all three figures, the figure on the left shows the thrust data points and the predicted thrust surface. The right hand figure shows a side view of the surface and data points. The values in the upper right hand corner of the left hand figure are two different error values. The top value is the average *F(x)*/point as calculated by the Fortran model. The lower value is the average thrust error (T_{off}) per point in lbs. In Figure 5-2, the lower value indicates that the surface over-predicts each data point by approximately 0.12 lbs. In Figure 5-4, the large error example, the surface over-predicts each data point by approximately 0.662 lbs. Although all three figures show examples of surfaces that over-predict the thrust, there are also surfaces that are not shown that under-predict the thrust. These three cases were simply chosen as examples of low, medium and high error cases. It should be noted that the T_{off}/point value is meaningful in these three figures because in all three figures, the points lay on only one side of the surface. The T_{off}/point value can be misleading on a figure in which the points lay on both sides of the surface.

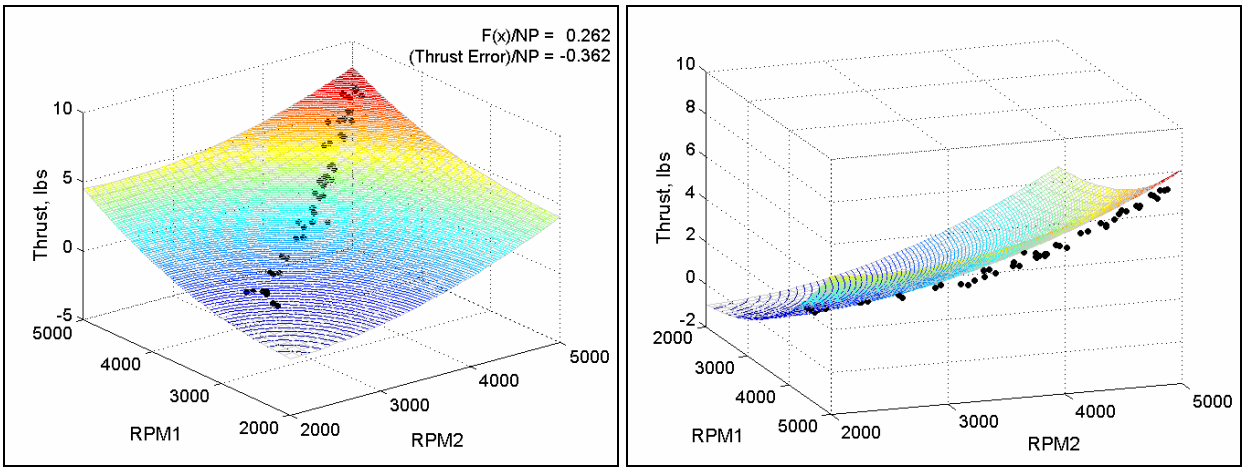

Figure 5-3. The thrust data points and predicted thrust surface for the 1462, 2.0 counter-rotating set at 14 ft/s.

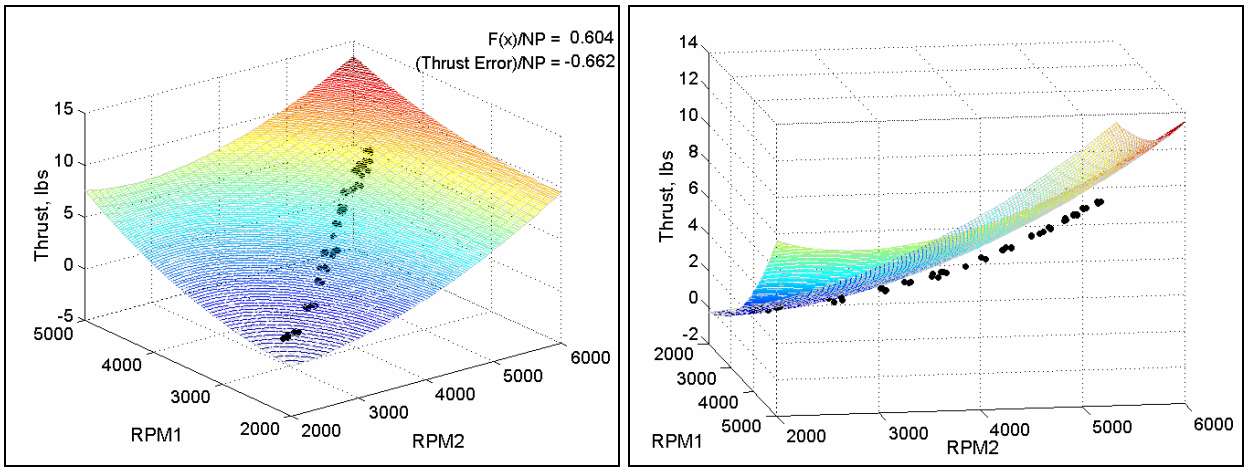

Figure 5-4. The thrust data points and predicted thrust surface for the 1462, Z55 counter-rotating set at 9 ft/s.

In all three figures it can be seen that the interaction model is not accurately predicting the thrust at higher RPMs. Although Tf_2 can boost or reduce the predicted thrust, it cannot fix RPM related predicttion problems.

A plot of a predicted torque surface and the corresponding data points is shown in Figure 5-5. The figure shows that when RPM_1 dominates RPM_2, the body will have a positive torque. Conversely, when RPM_2 dominates RPM_1, the body will undergo negative roll. Two different error values are again shown on the left hand figure. The top value is again the average *F(x)*/point for this case. The bottom value is the average torque error (Q_{off}) per point in in-lbs.

For this data set (the same set as in Figure 5-2), the torque surface over-predicts the average point by 0.046 in-lbs.

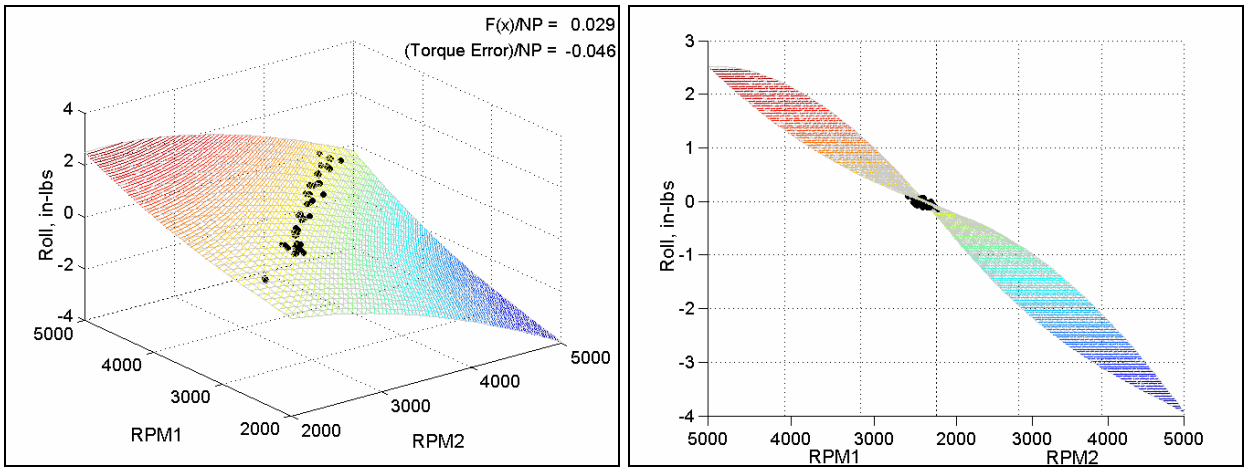

Figure 5-5. The torque data points and predicted torque surface for the 1457, 2.0 counter-rotating set at 9 ft/s.

After looking at the error values and the surface figures, it was decided that although Version 1 was a good start to the interaction model, it was not the desired solution.

Both Version 1 of the propeller interaction model and the 'ErrorSurface1.m' file (used to create the surface plots) are included in Appendix D. The data files corresponding to each counter-rotating propeller set that are needed to run both the Fortran and Matlab files are also included in Appendix D.

5.4 Propeller Interaction Model, Version 2

The second version of the counter-rotating propeller interaction model had five interaction coefficients. The first two interaction coefficients were w and avf_1.

The third coefficient was avf_2, an axial velocity factor that was applied to the forward propeller (this coefficient has the subscript '2' because it is denoting an effect from the aft propeller). This coefficient was included in order to model the effect of the aft propeller on the forward propeller. The value of avf_2 was determined in a similar manner as the value of avf_1. Because the aft propeller will speed up the flow ahead of it, the forward propeller will feel

suction from the aft propeller. This suction will cause an increase in V_{A1}. The new equation for V_{A1} is shown in Equation (5-15).

$$V_{A1} = V(1-w)(1+avf_2 a_2) \quad \textbf{(5-15)}$$

The increase in V_{A1} due to the forward propeller operating in the flow of the aft propeller is very similar to the increase in V_{A2} seen by the aft propeller when operating in the flow of the forward propeller. Because the forward propeller is ahead of the aft propeller, the value of avf_2 must be between 0 and 1 (just like the value of avf_1 had to be between 1 and 2).

Adding avf_2 as an interaction coefficient meant that the value of a_2 had to be calculated. a_2 represents the increase in flow speed over the aft propeller. The equations leading up to and including the calculation of a_2 are shown below:

$$C_{T2} = \frac{T_2}{\frac{1}{2}\rho A_2 V_{A2}^{\ 2}} \quad \text{(5-16)}$$

$$\eta_{I2} = \frac{2}{1+\sqrt{C_{T2}+1}} \quad \text{(5-17)}$$

$$a_2 = \frac{1}{\eta_{I2}} - 1 \quad \text{(5-18)}$$

Because the value of a_2 could not be found until the value of T_2 was calculated, and T_2 could not be calculated until a_2 was known, calculating a_2 had to be an iterative process. To incorporate a_2 into the interaction model, a guess (a_{2guess}) was made for a_2 before the calculation of V_{A1}. The model would calculate the rest of the flow as described in Section 5.1. Once T_2 had been calculated, the thrust loading coefficient, C_{T2}, ideal efficiency, η_{I2}, and a_2 were calculated as shown in Equations (5-16) through (5-18). The calculated a_2 was compared with a_{2guess} and if the percent error between the two values was less than 0.1%, the optimizer would accept the value of a_2 and continue the optimization. Just as a note, the values of a_1 and a_2 were always on the order of 0.01 to 0.1.

The fourth interaction coefficient was $rpmf_2$, an adjustment factor for RPM_2. This coefficient was added to address the thrust modeling inaccuracies seen at higher RPMs in Version 1. After analyzing the model and thinking about the counter-rotating flow complexities, it was decided that the aft propeller may have seen a larger effective RPM than was measured because of the swirl generated by the forward propeller. The swirl, or ω_{swirl}, would have to be a function of the increase in velocity over the forward propeller (a_1), and RPM_1. A corrected RPM_2 value was therefore assumed to follow:

$$RPM_{2corr} = RPM_2 + \omega_{swirl} = RPM_2 + RPM_1 a_1 rpmf_2 \qquad (5\text{-}19)$$

The value of RPM_{2corr} was used to calculate all of the flow properties for the aft propeller. The Qf_2 interaction coefficient was eliminated from this model because of the addition of RPM_{2corr}.

The fifth interaction coefficient was again Tf_2, the aft thrust factor.

Determining the Interaction Coefficients

The DOT program was again used to optimize for all of the interaction coefficients. The results of the first optimization are shown in Table 5-5. Table 5-5 is set up in the same way as Table 5-1. The table shows that most of the interaction coefficients did not converge well in this version of the model. The wake fraction again showed a speed dependence, but it was not the linear dependence seen in Version 1. Although avf_2 generally stayed close to 1, there were a few runs in which the value was drastically low. The $rpmf_2$ ranged from as low as 0.1 to as high as 1.7. Even Tf_2 bounced back and forth between being below 1 and decreasing the thrust, and being above 1 and increasing the thrust. The average *F(x)*/point for this optimization was 0.0293.

Table 5-5. The results of the first optimization run for Version 2.

Forward Propeller	Aft Propeller	Speed (ft/s)	*w*	avf_1	avf_2	$rpmf_2$	Tf_2	*F(x)*	Number of Points	*F(x)* per Point
1457R	2.0	4	0.086	1	1	1.67	1.07	0.41	22	0.0186
		9	0.154	1	1	0.399	1.42	0.523	43	0.0122
		14	0.199	1	1	0.758	1.26	0.592	25	0.0237
	2.1/3	4	0.294	1.01	0.999	1.37	0.5	0.566	15	0.0377
		9	0.216	1	0.999	0.2003	1.29	0.0975	14	0.0070
		14	0.265	1.06	0.99	0.249	1.27	0.0855	11	0.0078
	2055	4	0.27	1	1	0.122	0.5	1.73	27	0.0641
		9	0.114	1	0.999	0.224	1.52	0.9	37	0.0243
		14	0.198	1	0.996	0.397	1.12	1.98	43	0.0460
	Z55	4	0.0915	1	1	1.7	1.09	0.415	19	0.0218
		9	0.196	1	1	0.305	1.24	0.289	27	0.0107
		14	0.23	1	0.988	0.427	1.28	0.505	33	0.0153
1462R	2.0	4	0.0487	1.04	0.981	1.72	1.29	0.504	40	0.0126
		9	0.0918	1	0.211	0.552	1.02	1.08	32	0.0338
		14	0.168	1	0.171	0.977	0.844	3.95	54	0.0731
	2.1/3	4	0.161	1	1	0.149	0.998	0.408	28	0.0146
		9	0.122	1	1	0.302	1.66	0.965	33	0.0292
		14	0.211	1	0.692	0.48	1.04	4.1	67	0.0612
	2055	4	0.198	1	1	0.0946	0.982	0.46	32	0.0144
		9	0.105	1	0.953	1.06	1.87	0.974	32	0.0304
		14	0.213	1	1	0.405	1.39	2.88	53	0.0543
	Z55	4	0.0953	1	1	0.154	1.16	0.611	47	0.0130
		9	0.129	1.09	0.828	0.318	1.4	1.56	52	0.0300
		14	0.193	1.02	0.0417	0.786	0.722	2.58	54	0.0478

The constraints for this run are shown in Table 5-6.

Table 5-6. The constraints on the interaction coefficients.

Interaction Coefficient	Lower Bound	Upper Bound
w	0	0.4
avf_1	1	2
avf_2	0	1
$rpmf_2$	0	2
Tf_2	0.5	2

In order to determine why the interaction coefficients were bouncing around within their respective ranges, a more in-depth look was taken at the importance of each coefficient. To do this, three of the interaction coefficients were fixed while the other two were cycled through their ranges. The value of *F(x)*/point was calculated at each set of interaction coefficients. The results were then plotted as a surface. The results for *w* and $rpmf_2$ are shown in Figure 5-6.

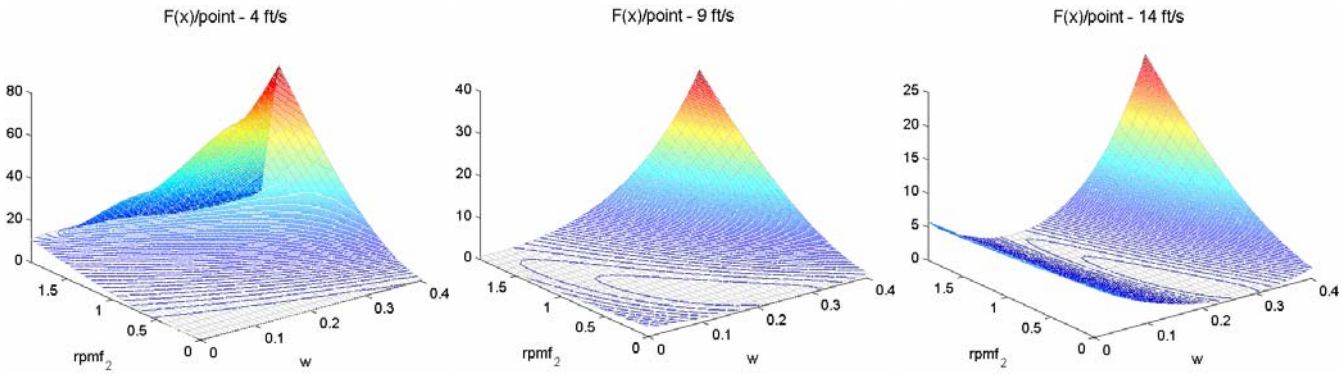

Figure 5-6. The *F(x)* surfaces for varying values of w and $rpmf_2$.

These surfaces were helpful in determining why the coefficients were not converging to one value. In looking at Figure 5-6, it can be seen that at low speeds, $rpmf_2$ was the dominant coefficient (between *w* and $rpmf_2$). The trough on the plot runs mostly parallel with the *w* axis. Therefore, once the right value of $rpmf_2$ was chosen, *w* could lay anywhere in that trough. As the speed increased, $rpmf_2$ and *w* became equally important. At high speeds, the roles of $rpmf_2$ and *w* switched and *w* became the dominant coefficient. At high speeds, as long as the correct value of *w* had been chosen, $rpmf_2$ could range anywhere between 0 and 2 and the error would not suffer greatly. Similarly, at low speeds, if the right value of $rpmf_2$ had been chosen, *w* was not very well constrained.

These plots were made for all combinations of the interaction coefficients and gave insight into why the coefficients were not converging.

Results

After analyzing the optimization results and coefficient interaction plots, it was determined that although the idea of the $rpmf_2$ was good, it was not being executed well in this

version of the interaction model. Although this version was not successful in predicting the thrust data, it was an important step towards the final version of the model.

Version 2 of the model is included in Appendix D. The Fortran program used to generate the error surface matrices as well as the corresponding 'speed2b.m' file (used to create the surface plots) are also included in Appendix D.

5.5 Propeller Interaction Model, Final Version

The third and final version of the interaction model has six interaction coefficients. The interaction coefficients are w, avf_1, avf_2, $rpmf_2$, Tf_2 and Qf_2. Because of the inclusion of avf_2, the a_2 iteration is still part of the model. All of the coefficients are defined as in Versions 1 and 2 except for $rpmf_2$.

Figure 5-7 shows a diagram of the velocity components seen by a section of the aft propeller blade. The main components, ω_2 and V_{A2}, account for the blade's forward speed and angular velocity. The smaller components, $a_1 avf_1$ and ω_{swirl}, are components of the forward propeller's downwash. In Version 2, ω_{swirl} was said to be a function of RPM_1 and a_1. This was only partially true. The amount of swirl will depend on the angle of attack of each blade section of the forward propeller, meaning that it will also depend on V_{A1}. Thinking about this physically, it makes sense for the swirl to be a function of RPM_1; if a propeller is operating in a flow with constant speed and the RPM is varied, the amount of swirl will definitely change. Conversely, it makes sense for the swirl to be a function of V; if a propeller is operating with a constant RPM and the flow speed is varied, the amount of swirl will also vary. Therefore, the ω_{swirl} equation from Version 2 was incorrect because it was not taking into account that swirl generated by the forward propeller will also depend on V_{A1}. A new and improved swirl term was created and incorporated into the RPM_{2corr} calculation. The new RPM_{2corr} is calculated as:

$$RPM_{2corr} = RPM_2 + rpmf_2 \frac{a_1 V_{A1} J_1}{0.7 R_1} \qquad (5\text{-}20)$$

where R_1 is the radius of the forward propeller.

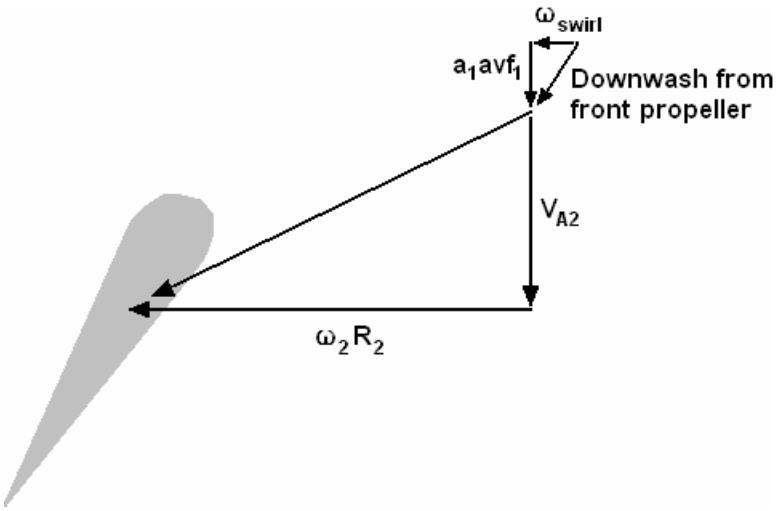

Figure 5-7. The velocity components of the flow as seen by a section of the aft propeller blade.

Determining the Interaction Coefficients

The DOT optimizer was used to determine the values of the interaction coefficients. The value of *w* was fixed first followed by avf_2, avf_1, $rpmf_2$, Tf_2 and Qf_2.

As more optimizations were run and more interaction coefficients were fixed, it was noticed that Version 3 was doing a phenomenal job of predicting the performance of six of the eight counter-rotating pairs. The performance of the other two pairs, however, was not predicted well by the chosen interaction coefficients. These two pairs were 1457R, 2055 and 1462R, 2055. Table 5-7 shows the *F(x)*/point values associated with each individual set of data with all six interaction coefficients fixed. The average *F(x)*/point for the sets containing propellers 2.0, 2.1/3 and Z55 is 0.064 while the average *F(x)*/point for the sets containing propeller 2055 is 0.56.

Because both of the 'bad' sets contained propeller 2055, the single propeller data sets for propeller 2055 were first examined. Since the shroud problems had required manual thrust corrections, it was thought that the problem would stem from the 2055 propeller performance curves. Examination of the curves, however, revealed nothing that seemed out of the ordinary. The six 2055 counter-rotating data sets were then examined. In the counter-rotating data sets, it was found that the velocity measurements for all six 2055 data sets were much lower than the velocity measurements for the rest of the data sets. The average velocity value of each of the twenty-four data sets is shown in Table 5-8. The three velocity averages (excluding the 2055 sets)

are 3.93 ft/s, 9.76 ft/s and 14.72 ft/s. The 2055 data set velocities fall far below these averages. It is difficult to say why these velocity measurements were not correct. The pressure transducer was a problematic piece of equipment during both single propeller and counter-rotating propeller tests. Looking back at the data, it was seen that the two 2055 counter-rotating sets were run on the same day as three other counter-rotating sets. It was suspected that these two sets were taken on the first tests of the day, but that still does not explain the velocity discrepancy. A reason for the discrepancy could not be found. To make the 2055 data consistent with the other counter-rotating data, a correction was applied to the six 2055 data sets to bring their velocities up to the average. The velocity values in each of the six sets were increased by a factor that brought the set's velocity up to the overall average. As a note, all of the tests for each speed were run at the same tunnel settings.

Table 5-7. *F(x)* results for Version 3 with all of the interaction coefficients fixed.

Forward Propeller	Aft Propeller	Speed (ft/s)	*F(x)*	Points	*F(x)* per Point
1457R	2.0	4	0.55	22	0.0250
		9	3.484	43	0.0810
		14	1.27	25	0.0508
	2.1/3	4	0.999	15	0.0666
		9	1.345	14	0.0961
		14	1.073	11	0.0975
	2055	**4**	**12.6**	**27**	**0.4667**
		9	**12.6**	**37**	**0.3405**
		14	**78.4**	**43**	**1.8233**
	Z55	4	0.33	19	0.0174
		9	1.57	27	0.0581
		14	0.681	33	0.0206
1462R	2.0	4	1.26	40	0.0315
		9	3.48	32	0.1088
		14	8.55	54	0.1583
	2.1/3	4	0.358	28	0.0128
		9	1.47	33	0.0445
		14	6.39	67	0.0954
	2055	**4**	**4.092**	**32**	**0.1279**
		9	**3.28**	**32**	**0.1025**
		14	**26.6**	**53**	**0.5019**
	Z55	4	0.618	47	0.0131
		9	3.83	52	0.0737
		14	4.984	54	0.0923

Table 5-8. Average velocity values for each data set.

Forward Propeller	Aft Propeller	Average Velocity Values at the Three Tested Speeds (ft/s)		
1457R	2.0	3.878	9.839	14.741
	2.1/3	4.053	9.846	14.911
	2055	**3.552**	**9.23**	**13.732**
	Z55	3.979	9.77	14.736
1462R	2.0	4.012	9.672	14.583
	2.1/3	3.965	9.662	14.606
	2055	**3.637**	**9.34**	**14.006**
	Z55	3.96	9.797	14.732

Once the correction had been made, the 2055 data sets began behaving like all of the other data sets and it was possible to fix interaction coefficients. While the interaction coefficients were being fixed, it was found that *w* was again showing a monotonic speed dependence. *w* was again increasing with increasing speed.

A speed dependence was also seen in the avf_2 coefficient. As speed was increasing, avf_2 was decreasing. This speed dependence was unexpected, but after thinking about the complexities of the counter-rotating flow a justification for the speed dependence was found. In order for the forward propeller to feel the flow increase from to the aft propeller, the flow increase must propagate forward. In slower flows, the flow increase will propagate more easily because the flow isn't providing as much resistance. In faster flows, the flow increase will not propagate as easily because the quickness of the flow is providing a lot of resistance. Therefore, in faster flows, the overall effect is similar to a larger axial spacing. In slower flows, the propellers are effectively right next to each other. This speed dependence is not seen in avf_1 because the flow increase from the forward propeller to the aft propeller is always moving in the same direction as the overall flow.

Once the interaction coefficients had been fixed, another Fortran program was written that could 'fine tune' the coefficient values. This program optimized for all twenty-four data sets

at the same time and had eight interaction coefficients. In this program, direct optimization for w and avf_2 was eliminated and was replaced with optimizing for the slopes and y-intercepts of the w and avf_2 linear trendlines.

Results

The values of the final interaction coefficients are shown in Table 5-9. These final interaction coefficients corresponded to a final *F(x)*/point value of 0.0612.

Matlab was again used to plot the data points and predicted thrust surfaces (as functions of RPM_1 and RPM_2). Examples of low error, typical error and large error data sets are shown in Figure 5-8, Figure 5-9, and Figure 5-10, respectively. The "(Thrust Error)/NP" value has been left off of these figures because the data points lay on both sides of the surfaces. In this case, such a value would be misleading.

Table 5-9. The final values of the interaction coefficients.

Coefficient	Final Value
w	$0.0129V - 0.023$
avf_1	1.0002
avf_2	$-0.0469V + 0.828$
$rpmf_2$	0.244
Tf_2	1.18
Qf_2	1.07

In looking at the figures, it can be seen just how well Version 3 predicts the counter-rotating data. In all three figures, the surface cuts right through the data points.

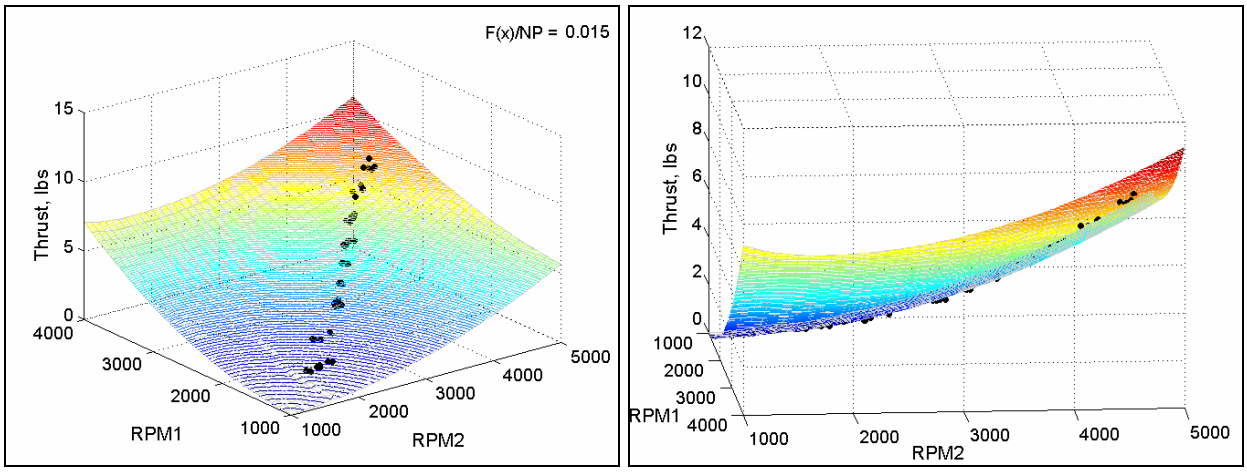

Figure 5-8. The thrust data points and predicted thrust surface for the 1462, Z55 counter-rotating set at 4 ft/s.

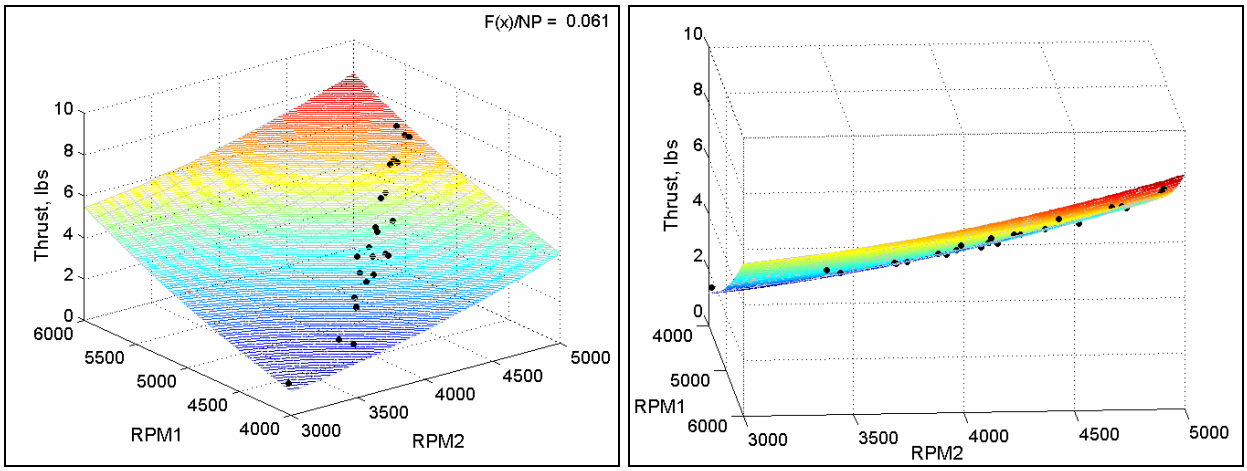

Figure 5-9. The thrust data points and predicted thrust surface for the 1457, 2.0 counter-rotating set at 14 ft/s.

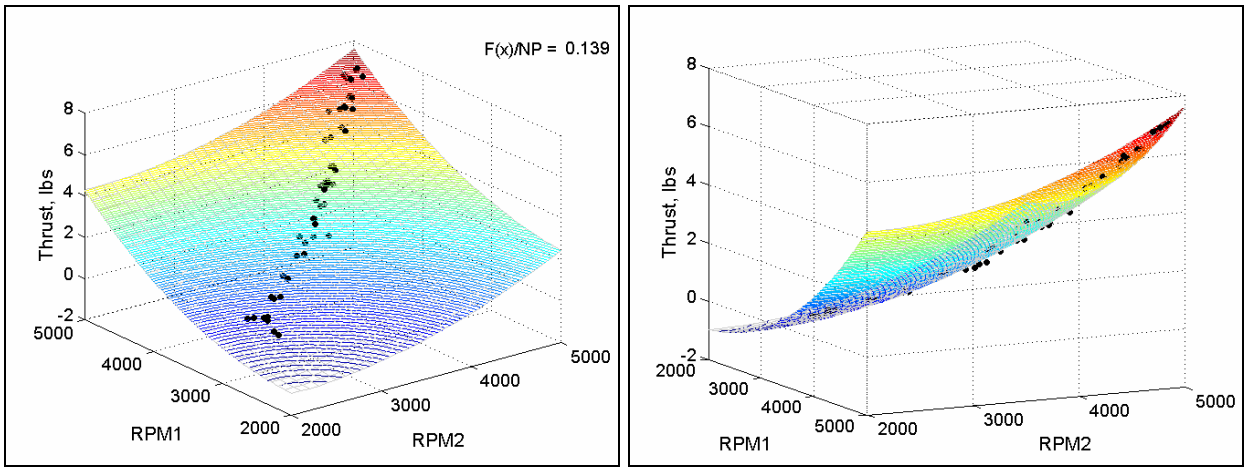

Figure 5-10. The thrust data points and predicted thrust surface for the 1462, 2.0 counter-rotating set at 14 ft/s.

The Fortran Version 3 model and 'ErrorSurface3.m' file are given in Appendix D.

Deliverables

Once the interaction method and interaction coefficients had been finalized, two Fortran deliverables were written. The first program, 'CR Prop Predictor.for', takes the diameter of the forward and aft propellers, the flow velocity, and the RPMs of the forward and aft propellers as inputs. The user is required to insert the propeller performance curve equations (K_T and K_Q) for the forward and aft propellers into function subroutines in the code. The program calculates the thrust and torque produced by the counter-rotating set at the given flow speed and RPMs. It also calculates an overall propulsive efficiency using Equation (5-21).

$$\eta = \frac{(T_1 + T_2)V}{2\pi n_1 Q_1 + 2\pi n_2 Q_2} \qquad \textbf{(5-21)}$$

The efficiency defined in Equation (5-21) is the total power out of the propeller divided by the total power into the propeller.

A sample run of the program is shown in Figure 5-11. This run was for the counter-rotating propeller set 1457R, 2.0 with both propellers operating at 5000 RPM. The program asks the user if they would like to enter their own value of *w* or use the default provided value. After

the thrust and torque have been calculated, the program prompts the user to enter ‘yes’ or ‘no’ with regards to whether or not they would like to calculate the thrust and torque at another point.

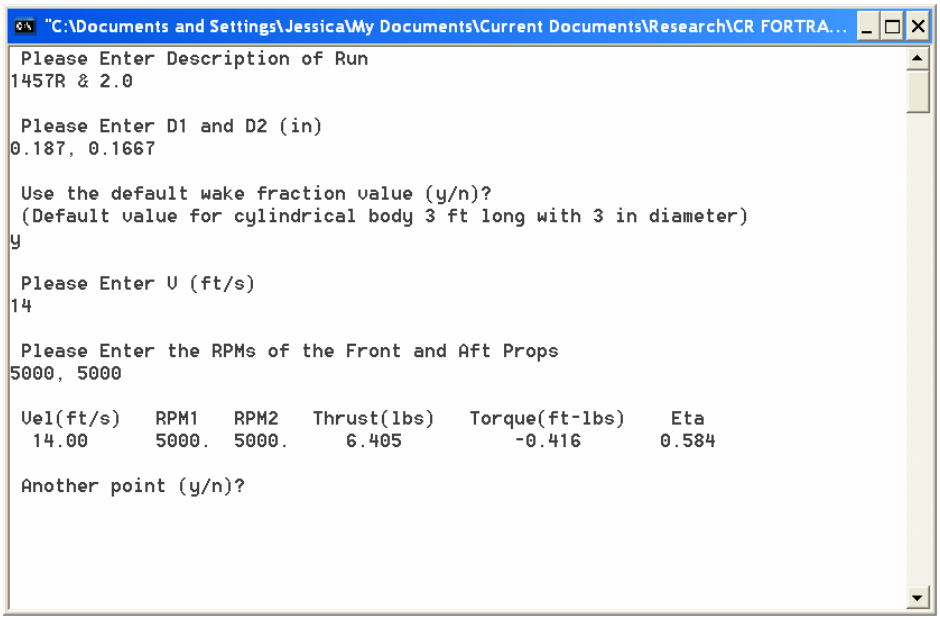

Figure 5-11. A sample run of ‘CR Prop Predictor.for’

The second program is titled ‘CR RPM Optimizer.for’ and takes as inputs the diameters of the forward and aft propellers, the propeller performance curves (manual user input, same as above), the flow velocity and the desired thrust. The program optimizes for the RPM_1 and RPM_2 values that will give the desired thrust with zero torque. This program also outputs the propulsive efficiency, η. A sample run is shown in Figure 5-12. The run is again for the 1457R, 2.0 counter-rotating propeller set.

The propulsive efficiency that was calculated in both programs was used to select the optimal counter-rotating pair for the HSAUV. Field tests of the AUV revealed that when the AUV was travelling at 10.5 knots, the RPMs of the forward and aft propellers were 6270 and 6600, respectively. These values were put into the CR Prop Predictor.for program in order to determine the associated thrust. It was found that HSAUV needed 10.8 lbs of thrust to travel at 10.5 knots. The thrust and velocity values were then put into the CR RPM Optimizer.for program and the propulsive efficiencies for each of the fourteen possible counter-rotating pairs were

calculated. These values were compared and the best pair for the HSAUV was selected. The counter-rotating pair that gave the highest efficiency was the 1462R, 2.1/3. with an efficiency of 66.4%. According to this prediction program, this is the counter-rotating pair that will work most efficiently for the HSAUV for the specified flow conditions.

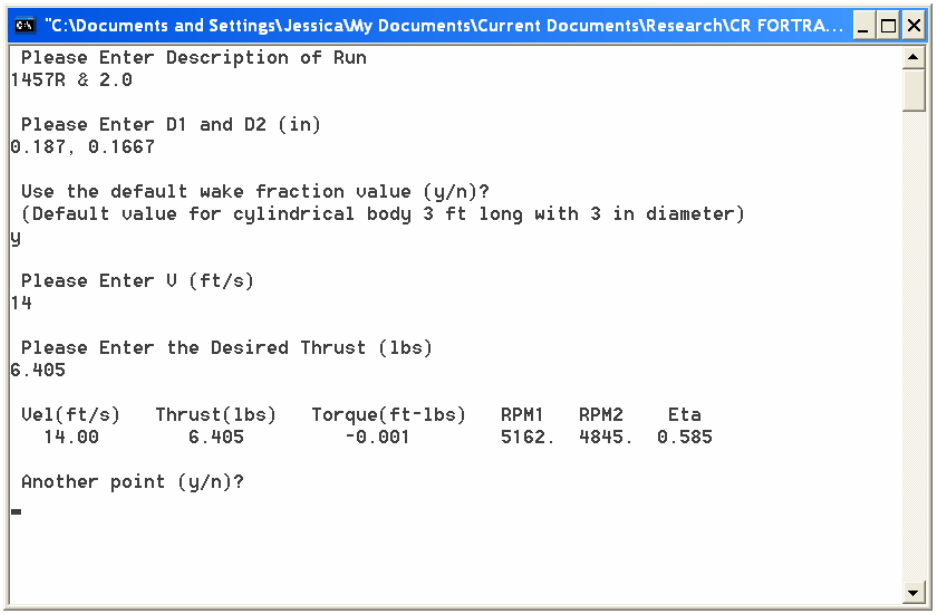

Figure 5-12. A sample run of 'CR RPM Optimizer.for'

Both 'CR Prop Predictor.for' and 'CR RPM Optimizer.for' are given in Appendix D.

Chapter 6 – Conclusions and Future Work

This goal of this project was to determine if single propeller data could be used to predict the counter-rotating propeller performance of two unmatched propellers and, if so, use this information to choose the most efficient counter-rotating pair for the HSAUV.

Single propeller tests were run in the Virginia Tech water tunnel in order to generate the propeller performance curves. The propellers were then run in counter-rotating sets and the counter-rotating data was recorded. The single propeller data was used in a momentum theory based algorithm to predict the counter-rotating performance. The counter-rotating data was used as a comparison to determine the effectiveness of each interaction model.

6.1 Results

The third version of the propeller interaction model proved to be a very accurate solution to this study. Two of the six interaction coefficients demonstrated linear speed dependencies (avf_2 and w) while the other four coefficients remained constant. A Matlab m-file was created that plotted the predicted counter-rotating thrust performance as a surface, and plotted the thrust data points on the surface. These surfaces showed that the prediction model was in fact a very accurate solution.

The interaction coefficient results were used to build two counter-rotating prediction programs in Fortran. The first program takes as inputs V, RPM_1 and RPM_2 and gives predicted thrust and torque values. The second program takes V and the desired thrust as inputs and outputs the RPM_1 and RPM_2 values that will generate the desired thrust with zero roll.

The 1462R, 2.1/3 counter-rotating pair was found to be most efficient for the HSAUV.

6.2 Future Work

Future work for this project would include a more in-depth look at the speed dependence demonstrated by *w*. Although the *w* speed dependence was needed to match the data, there is still no justification for the fact that *w* is increasing with increasing speed. A more thorough research search that encompassed wake fraction studies for AUVs and submarines would be useful. Water tunnel testing to determine AUV wake fractions at different flow speeds would also be an interesting side project.

Future work would also include a continuation of the study begun in Section 3.4 (Wageningen B Matching). The proposed solution is only useful for propellers for which the propeller performance curves are known. Propeller manufactures often do know propeller performance curves. Although the Wageningen B series was not able to predict the propeller performance curves, another series may yield better results.

Lastly, further work would include testing these results on different propellers (different brands, different diameters, etc.). This prediction method was only validated for Octura propellers with diameters under 3 inches, so it would be useful to know if these results could be applied to larger propellers or propellers from a different manufacturer.

References

[1] Ginzel, F., “Calculation of Counterrotating Propellers,” National Advisory Committee for Aeronautics, Technical Memorandum No. 1208, March 1949

[2] Playle, S. C., Korkan K. D. and von Lavante, E., “A Numerical Method for the Design and Analysis of Counter-Rotating Propellers,” *Journal of Propulsion and Power*, Vol. 2, Jan.-Feb. 1986

[3] Cox, B. D. and Reed, A. M., “Contrarotating Propellers – Design Theory and Application,” No. 15, SNAME Propellers ’88 Symposium, Virginia Beach, Virginia, September 1988

[4] Korkan, K. D. and Gazzaniga, J. A., “Off-Design Analysis of Counter-Rotating Propeller Configurations,” *Journal of Propulsion and Power*, Vol. 3, Jan.-Feb. 1987

[5] Reed, A. M., “Contrarotating Propeller Design Program,” Version 3.0, David Taylor Naval Ships Research and Development Center, Code 1544, March 1985

[6] Hecker, R. and McDonald, N.A., “The Effect of Axial Spacing and Diameter on the Powering Performance of Counterrotating Propellers,” David Taylor Model Basin, Hydromechanics Laboratory, Research and Development Report 1342, February 1960

[7] Carnvale, A. and Wyman S., “Low Velocity Water Current Facility,” Testing Division, Oceanographic Instrumentation Center, Naval Oceanographic Office, Washington D.C., 1968

[8] Lewis, E. V. (ed.), *Principles of Naval Architecture*, Vol. 2, Resistance, Propulsion and Vibration, SNAME, Jersey City, New Jersey, 1988

Appendix A – Baseline Velocity Results

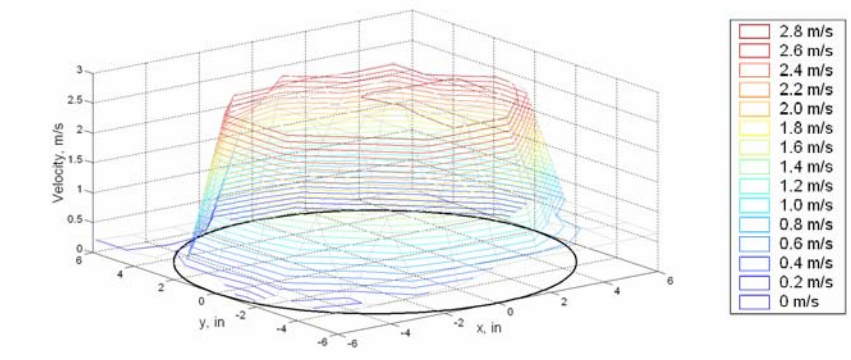

Figure A-1. Baseline results for the original '2 knot' setting at z = 4 inches.

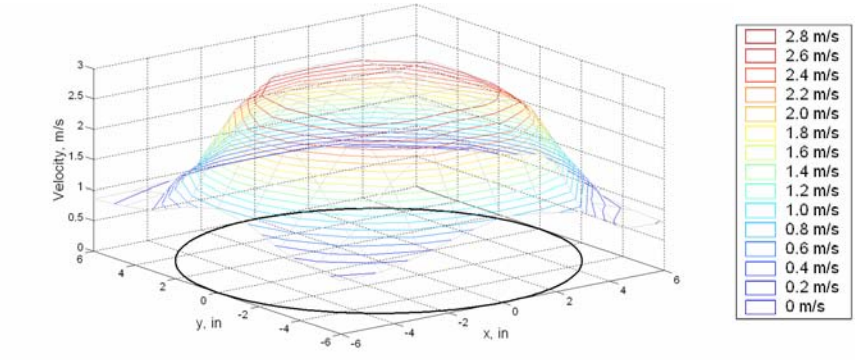

Figure A-2. Baseline results for the original '2 knot' setting at z = 16 inches.

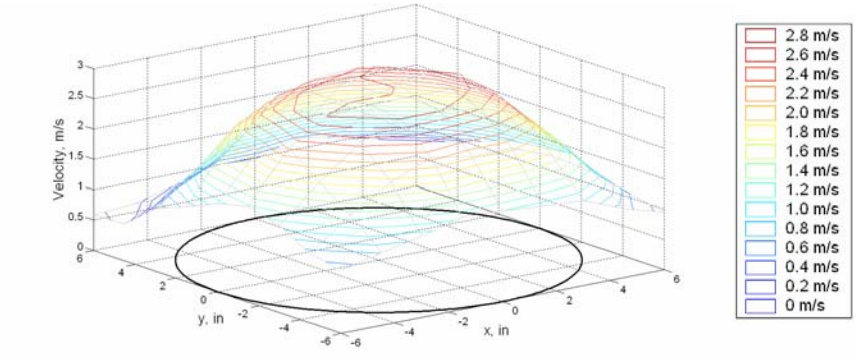

Figure A-3. Baseline results for the original '2 knot' setting at z = 27 inches.

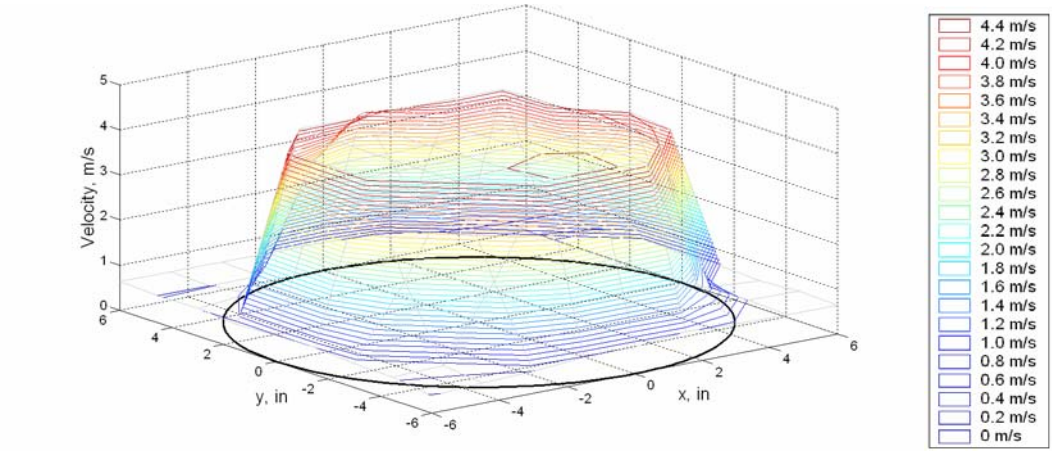

Figure A-4. Baseline results for the original '4 knot' setting at z = 4 inches.

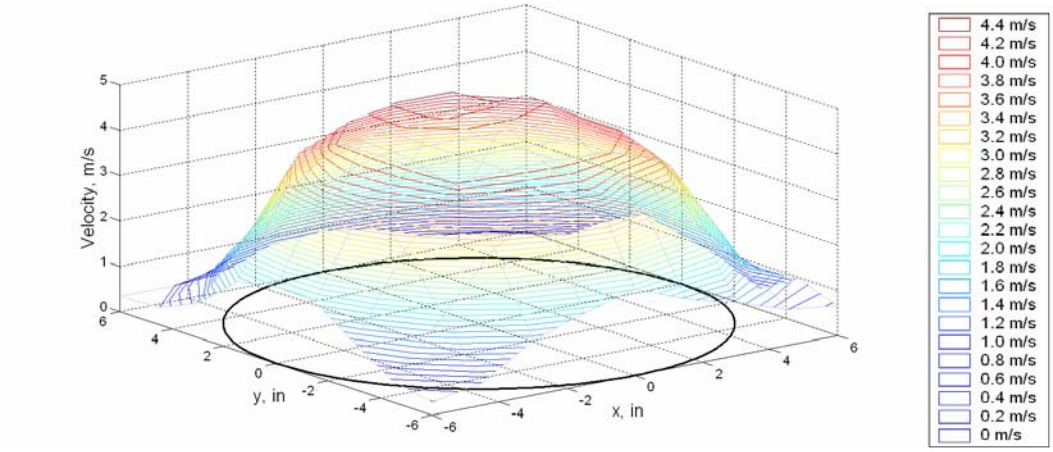

Figure A-5. Baseline results for the original '4 knot' setting at z = 16 inches.

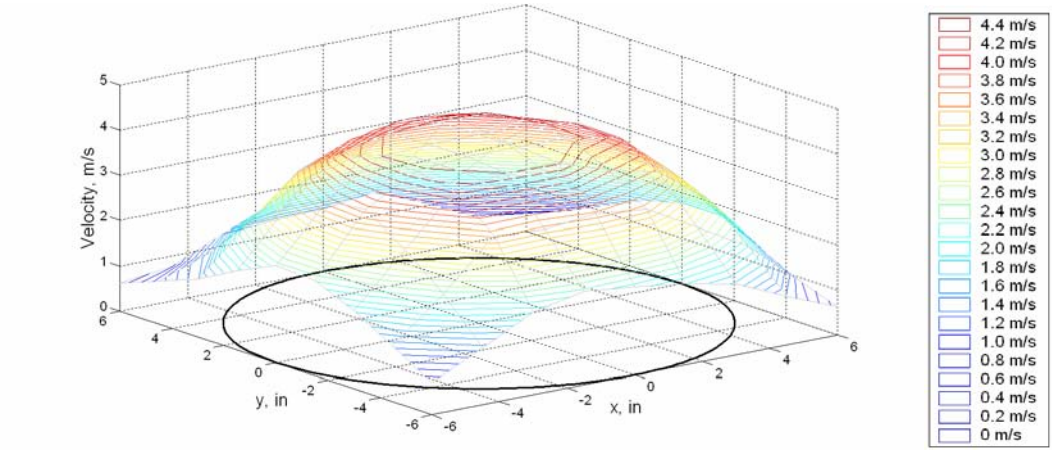

Figure A-6. Baseline results for the original '4 knot' setting at z = 27 inches.

Appendix B – Propeller Performance Curves

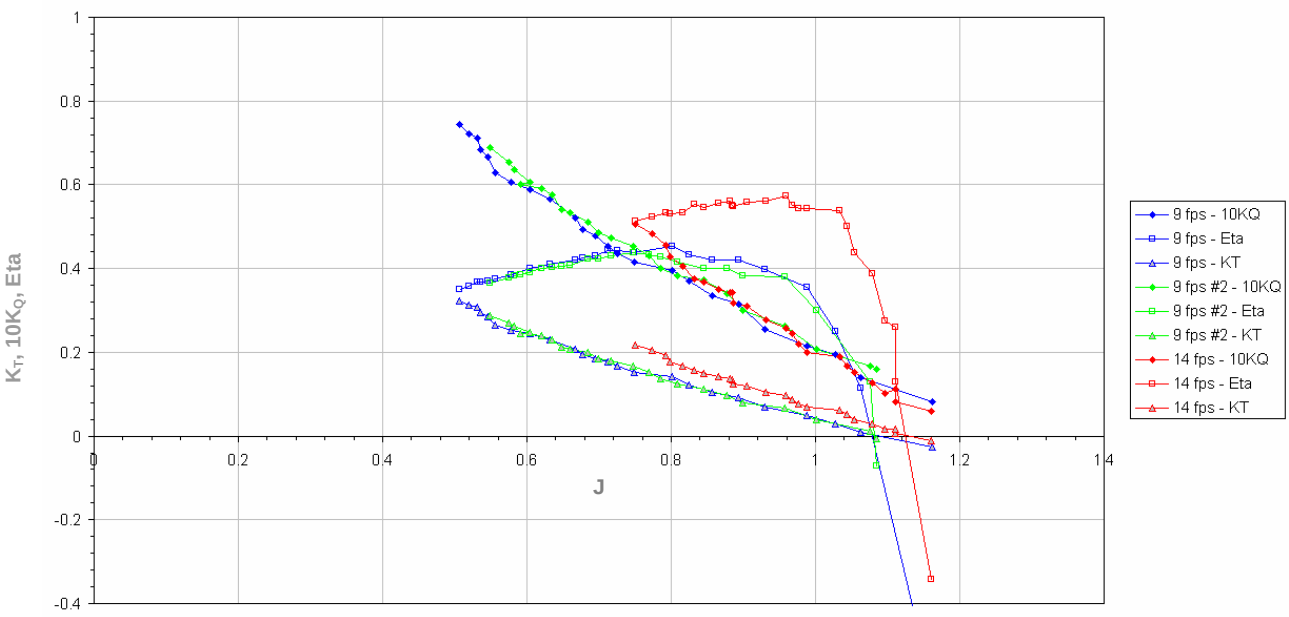

Figure B-1. The uncorrected propeller performance curves for propeller 1457.

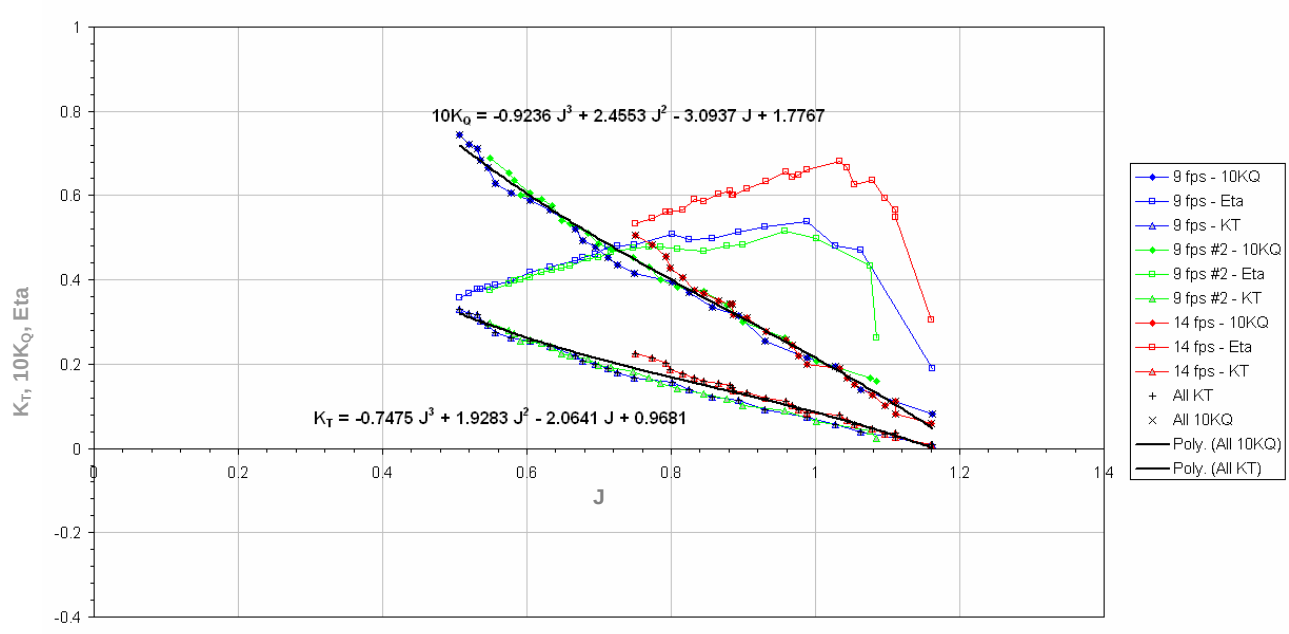

Figure B-2. The corrected propeller performance curves for propeller 1457.

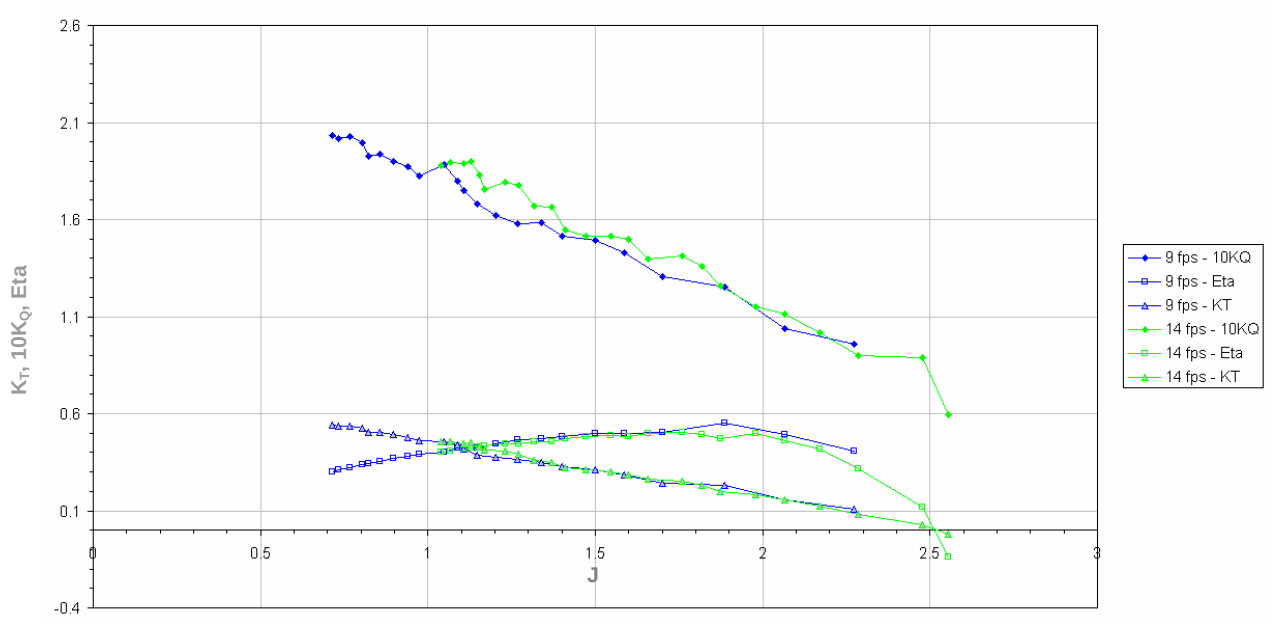

Figure B-3. The uncorrected propeller performance curves for propeller 2352.

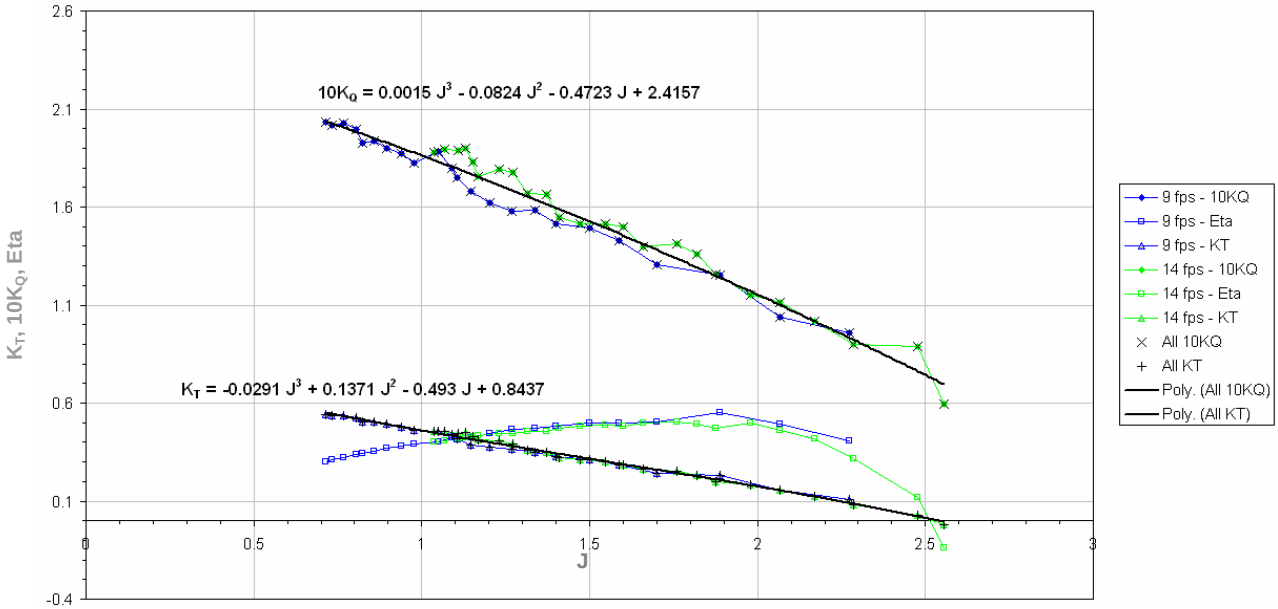

Figure B-4. The corrected propeller performance curves for propeller 2352.

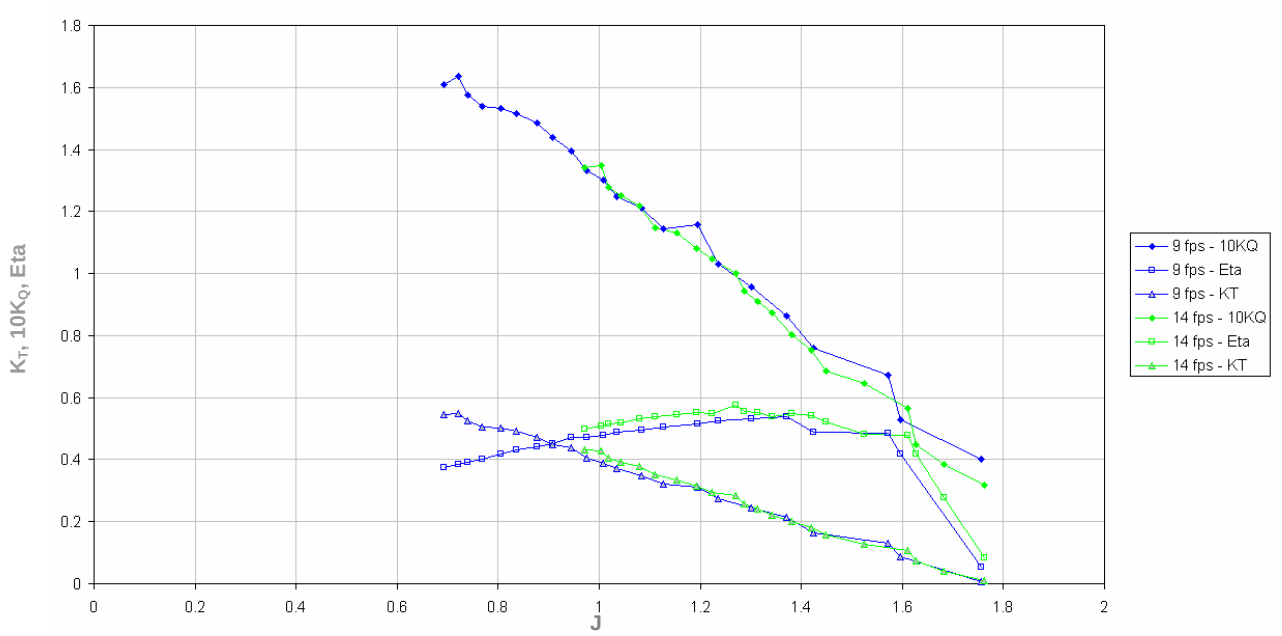

Figure B-5. The uncorrected propeller performance curves for propeller 2.1/3.

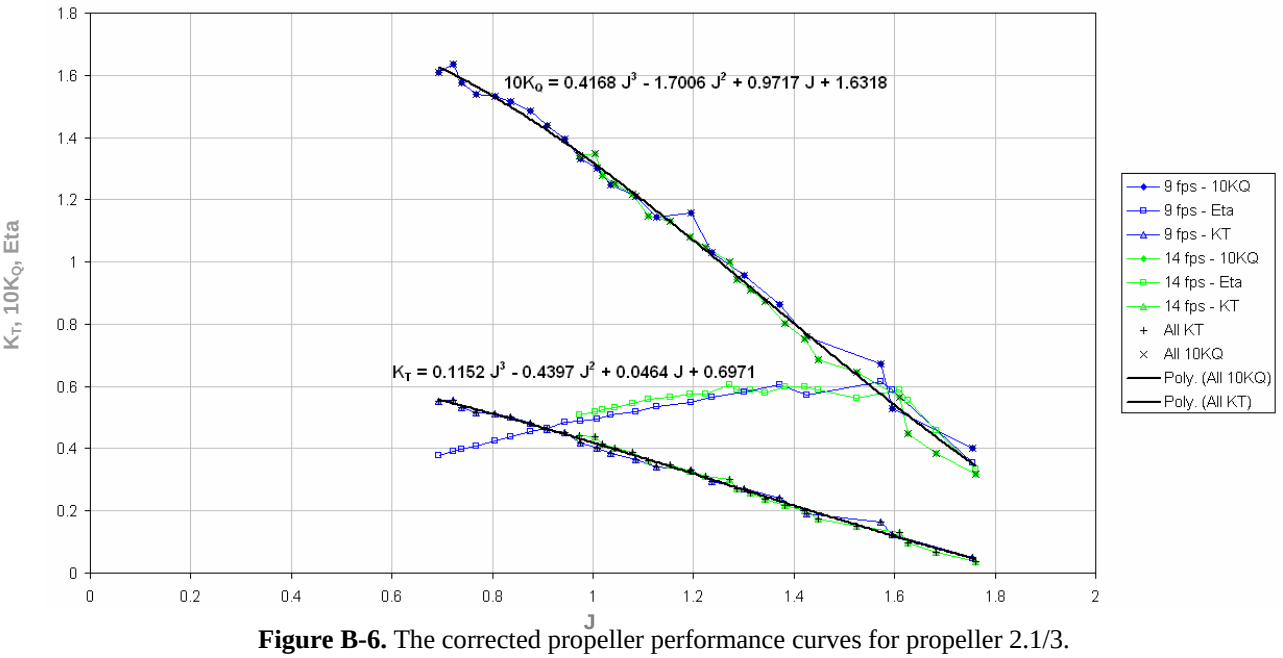

Figure B-6. The corrected propeller performance curves for propeller 2.1/3.

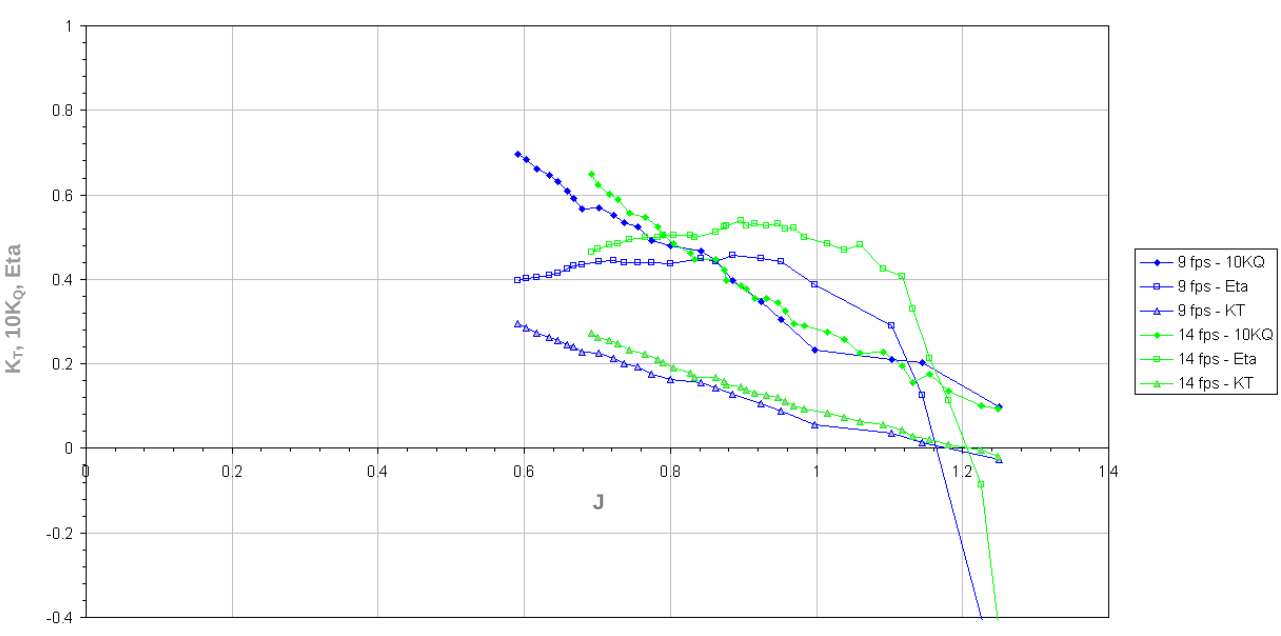

Figure B-7. The uncorrected propeller performance curves for propeller 1455.

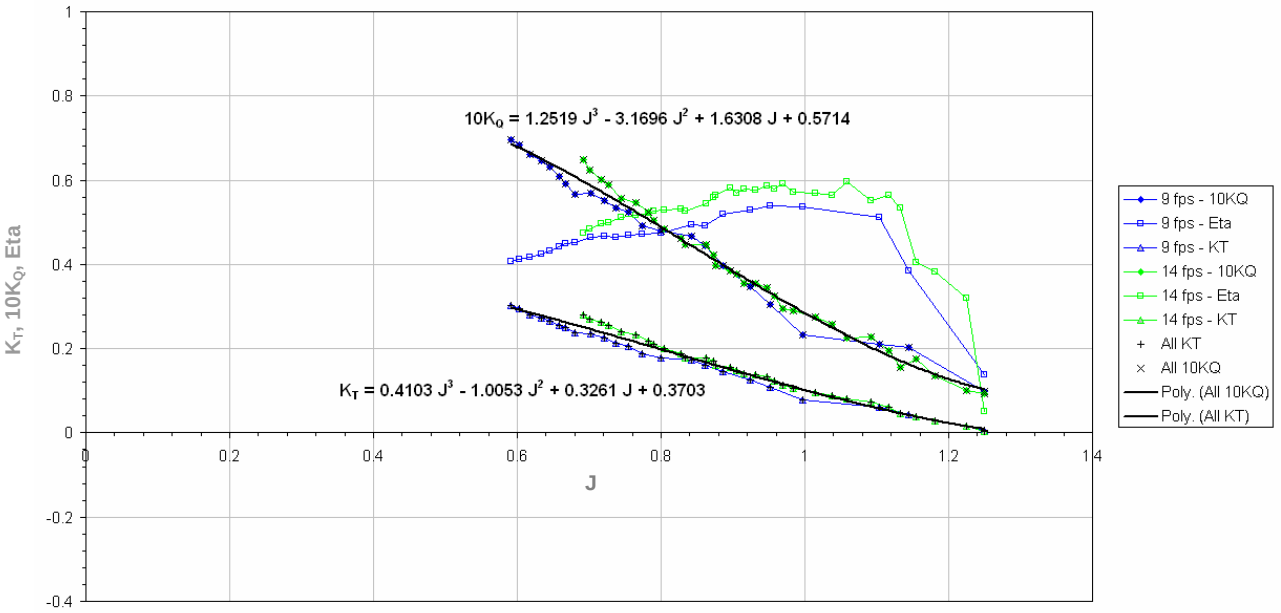

Figure B-8. The corrected propeller performance curves for propeller 1455.

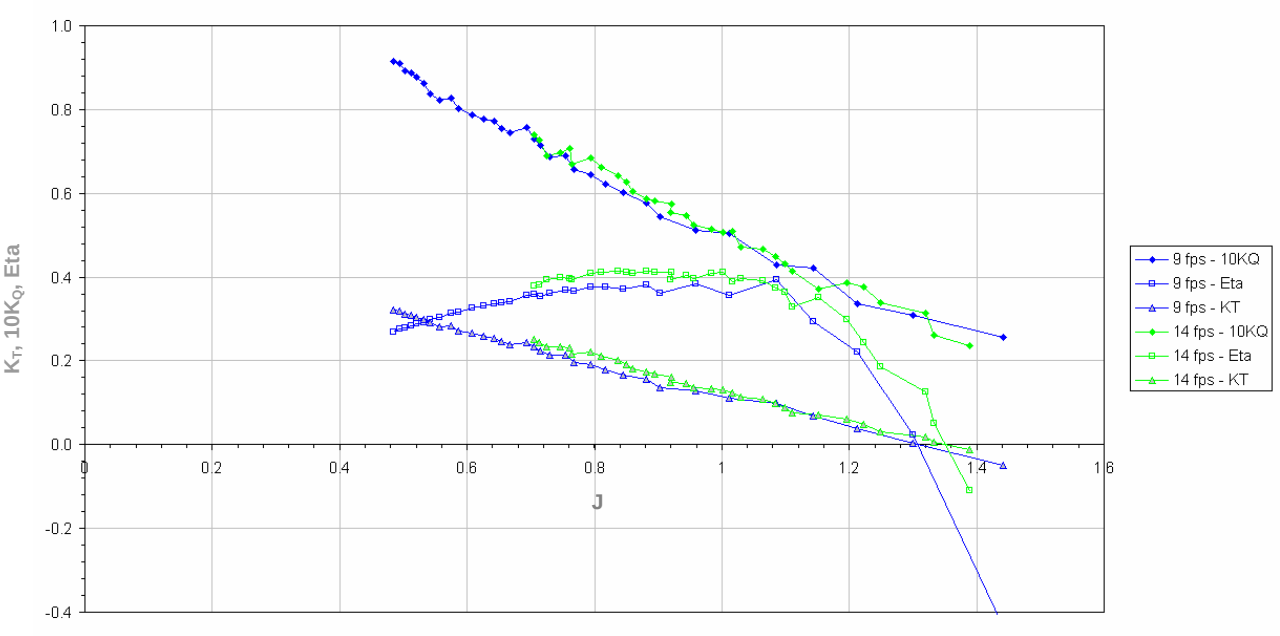

Figure B-9. The uncorrected propeller performance curves for propeller 1755.

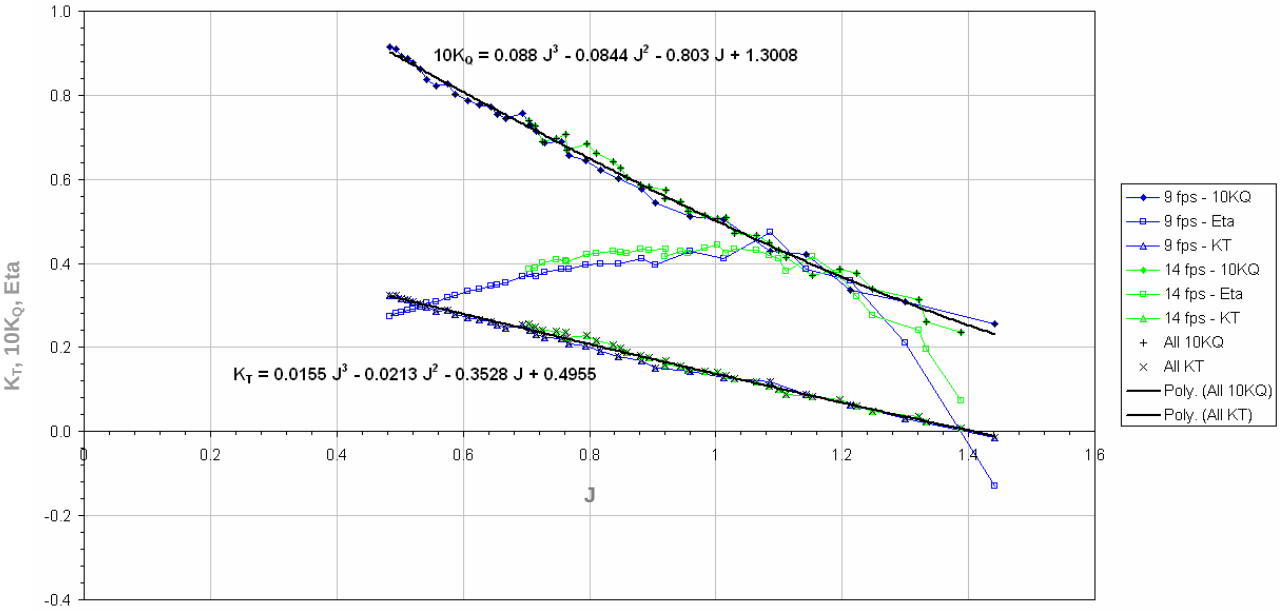

Figure B-10. The corrected propeller performance curves for propeller 1755.

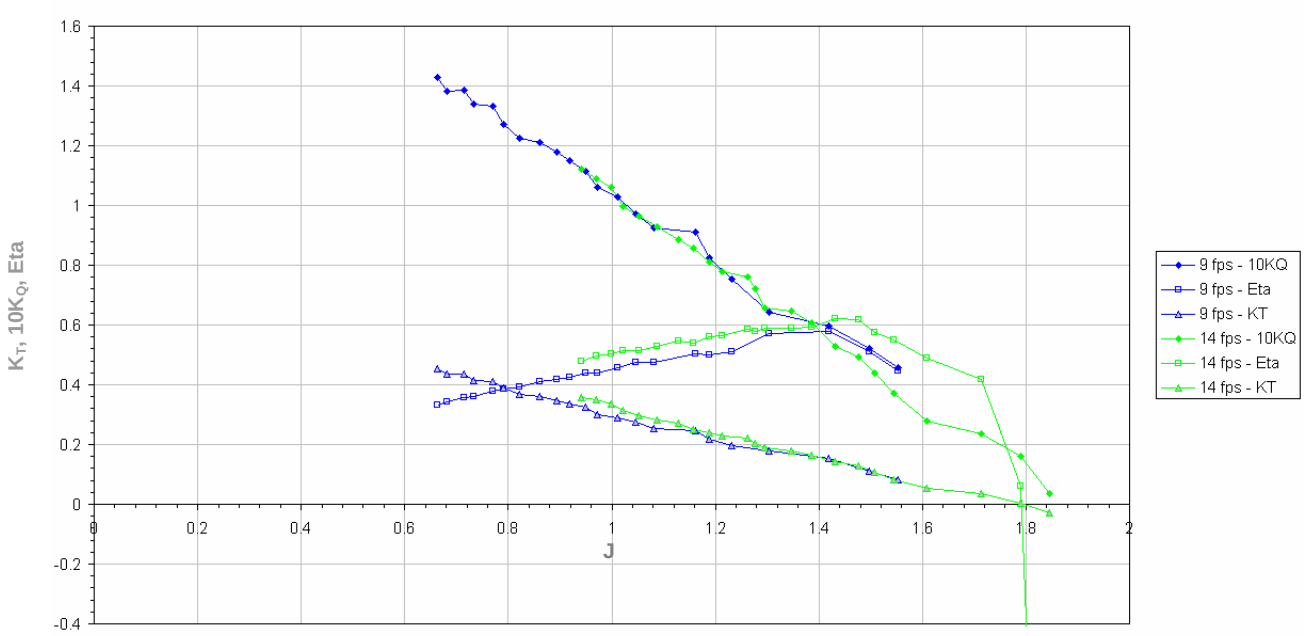

Figure B-11. The uncorrected propeller performance curves for propeller 2055.

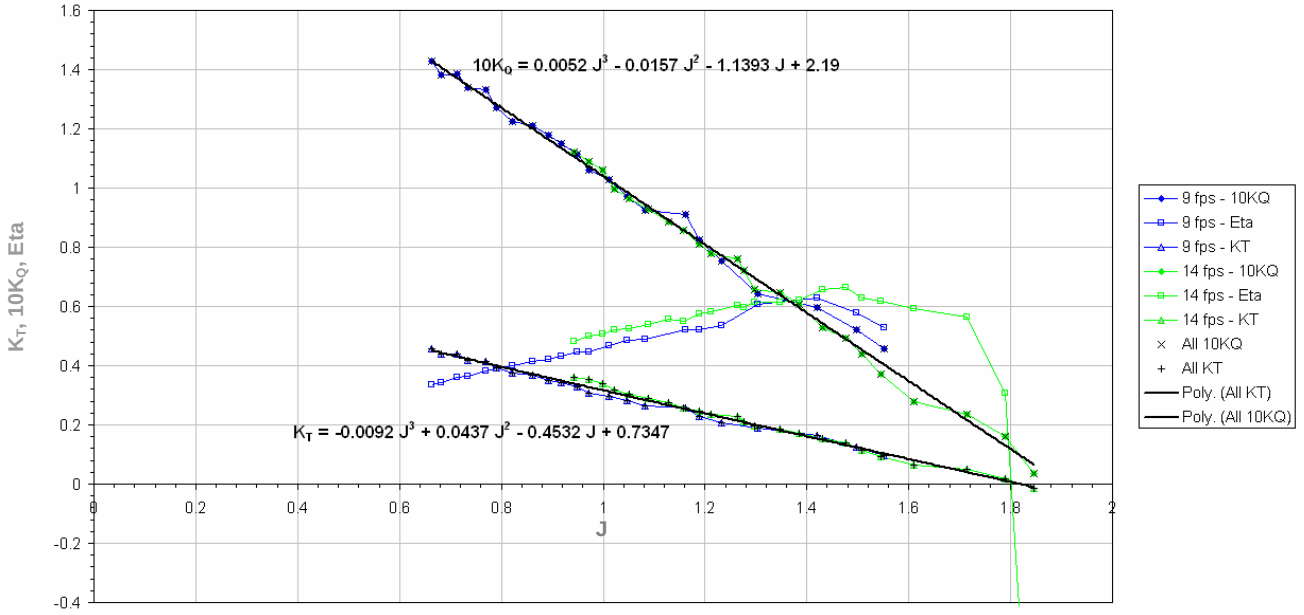

Figure B-12. The corrected propeller performance curves for propeller 2055.

Appendix C – Counter-Rotating Test Results

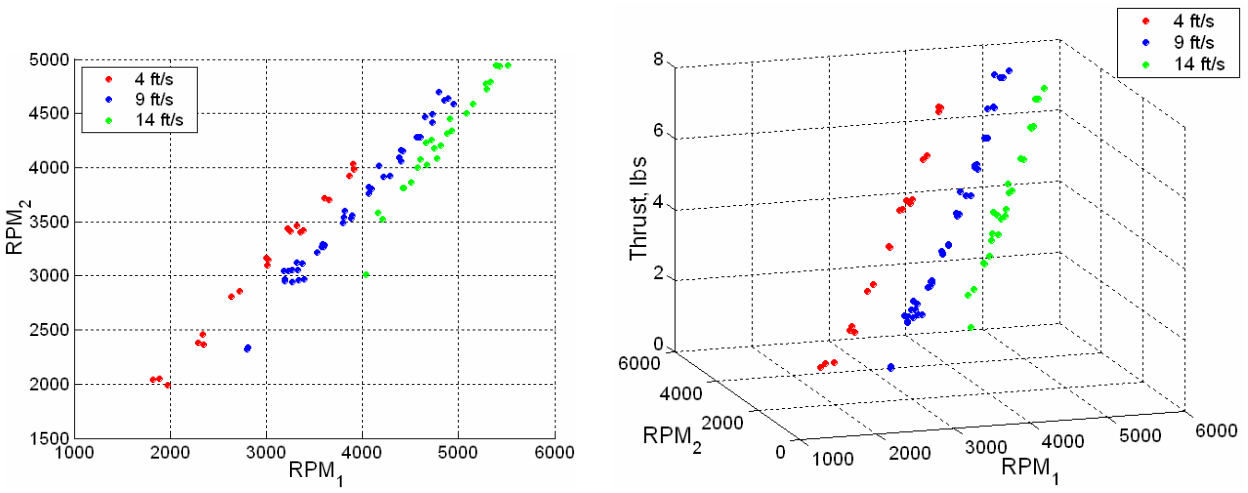

Figure C-1. Counter-rotating test results for propellers 1457R and 2.0.

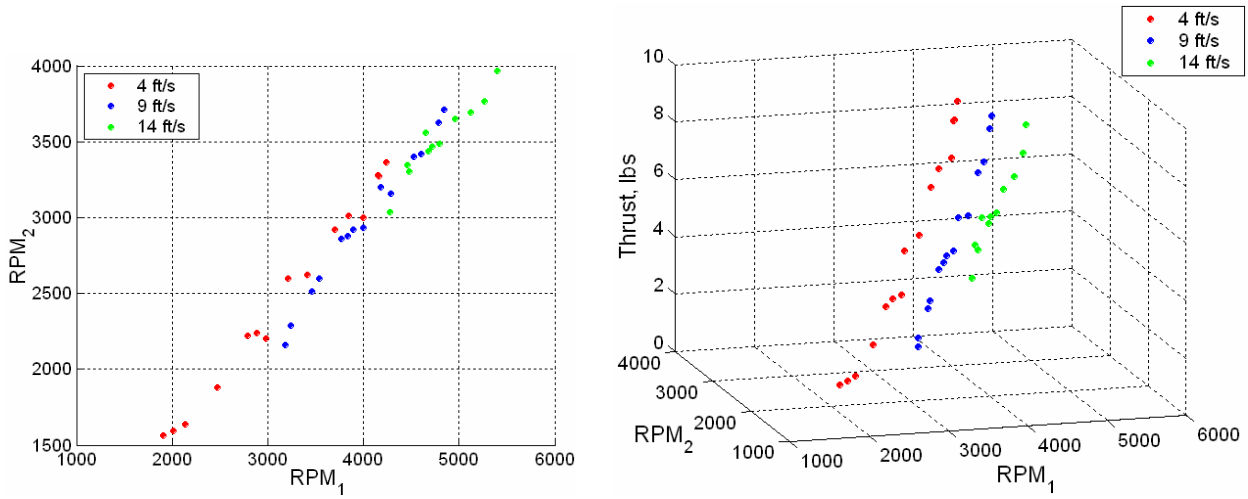

Figure C-2. Counter-rotating test results for propellers 1457R and 2.1 3.

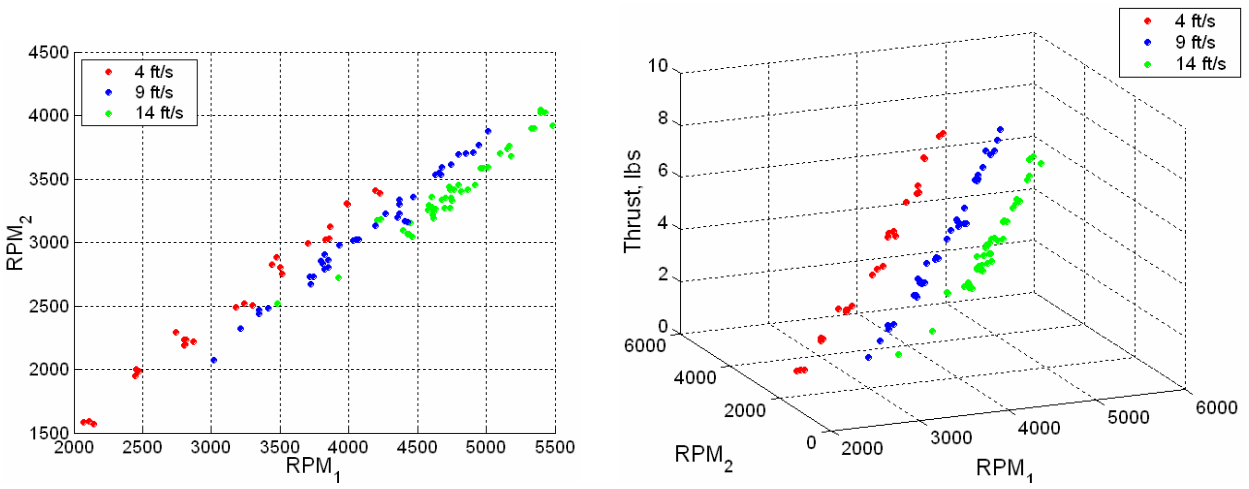

Figure C-3. Counter-rotating test results for propellers 1457R and 2055.

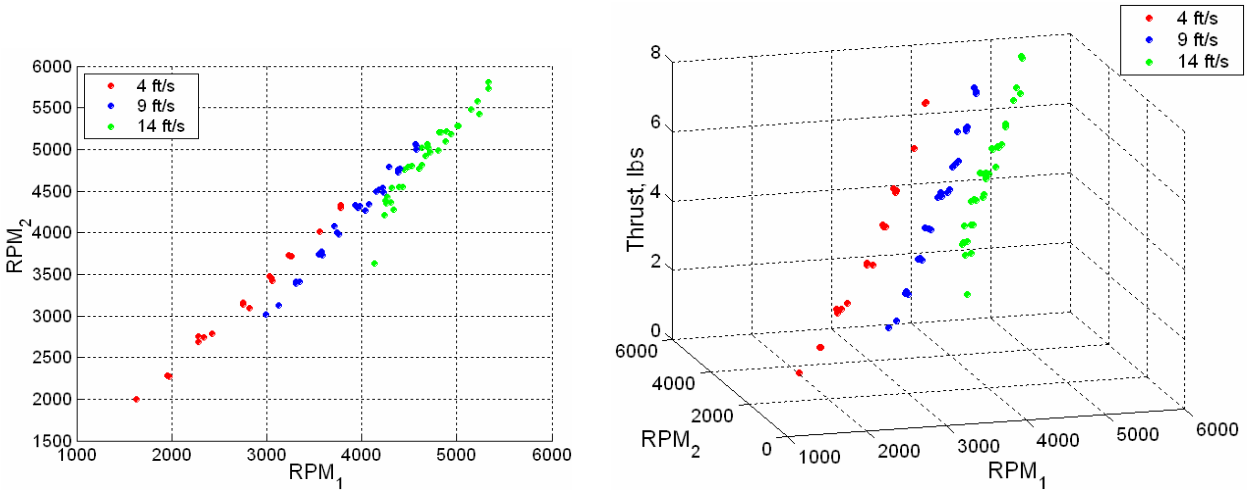

Figure C-4. Counter-rotating test results for propellers 1457R and Z55.

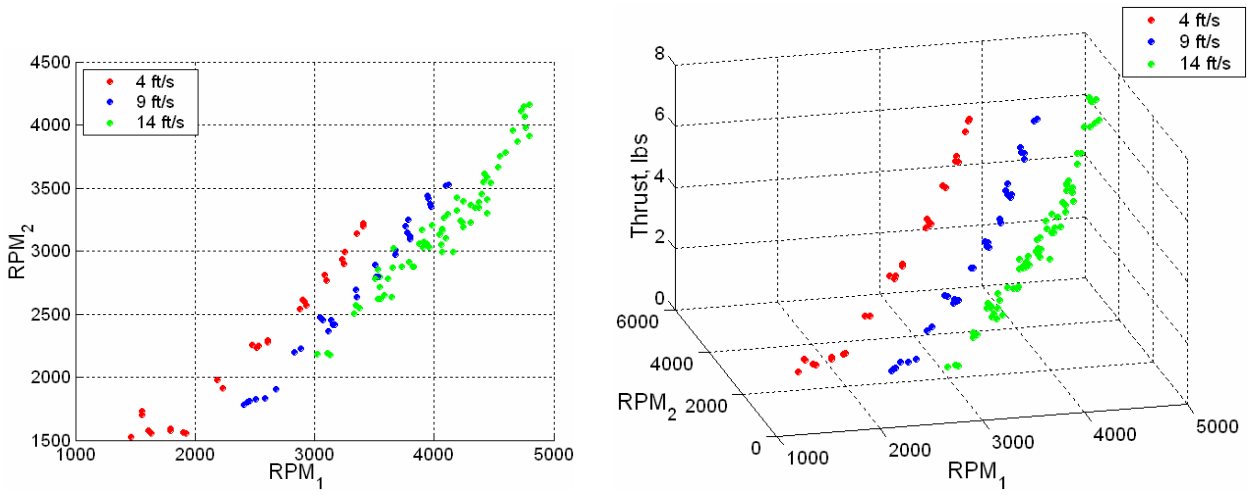

Figure C-5. Counter-rotating test results for propellers 1462R and 2.1/3.

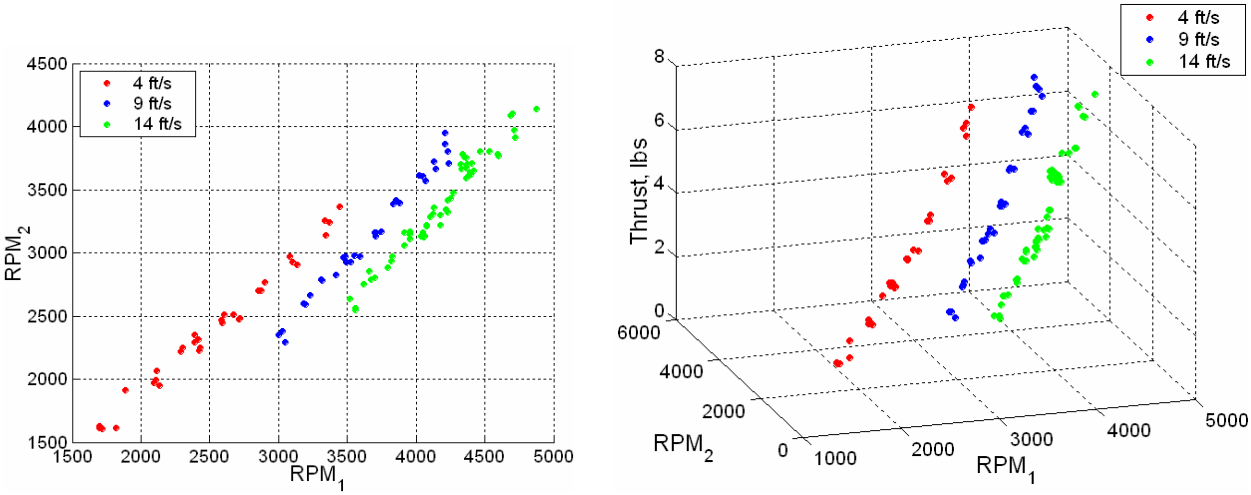

Figure C-6. Counter-rotating test results for propellers 1462R and 2055.

Appendix D – Propeller Interaction Models

The data files needed to run the interaction models and ErrorSurface programs. Each data file corresponds to one of the eight counter-rotating pairs.

```
1457R & 2.0
D1 & D2, respectively
0.186667
0.166667
Number of Points
22        4 ft/s
43        9 ft/s
25        14 ft/s
Single Prop Data
-0.7475   1457R KT coeffs
1.9283
-2.0641
0.9681
-0.9236   1457R 10KQ coeffs
2.4553
-3.0937
1.7767
0.0328    2.0 KT coeffs
-0.0108
-0.5155
0.7327
0.2391    2.0 10KQ coeffs
-0.6129
-0.7371
1.8599
CR data
                                      Velocity
Voltage   Amperage  RPM 1     RPM 2    (ft/s)     Thrust    RM
12.02     2.7       1821.131  2042.595 3.756      1.063495  -0.054
12.01     2.9       1886.292  2046.932 3.978      1.164212  -0.014
12.01     3         1977.116  1989.017 3.91       1.196239  -0.019
11.96     4.8       2290.377  2384.17  4.041      1.903954  -0.028
11.95     5.2       2334.133  2459.224 3.966      1.959506  -0.034
11.96     4.8       2346.068  2364.64  3.861      1.852419  -0.077
11.89     7.3       2639.796  2805.074 3.904      2.766892  -0.068
11.87     7.9       2720.927  2854.248 3.826      2.928426  0.002
11.81     10.7      3004.558  3163.594 3.7        3.823402  0.001
11.82     10.4      3009.348  3092.833 3.93       3.8134    -0.012
11.82     10.4      3023.63   3149.086 3.87       3.808934  -0.025
```

11.76	13.6	3217.909	3433.21	3.864	4.69559	-0.095
11.76	13.6	3252.064	3414.819	3.99	4.70892	-0.128
11.74	14.2	3321.623	3465.972	3.915	4.928529	0.013
11.74	14.2	3356.622	3406.426	4.023	4.877879	-0.077
11.73	14.6	3385.529	3418.507	3.942	4.9821	0.006
11.67	18.4	3605.067	3715.989	3.916	5.936587	0.012
11.66	19	3654.286	3698.865	3.983	6.045507	-0.016
11.67	18.4	3657.648	3705.288	3.952	6.035685	0.032
11.59	23.5	3867.464	3925.412	3.832	7.163766	-0.02
11.58	24.2	3899.439	4031.361	3.888	7.240968	-0.002
11.59	23.5	3910.943	3984.445	3.961	7.219212	-0.055
12.29	3.5	2803.93	2325.362	9.878	0.814737	0.003
12.29	3.5	2807.484	2339.539	9.824	0.772857	-0.06
12.12	6.8	3181.764	3042.268	9.844	1.89677	-0.14
12.1	6.5	3190.502	2951.175	9.729	1.776098	-0.101
12.1	6.5	3192.026	2965.003	9.787	1.722482	-0.098
12.12	6.8	3227.056	3043.652	9.881	1.856176	-0.092
12.14	7.1	3272.238	3054.304	9.811	2.034966	-0.067
12.08	6.8	3272.405	2940.32	9.802	1.858659	0.011
12.18	7.3	3315.909	3118.324	9.761	2.256718	-0.059
12.14	7.1	3323.464	3057.385	9.806	2.02824	-0.1
12.08	6.8	3340.076	2961.607	9.823	1.922711	0.042
12.18	7.3	3370.502	3115.27	9.831	2.162876	-0.07
12.06	6.8	3394.432	2972.847	9.847	1.914208	0
12	9.1	3533.26	3217.034	9.791	2.550062	-0.066
12	9.1	3573.758	3270.429	9.888	2.550201	-0.107
11.94	9.6	3587.66	3295.468	9.879	2.679884	0.005
11.96	9.5	3597.863	3263.99	9.841	2.650333	-0.052
11.96	9.5	3600.839	3285.664	9.842	2.702479	0.063
11.88	12.2	3793.4	3492.232	9.783	3.337902	-0.048
11.88	12.2	3810.719	3539.616	9.825	3.356002	-0.065
11.88	12.2	3815.013	3599.255	9.843	3.370625	-0.026
11.86	12.7	3880.035	3529.844	9.859	3.54896	-0.012
11.84	12.9	3895.911	3557.055	9.868	3.579275	-0.003
11.8	15.6	4066.907	3821.416	9.804	4.316949	0.065
11.8	15.6	4067.376	3762.446	9.758	4.261284	-0.025
11.79	15.8	4095.598	3801.848	9.852	4.278938	-0.053
11.74	18.3	4167.246	4019.788	9.884	4.818615	-0.011
11.76	17.9	4216.026	3911.157	9.904	4.743547	0.045
11.76	17.9	4286.653	3920.351	9.887	4.725055	-0.031
11.71	20.8	4384.929	4092.36	9.876	5.428445	0.008
11.7	21.3	4401.135	4157.957	9.814	5.464402	0.012
11.71	20.8	4403.441	4055.258	9.847	5.406208	0.042
11.71	20.8	4421.752	4156.407	9.852	5.485938	-0.016
11.66	24.3	4560.916	4276.914	9.87	6.166567	0.01
11.66	24.3	4584.953	4279.977	9.897	6.16252	0.06
11.64	24.7	4602.938	4280.881	9.851	6.135792	-0.012
11.6	28.6	4652.998	4470.228	9.83	6.88973	-0.011
11.58	29	4724.363	4490.291	9.862	6.926398	-0.027

11.6	28.6	4728.416	4417.329	9.811	6.934966	0.061
11.52	33.4	4797.359	4701.081	9.846	7.718062	-0.09
11.54	32.6	4850.919	4625.282	9.847	7.657208	-0.04
11.54	32.6	4892.373	4636.407	9.871	7.669713	-0.066
11.5	34.9	4953.827	4583.714	9.844	7.84877	0.099
12.48	6.6	4039.424	3012.208	14.741	1.439519	0.279
12.42	9.3	4158.393	3584.036	14.788	2.082799	0.036
12.42	9.3	4212.322	3520.479	14.756	2.280797	0.02
12.35	12	4422.93	3810.064	14.761	2.84589	-0.051
12.35	12	4430.476	3815.493	14.699	2.830361	-0.096
12.31	12.8	4507.022	3862.045	14.738	3.006864	-0.045
12.27	14.6	4574.979	3997.248	14.753	3.400141	-0.013
12.27	15	4601.991	4076.486	14.723	3.552591	0.031
12.17	17.2	4659.531	4229.187	14.659	4.064663	-0.095
12.27	14.6	4673.765	4026.475	14.696	3.522708	0.011
12.17	17.2	4721.66	4254.952	14.715	3.973847	-0.1
12.2	16.9	4745.134	4177.227	14.712	3.886193	0.019
12.2	16.9	4775.602	4085.871	14.692	4.007837	-0.028
12.13	17.7	4816.307	4206.798	14.699	4.140361	-0.035
12.09	20.1	4882.826	4316.381	14.746	4.540612	0.06
12.06	20.7	4907.741	4454.886	14.718	4.750501	-0.031
12.09	20.1	4928.91	4338.963	14.666	4.598183	0.119
12	23.6	5087.382	4502.793	14.618	5.390772	0.017
11.98	23.8	5153.131	4587.877	14.76	5.333671	-0.003
11.9	27.8	5286.246	4772.739	14.703	6.144232	-0.061
11.92	27.5	5294.952	4724.995	14.743	6.139956	0.047
11.92	27.5	5336.993	4788.674	14.746	6.153612	-0.035
11.85	32.1	5394.497	4948.24	14.809	6.847403	-0.059
11.85	32.1	5431.266	4939.352	14.937	6.833605	-0.101
11.83	33.7	5512.445	4944.478	14.834	7.118892	0.141

```
1457R & 2.1/3
D1 & D2, respectively
0.186667
0.175
Number of Points
15        4 ft/s
14        9 ft/s
11        14 ft/s
Single Prop Data
-0.7475   1457R KT coeffs
1.9283
-2.0641
0.9681
-0.9236   1457R 10KQ coeffs
2.4553
-3.0937
```

1.7767
0.1152 2.1/3 KT coeffs
-0.4397
0.0464
0.6971
0.4168 2.1/3 10KQ coeffs
-1.7006
0.9717
1.6318
CR data

Voltage	Amperage	RPM 1	RPM 2	Velocity (ft/s)	Thrust	RM
11.96	3	1902.857	1563.815	4.047	1.207313	-0.123
11.94	3.1	2014.286	1595.911	4.079	1.303239	-0.061
11.93	3.4	2134.286	1636.825	4.03	1.409296	0.03
11.88	5.4	2468.571	1877.285	4.06	2.192094	-0.002
11.81	8.1	2794.286	2223.632	4.022	3.104819	-0.104
11.78	8.6	2888.571	2236.249	4.05	3.332493	-0.11
11.77	9.4	2982.857	2200.37	4.084	3.489542	-0.062
11.7	12.9	3214.286	2596.75	4.026	4.588058	-0.203
11.67	14.6	3411.429	2622.074	4.087	5.058723	-0.042
11.59	20	3702.857	2919.078	4.031	6.374356	-0.193
11.55	21.9	3848.571	3012.015	4.073	6.908877	-0.086
11.52	24.1	3994.286	2997.737	4.064	7.248335	0.017
11.44	29.4	4157.143	3281.662	4.053	8.232673	-0.092
11.43	29.4	4165.714	3275.749	4.04	8.255894	-0.097
11.39	32.2	4242.857	3365.683	4.049	8.803433	-0.279
11.94	5.6	3180	2158.833	9.816	1.701693	0.084
11.91	6.2	3240	2289.719	9.838	1.868896	0.026
11.86	8.7	3462.857	2511.776	9.839	2.618041	-0.107
11.84	9.2	3540	2597.556	9.852	2.775938	-0.046
11.79	12.1	3771.429	2861.087	9.836	3.566604	-0.146
11.77	12.8	3840	2879.731	9.857	3.742667	-0.121
11.75	13.6	3891.429	2918.286	9.858	3.932813	-0.106
11.74	14.1	3994.286	2930.594	9.857	4.088667	-0.024
11.68	17.9	4182.857	3201.136	9.86	4.949106	-0.179
11.67	18.2	4285.714	3155.484	9.866	5.013982	0.023
11.6	23.5	4525.714	3398.201	9.83	6.22373	-0.076
11.57	25	4602.857	3421.836	9.84	6.581187	0.007
11.5	29.7	4782.857	3626.783	9.837	7.47775	-0.073
11.48	31.8	4842.857	3709.708	9.857	7.844667	-0.029
12.13	10.6	4277.143	3037.525	14.813	2.964281	0.058
12.03	13.7	4457.143	3346.662	14.85	3.79141	-0.14
12	13.6	4482.857	3305.127	14.898	3.642987	-0.059
11.93	17	4654.286	3561.609	14.936	4.458384	-0.156
11.91	16.8	4680	3436.036	14.927	4.395393	-0.11
11.88	17.6	4714.286	3470.283	14.908	4.575194	-0.093
11.84	18.2	4800	3483.481	14.913	4.711299	-0.035
11.78	21.3	4962.857	3650.932	14.949	5.304262	-0.142

11.72	23.1	5125.714	3694.377	14.927	5.698393	0.058
11.65	26.9	5262.857	3766.196	14.949	6.408262	0.064
11.56	31.3	5400	3964.293	14.952	7.170926	-0.01

1457R & 2055
D1 & D2, respectively
0.186667
0.180833
Number of Points
27
37
43
Single Prop Data
-0.7475 1457R KT coeffs
1.9283
-2.0641
0.9681
-0.9236 1457R 10KQ coeffs
2.4553
-3.0937
1.7767
-0.0092 2055 KT coeffs
0.0437
-0.4532
0.7347
0.0052 2055 10KQ coeffs
-0.0157
-1.1393
2.19
CR data

Voltage	Amperage	RPM 1	RPM 2	Velocity (ft/s)	Thrust	RM
12.3	3.4	2110.162	1589.831	4.146135	1.28633	0.064
12.2	5.7	2455.404	1998.371	4.129212	2.136292	-0.043
12.09	8.5	2870.73	2222.281	4.246545	3.052571	-0.028
11.97	12.5	3242.02	2516.377	4.034443	4.161562	0.001
11.78	23.3	3866.265	3122.224	3.642958	6.737299	-0.086
12.33	3.4	2069.36	1580.914	4.253314	1.263998	0.005
12.33	3.4	2145.477	1566.935	4.183366	1.282627	0.032
12.24	5.5	2444.37	1950.169	4.064905	2.046389	-0.045
12.24	5.5	2472.946	1982.693	4.35598	2.110545	-0.034
12.2	5.7	2804.442	2232.617	3.9927	2.91408	-0.067
12.2	5.7	2739.019	2289.228	3.814444	2.961772	-0.135
12.12	7.9	2814.47	2235.035	4.046853	2.863305	-0.044
12.12	7.9	2808.702	2197.058	4.401108	2.954472	-0.077
12.12	7.9	2806.008	2189.385	4.228494	2.943438	-0.082
12.03	11.6	3180.745	2491.202	4.437211	3.964835	-0.146
12	12.8	3297.543	2503.618	3.80429	4.237199	0.073

11.9	17.1	3444.177	2828.042	4.17434	5.108068	-0.205
11.9	17.1	3477.353	2885.176	3.752393	5.205294	-0.179
11.89	16.9	3517.119	2752.19	3.834752	5.174922	-0.078
11.89	16.9	3502.098	2804.24	3.974649	5.326015	-0.097
11.82	21.2	3706.022	2992.218	3.488394	6.225135	-0.126
11.8	22.5	3832.954	3020.726	3.976905	6.495147	-0.047
11.8	22.5	3855.974	3030.573	4.381929	6.539224	-0.116
11.72	28.6	3994.12	3297.008	3.508702	7.601188	-0.129
11.72	28.6	3987.084	3310.022	4.090853	7.629957	-0.212
11.68	32.5	4195.675	3405.602	4.202545	8.322818	-0.151
11.68	32.5	4226.301	3388.577	4.211571	8.43638	-0.191
12.2	7.4	3415.206	2483.208	9.450056	1.977471	-0.048
12.13	10.5	3717.698	2729.812	9.74418	2.823288	-0.028
12.09	12	3854.033	2864.956	9.51671	3.165835	-0.053
12.04	14.9	4049.418	3023.315	9.87643	3.953507	-0.027
11.94	22.3	4466.012	3355.161	9.703976	5.481496	0.042
11.86	27.7	4743.045	3614.553	9.757934	6.787274	-0.042
11.77	34.8	5011.165	3872.524	9.837284	7.975791	-0.081
12.32	4.3	3017.741	2074.201	10.03725	1.137228	-0.028
12.27	5.8	3212.969	2319.514	9.815066	1.513557	-0.074
12.22	7	3350.045	2439.572	9.758992	1.859427	-0.001
12.22	7	3349.098	2463.57	9.835168	1.973483	-0.054
12.13	10.3	3725.403	2669.177	9.793906	2.839484	-0.035
12.13	10.3	3745.997	2732.388	9.845748	2.735025	-0.03
12.1	12.3	3823.205	2906.258	9.67012	3.282638	-0.118
12.1	12.3	3795.366	2851.077	9.81295	3.36425	-0.097
12.09	11.9	3825.813	2790.977	9.859502	3.205033	-0.036
12.09	11.9	3811.747	2835.279	9.750528	3.188204	-0.037
12.09	11.9	3851.76	2803.782	9.662714	3.197577	-0.03
12.06	14	3929.343	2975.664	10.08274	3.818008	-0.119
12.05	14.9	4032.767	3016.061	9.7336	3.885761	-0.037
12.05	14.9	4072.756	3019.863	9.621452	3.932683	-0.002
12.01	17.5	4195.163	3127.791	9.947316	4.538916	-0.126
11.99	18.7	4270.293	3225.676	9.85527	4.797415	-0.077
11.99	18.7	4351.412	3196.071	10.08168	4.92385	-0.077
11.98	19.5	4407.071	3169.2	9.761108	5.017733	0.049
11.98	19.5	4428.817	3163.345	9.80237	5.023712	0.106
11.98	19.5	4370.796	3226.059	9.672236	5.021941	0.079
11.96	20.5	4365.117	3296.023	9.450056	5.042471	-0.079
11.96	20.5	4369.066	3332.467	9.771688	5.113264	-0.177
11.9	25.6	4667.243	3533.606	9.838342	6.405946	-0.007
11.9	25.6	4656.393	3546.957	9.96107	6.344944	-0.065
11.9	25.6	4633.335	3533.287	9.863734	6.399651	-0.035
11.88	26.1	4680.377	3592.05	9.930388	6.520424	-0.105
11.82	30.4	4850.272	3700.586	9.714556	7.170018	-0.041
11.82	30.4	4903.205	3706.04	9.826704	7.30325	-0.025
11.81	31	4795.554	3696.227	9.764282	7.335192	-0.114
11.8	32.3	4946.232	3767.338	9.890184	7.646521	-0.116
12.19	12.1	4392.049	3094.286	14.6836	2.663009	-0.03

12.08	14.2	4618.953	3186.736	14.58815	3.116313	-0.019
12.02	15.6	4737.826	3266.497	14.8241	3.395716	0.005
11.97	16.6	4678.87	3338.587	14.65571	3.585949	-0.059
11.91	17.5	4701.42	3349.008	14.64713	3.887087	0.059
11.87	19	4921.87	3453.008	14.6954	4.053576	-0.033
11.82	22.7	4979.248	3586.486	14.62461	4.674203	-0.051
11.78	26	5167.485	3761.582	14.68682	5.296709	0.027
11.71	31.1	5481.955	3918.131	14.5549	6.446136	0.004
12.48	4.4	3483.382	2518.784	14.84662	0.760666	-0.06
12.39	6.7	3925.934	2725.915	14.6096	1.374952	-0.1
12.3	10.5	4232.529	3184.424	15.09758	2.394329	-0.31
12.3	10.5	4209.643	3175.432	14.93778	2.458778	-0.267
12.25	11.8	4449.048	3151.099	15.15014	2.739103	-0.156
12.22	11.8	4431.941	3064.66	14.7458	2.750568	-0.05
12.22	11.8	4462.811	3047.314	14.5002	2.592365	-0.035
12.22	11.8	4429.734	3068.987	14.77691	2.61537	0.044
12.14	13.9	4586.116	3295.456	14.74366	3.1271	-0.079
12.14	13.9	4614.677	3241.651	14.65035	3.150785	-0.07
12.11	14.6	4578.528	3258.125	14.82946	3.189894	-0.099
12.11	14.6	4608.341	3219.215	14.71363	3.143548	-0.127
12.1	14.3	4604.613	3280.805	15.18767	3.254539	-0.078
12.05	15.4	4698.3	3272.843	14.32967	3.33095	-0.055
12.05	15.4	4628.292	3260.141	14.81552	3.297833	-0.012
11.99	16.6	4604.898	3360.165	14.82302	3.561481	-0.109
11.99	16.6	4749.29	3329.983	14.92169	3.63322	0.03
11.93	17.9	4736.417	3413.218	14.98068	3.938285	-0.088
11.93	17.9	4732.879	3438.859	14.97425	3.914857	-0.06
11.91	18.7	4765.231	3413.804	14.50664	4.001747	-0.041
11.91	18.7	4795.075	3449.135	15.05683	4.107227	-0.119
11.91	17.6	4745.295	3352.876	14.29214	3.626992	0.03
11.88	18.6	4862.147	3413.208	14.63963	4.037458	0.067
11.88	18.6	4815.545	3403.856	14.941	4.15749	-0.066
11.83	22.2	5011.084	3588.515	14.57313	4.60007	-0.039
11.83	22.2	4959.04	3581.793	14.56241	4.619755	0.036
11.78	26	5154.817	3740.481	14.46695	5.111231	-0.121
11.78	25.5	5101.392	3702.81	14.86592	5.025915	0.043
11.78	25.5	5178.41	3676.98	14.9528	5.262102	-0.039
11.74	29	5330.51	3900.985	14.49806	5.879904	-0.022
11.74	29	5350.607	3899.633	14.5978	6.0444	0.037
11.7	32.3	5430.789	4019.284	14.88845	6.670879	-0.008
11.7	32.3	5395.393	4023.608	14.65035	6.605785	-0.094
11.7	32.3	5392.842	4046.367	14.53023	6.572822	-0.136

1457R & Z55
D1 & D2, respectively
0.186667
0.180833

Number of Points
19
27
33
Single Prop Data
-0.7475 1457R KT coeffs
1.9283
-2.0641
0.9681
-0.9236 1457R 10KQ coeffs
2.4553
-3.0937
1.7767
0.1486 Z55 KT coeffs
-0.2762
-0.2836
0.5001
0.4171 Z55 10KQ coeffs
-0.9068
-0.2116
0.9436
CR data

Voltage	Amperage	RPM 1	RPM 2	Velocity (ft/s)	Thrust	RM
12.07	3.5	1969.894	2271.641	4.059	1.378034	-0.043
12	6	2427.045	2781.641	3.94	2.334983	-0.046
11.93	8.8	2755.393	3160.722	4.031	3.246356	0.002
11.87	11.8	3034.512	3480.588	3.983	4.173507	-0.006
11.82	15.1	3234.094	3732.14	3.923	5.067993	0.016
12.12	2.4	1630.928	1995.38	3.766	0.824052	-0.016
12.09	3.6	1956.43	2287.486	4.007	1.369927	-0.049
12.03	5.5	2281.135	2691.715	3.961	2.104212	-0.022
12.03	5.5	2336.748	2742.18	4.048	2.181373	-0.079
12.03	5.5	2276.788	2757.811	4.047	2.184313	-0.076
11.95	8.4	2819.451	3093.629	4.058	3.208974	-0.064
11.95	8.4	2750.426	3138.654	3.88	3.196508	0.011
11.88	11.4	3046.718	3455.495	3.946	4.118334	-0.024
11.88	11.4	3054.926	3420.64	3.966	4.130506	-0.002
11.83	14.6	3255.136	3720.927	3.956	4.954919	-0.023
11.83	14.6	3259.36	3716.62	3.888	5.008968	-0.012
11.77	18.5	3560.439	4009.066	4.036	6.035655	0.006
11.7	23.2	3773.968	4328.856	4.01	7.156105	0.091
11.7	23.2	3774.398	4302.41	4.06	7.182094	-0.014
12.23	7.3	3308.297	3389.499	9.755	2.22785	-0.045
12.13	9.9	3572.433	3771.425	9.767	2.978585	0.012
12.02	12.5	3712.726	4078.061	9.816	3.698693	-0.051
11.88	16	3965.267	4301.139	9.787	4.569482	0.007
11.8	18.9	4179.504	4517.278	9.803	5.227804	-0.07
11.74	22.4	4285.828	4793.767	9.786	6.030337	-0.075

11.67	27.6	4564.291	5060.573	9.777	7.140033	-0.023
12.38	4.6	2995.154	3011.674	9.666	1.428048	0.029
12.32	5.6	3124.413	3127.207	9.684	1.538627	0.08
12.28	7.2	3303.546	3415.029	9.738	2.175396	0.03
12.28	7.2	3342.206	3413.716	9.809	2.147675	0.016
12.16	9.7	3545.65	3737.5	9.789	2.976772	-0.014
12.16	9.7	3587.99	3724.855	9.83	2.93173	0.015
12.05	12.4	3757.808	3978.796	9.833	3.663167	-0.047
12.05	12.4	3744.31	3998.803	9.844	3.70177	-0.02
11.95	15.2	3933.222	4332.022	9.758	4.414284	-0.048
11.95	15.2	3978.855	4320.275	9.659	4.431046	-0.087
11.91	15.9	4073.338	4337.834	9.779	4.619323	0.022
11.91	15.9	4035.637	4267.442	9.725	4.578522	0.033
11.81	19.2	4221.716	4539.372	9.757	5.291139	0.055
11.81	19.2	4231.146	4487.997	9.797	5.322933	0.064
11.8	18.9	4155.583	4498.397	9.795	5.168643	0.016
1174	22.9	4385.78	4726.012	9.753	6.084561	0.019
1174	22.9	4383.148	4756.554	9.776	6.137888	0.065
11.73	23.1	4406.893	4773.496	9.833	6.170167	0.109
11.68	27.4	4576.663	5046.439	9.711	7.028506	0.006
11.68	27.4	4570.727	4999.143	9.804	6.994949	-0.007
12.33	11.5	4303.128	4360.788	14.483	3.063672	0.051
12.29	13.3	4426.338	4553.709	14.634	3.450238	-0.058
12.2	16.1	4628.747	4817.455	14.697	4.145925	0.005
12.14	18.7	4808.576	4982.442	14.725	4.831027	0.03
12.09	21	4817.087	5204.758	14.823	5.264476	-0.022
12.03	23.8	5014.424	5287.103	14.869	5.897593	-0.017
11.95	27.7	5218.843	5582.285	14.786	6.789361	-0.034
11.87	32	5332.718	5803.557	14.87	7.556813	-0.016
12.49	7.4	4135.785	3626.22	14.699	1.901361	0.284
12.39	10.3	4242.027	4211.643	14.736	2.764428	0.175
12.39	10.3	4337.858	4270.349	14.716	2.756065	0.113
12.36	11.3	4252.654	4383.201	14.69	2.978402	0.007
12.36	11.3	4254.751	4354.227	14.671	3.05527	-0.013
12.36	11.3	4263.34	4433.631	14.585	2.962634	-0.053
12.3	13.2	4313.454	4542.408	14.571	3.42261	-0.003
12.3	13.2	4396.793	4552.944	14.68	3.443226	0.059
12.23	15.5	4454.262	4751.839	14.563	4.030884	-0.037
12.23	15.5	4492.502	4788.637	14.627	4.057721	-0.083
12.23	15.5	4523.87	4796.178	14.668	4.027618	-0.069
12.21	15.9	4606.383	4772.653	14.75	4.102485	0.029
12.17	17.8	4671.073	4922.281	14.701	4.549796	0.106
12.16	18.3	4686.837	5067.261	14.838	4.620771	-0.104
12.16	18.3	4630.739	5022.022	14.806	4.680745	-0.033
12.16	18.3	4704.174	5017.608	14.775	4.660953	-0.016
12.16	18.3	4716.656	4962.303	14.825	4.662915	-0.085
12.1	21.3	4941.47	5184.573	14.8	5.387429	0.122
12.1	21.3	4882.382	5098.14	14.732	5.336555	0.051
12.09	21	4837.147	5211.513	14.723	5.230591	-0.06

12.09	21	4888.885	5212.076	14.738	5.316864	0.043
12.05	23.3	5011.296	5278.443	14.798	5.825991	-0.121
11.98	26.6	5149.59	5483.203	14.84	6.463211	0.161
11.98	26.6	5233.403	5425.073	14.793	6.695895	0.168
11.89	31.4	5330.025	5731.348	14.758	7.549234	-0.016

1462R & 2.0
D1 & D2, respectively
0.203333
0.166667
Number of Points
40
32
54
Single Prop Data
-0.0475 1462R KT coeffs
0.2747
-0.834
0.6734
0.1821 1462R 10KQ coeffs
-0.3066
-0.7631
1.0905
0.0328 2.0 KT coeffs
-0.0108
-0.5155
0.7327
0.2391 2.0 10KQ coeffs
-0.6129
-0.7371
1.8599
CR data

Voltage	Amperage	RPM 1	RPM 2	Velocity (ft/s)	Thrust	RM
12.42	1.7	1538.034	1786.531	3.932	0.782517	-0.018
12.37	2.8	1949.354	2145.34	4.102	1.443633	0.005
12.3	4.7	2265.308	2657.723	3.969	2.415682	0.002
12.24	7.1	2544.109	3043.482	3.999	3.458453	-0.033
12.18	9.7	2755.795	3341.933	3.97	4.447741	-0.04
12.14	12	3031.652	3472.478	3.991	5.201979	0.022
12.05	17	3244.35	3944.568	3.945	6.799275	-0.062
11.98	20.7	3477.442	4148.371	4.044	7.817134	-0.013
12.42	1.6	1540.51	1772.896	4.095	0.777208	-0.055
12.42	1.6	1509.266	1780.94	3.964	0.770388	0.013
12.38	3	1922.118	2232.48	4.132	1.524463	-0.024
12.38	3	1909.424	2250.088	3.889	1.515026	-0.048
12.38	3	1878.52	2308.959	4.166	1.597553	-0.079
12.37	2.8	1913.039	2152.47	4.07	1.438696	-0.031

12.37	2.8	1926.031	2152.705	3.875	1.424221	-0.006
12.33	4.5	2244.543	2553.923	4.005	2.285808	0.011
12.33	4.5	2211.182	2590.279	3.755	2.291439	0.093
12.31	4.9	2258.113	2691.836	3.949	2.537509	-0.118
12.31	4.9	2261.889	2785.732	3.929	2.539342	-0.115
12.26	7	2518.166	2967.311	4.1	3.317512	-0.048
12.26	7	2509.957	2949.416	4.062	3.319214	-0.075
12.26	7	2578.405	2986.93	4.066	3.328455	-0.073
12.24	7.4	2560.121	3099.16	3.883	3.49368	-0.01
12.23	7.5	2539.96	3034.227	4.067	3.548515	-0.108
12.19	9.7	2799.463	3317.928	3.957	4.360978	0.032
12.19	9.7	2793.538	3353.405	4.152	4.38569	0.003
12.14	11.8	2998.833	3507.455	4.046	5.153253	-0.002
12.1	14.6	3218.873	3648.052	3.994	6.085157	0.068
12.1	14.6	3217.715	3685	4.141	6.105014	0.025
12.08	15.3	3199.594	3769.38	3.945	6.335275	0.113
12.08	15.3	3247.148	3691.222	4.032	6.232416	0.148
12.08	14.8	3216.245	3638.914	4.029	6.075237	0.294
12.08	14.8	3210.78	3572.646	3.98	6.08133	0.245
12.07	16	3284.843	3764.228	4.011	6.522165	0.125
12.05	17	3248.541	3889.08	4.061	6.747154	-0.004
12	20.2	3394.467	4131.238	3.911	7.746297	0.01
12	20.2	3450.162	4136.519	4.06	7.902094	-0.172
11.93	24.6	3635.132	4329.813	4.085	8.657602	0.09
11.93	24.6	3589.968	4299.812	4.075	8.738998	0.098
11.93	24.5	3658.552	4150.73	4.059	8.707034	0.265
12.46	1.4	2370.691	1900.218	9.511	0.157031	0.029
12.33	4.5	3030.974	2909.759	9.625	1.736189	0.008
12.27	6.6	3254.647	3267.964	9.711	2.601506	-0.03
12.21	8.9	3421.846	3643.007	9.629	3.53876	-0.004
12.17	10.6	3608.199	3797.861	9.757	4.175139	0.009
12.14	12.3	3753.158	3914.238	9.693	4.830919	0.006
12.08	15.5	3870.421	4210.155	9.779	5.851323	-0.086
12.01	19.2	4120.865	4437.151	9.785	6.911192	0.021
12.49	1.4	2195.468	1885.7	9.366	0.049757	-0.059
12.49	1.4	2277.53	1930.125	9.619	0.136334	-0.036
12.47	1.4	2345.082	1906.679	9.697	0.231494	0.071
12.47	1.4	2350.123	1876.303	9.51	0.23289	0.004
12.42	2.7	2682.209	2426.592	9.724	0.897378	0.037
12.35	4.5	3017.183	2860.443	9.75	1.723128	0.029
12.35	4.5	2976.288	2893.488	9.792	1.728207	-0.007
12.28	6.4	3221.421	3216.222	9.42	2.498271	0.006
12.23	8.5	3465.72	3627.126	9.746	3.45855	-0.034
12.23	8.5	3444.146	3631.951	9.473	3.386687	-0.014
12.23	8.5	3448.559	3658.99	9.717	3.55437	0.016
12.22	8.5	3410.515	3541.341	9.433	3.338086	0.075
12.21	8.9	3484.645	3651.36	9.769	3.509875	-0.088
12.21	8.9	3533.817	3635.519	9.724	3.499378	-0.057
12.21	8.9	3439.806	3614.238	9.777	3.489033	-0.058

12.18	10.6	3610.09	3814.133	9.474	4.094828	-0.004
12.14	12	3735.382	3968.969	9.553	4.717961	-0.021
12.14	12	3795.655	3984.592	9.811	4.835966	0.008
12.1	14.4	3913.624	4127.238	9.782	5.623757	0.062
12.1	14.4	3857.002	4180.479	9.849	5.6725	0.082
12.08	15.4	3920.16	4230.301	9.857	5.837667	-0.096
12.08	15.4	3903.856	4291.491	9.718	5.832514	-0.031
12.02	18.7	4125.946	4445.475	9.698	6.819637	0.031
12.02	18.7	4108.566	4439.128	9.775	6.931743	0.015
12.47	3.2	3180.355	2598.35	14.633	0.203022	0.032
12.45	4.3	3339.101	2905.628	14.443	0.611102	-0.039
12.39	5.8	3501.217	3112.178	14.604	1.008741	-0.008
12.35	7.4	3657.119	3416.107	14.659	1.563663	-0.026
12.32	8.9	3909.957	3560.963	14.571	1.85861	-0.01
12.28	10.7	3974.075	3771.688	14.71	2.395757	-0.008
12.25	12.8	4128.845	3915.803	14.708	2.860321	-0.037
12.21	14.5	4236.168	4095.384	14.662	3.382314	-0.001
12.18	16.9	4294.425	4204.613	14.601	3.897092	0.044
12.11	21.7	4461.646	4496.102	14.602	5.035309	-0.005
12.04	27.3	4717.189	4729.97	14.662	6.027314	0.03
0	0	2888.504	2544.403	14.554	-0.12706	-0.269
12.52	2.4	2967.56	2547.159	14.236	-0.06887	-0.207
12.49	3.2	3257.647	2477.737	14.546	0.185218	0.044
12.49	3.2	3221.345	2525.059	14.605	0.221958	0.084
12.48	3	3090.294	2578.889	14.587	0.218066	-0.059
12.48	3	3147.114	2622.119	14.555	0.242158	0.03
12.45	4.3	3336.406	2836.524	14.229	0.610654	0.039
12.45	4.3	3374.807	2836.022	14.278	0.626001	0.012
12.4	5.7	3572.922	3114.143	14.708	0.989321	0.026
12.36	7.2	3683.619	3359.973	14.461	1.508956	0.007
12.33	8.8	3823.405	3646.056	14.733	1.902773	-0.109
12.31	9.7	3749.152	3758.124	14.622	1.903638	-0.061
12.31	8.8	3801.427	3533.163	14.539	1.75671	0.137
12.29	10.7	3926.413	3727.648	14.735	2.50521	0.021
12.29	10.7	3868.964	3705.275	14.415	2.380117	-0.02
12.25	12.4	4041.746	3893.323	14.666	2.777183	-0.088
12.25	12.4	4014.159	3912.102	14.72	2.877937	0.021
12.22	14.6	4219.611	4117.994	14.664	3.463749	0.008
12.22	14.6	4167.501	4122.562	14.697	3.384925	-0.05
12.22	14.6	4208.805	4056.704	14.676	3.389357	-0.14
12.22	14.6	4146.92	4051.619	14.758	3.435234	-0.085
12.21	14.6	4220.752	4036.027	14.54	3.304926	0.024
12.21	14.6	4186.735	4039.668	14.384	3.330504	0.034
12.21	14.6	4081.801	3971.821	14.719	3.400719	0.048
12.18	16.7	4324.616	4253.193	14.576	3.81669	-0.016
12.18	16.7	4220.823	4193.911	14.367	3.860884	-0.018
12.13	20.8	4508.619	4366.586	14.704	4.79045	0.139
12.13	20.4	4509.632	4304.655	14.46	4.733741	0.152
12.12	21.5	4542.206	4508.881	14.723	5.073591	0.015

12.12	21.5	4501.081	4523.283	14.731	4.929336	0.044
12.07	25.7	4616.095	4615.005	14.689	5.811184	0.18
12.07	25.7	4632.763	4611.133	14.737	5.835646	0.08
12.07	25.7	4589.441	4667.655	14.599	5.71166	0.173
12.07	25	4729.182	4517.992	14.534	5.576634	0.171
12.07	25	4650.642	4580.379	14.704	5.68245	0.244
12.05	27.1	4672.659	4722.408	14.707	6.037103	0.033
12.05	27.1	4693.854	4746.295	14.488	6.060745	-0.004
11.99	32	4745.357	4899.955	14.569	6.787178	0.027
11.99	32	4842.524	4904.099	14.708	6.914321	0.142
11.99	31.3	4791.832	4803.974	14.655	6.679794	0.073
11.99	31.3	4886.277	4795.446	14.65	6.778709	0.057
11.99	30.5	4882.318	4825.437	14.647	6.606058	0.335
11.98	32.7	4851.766	4922.586	14.733	6.963773	0.01

1462R & 2.1 3
D1 & D2, respectively
0.203333
0.175
Number of Points
28
33
67
Single Prop Data
-0.0475 1462R KT coeffs
0.2747
-0.834
0.6734
0.1821 1462R 10KQ coeffs
-0.3066
-0.7631
1.0905
0.1152 2.1/3 KT coeffs
-0.4397
0.0464
0.6971
0.4168 2.1/3 10KQ coeffs
-1.7006
0.9717
1.6318
CR data

Voltage	Amperage	RPM 1	RPM 2	Velocity (ft/s)	Thrust	RM
12.45	3.6	1925.668	1551.194	4.028	1.389177	0.025
12.4	5.7	2229.672	1914.837	3.925	2.281109	0.001
12.22	9.9	2607.003	2289.572	4.023	3.629879	-0.007
12.12	13.5	2932.54	2571.661	4.053	4.679673	-0.052
12.04	18.2	3087.467	2811.844	3.893	5.706256	0.062

11.99	21.4	3250.335	2990.711	3.861	6.499419	-0.001
11.92	25.9	3404.522	3197.362	3.888	7.525968	-0.077
12.52	2.6	1463.387	1528.322	3.932	0.912517	-0.155
12.51	2.8	1636.458	1551.509	3.994	1.079157	-0.155
12.51	2.8	1611.233	1578.785	4.047	1.098313	-0.182
12.48	3.1	1558.727	1702.399	4.003	1.16469	-0.267
12.48	3.1	1553.865	1726.293	4.082	1.168421	-0.291
12.47	3.2	1796.954	1594.073	3.915	1.207529	-0.045
12.47	3.2	1791.379	1579.383	4.108	1.272998	-0.078
12.45	3.6	1905.285	1564.584	3.826	1.348426	0.031
12.4	5.7	2187.636	1980.494	3.713	2.255116	0.038
12.31	8.6	2516.707	2231.738	4.032	3.216416	-0.056
12.31	8.6	2475.208	2258.87	3.986	3.308684	-0.055
12.31	8.6	2531.26	2251.223	3.927	3.295226	-0.057
12.25	9.5	2606.266	2278.361	4.022	3.564819	-0.058
12.15	13.7	2914.46	2597.508	4.085	4.708602	-0.114
12.15	13.7	2897.498	2615.404	4.069	4.806636	-0.136
12.13	13.3	2876.308	2541.217	4.019	4.577641	0.014
12.06	17.3	3101.552	2767.862	3.971	5.6608	-0.01
12.01	20.4	3230.383	2932.18	3.993	6.397098	-0.035
12.01	20.4	3246.586	2895.293	3.973	6.381917	0.009
11.96	24.2	3354.263	3137.603	3.963	7.16333	-0.051
11.94	24.9	3404.038	3216.683	4.005	7.467808	-0.145
12.43	3.4	2678	1903.859	9.574	0.765937	-0.015
12.35	5.6	2883.129	2225.466	9.726	1.546666	-0.064
12.25	7.9	3169.298	2416.895	9.57	2.221369	0.004
12.19	10.8	3342.882	2694.941	9.647	3.02733	-0.064
12.14	12.7	3530.028	2795.905	9.557	3.591528	0.012
12.11	15.1	3673.678	3005.435	9.672	4.323907	0.02
12.04	19.3	3786.019	3245.553	9.684	5.277627	-0.089
11.98	23.5	3973.602	3350.215	9.628	6.174617	0.078
11.92	28.8	4120.709	3524.159	9.635	7.118616	0.015
12.49	3	2412.044	1781.884	9.419	0.532131	-0.095
12.49	3	2436.775	1805.307	9.574	0.572937	-0.057
12.49	3	2452.99	1810.954	9.683	0.603484	-0.099
12.45	3.2	2582.109	1834.446	9.597	0.742203	-0.056
12.45	3.2	2509.558	1825.516	9.692	0.757776	0.012
12.38	5.5	2833.073	2199.662	9.584	1.467356	-0.035
12.29	8.2	3049.101	2476.811	9.739	2.37554	-0.16
12.27	8.2	3069.675	2452.178	9.753	2.339561	-0.084
12.27	8.2	3138.732	2451.935	9.482	2.240951	-0.1
12.26	7.7	3151.996	2419.349	9.612	2.177337	0.071
12.26	7.7	3117.889	2366.587	9.522	2.162581	0.047
12.21	10.5	3353.26	2636.319	9.822	3.064566	0.024
12.15	13	3532.319	2854.058	9.753	3.717561	-0.073
12.14	13.1	3509.776	2893.779	9.626	3.716332	-0.105
12.14	12.7	3514.724	2795.597	9.645	3.626044	0.047
12.11	15.1	3677.771	2973.432	9.684	4.219627	0.015
12.06	18.2	3760.811	3198.778	9.713	5.104794	-0.129

12.06	18	3799.085	3117.986	9.662	4.935475	0.108
12.06	18	3800.422	3095.654	9.862	5.026398	0.076
12.06	18	3773.839	3148.824	9.713	5.022794	0.058
11.99	23.6	3948.773	3419.143	9.656	6.144616	-0.13
11.99	23.6	3945.671	3440.693	9.783	6.293902	-0.236
11.99	22.8	3971.188	3371.397	9.742	5.952973	-0.056
11.94	27.7	4097.812	3518.062	9.547	7.067113	0.033
12.68	2.8	3106.851	2191.607	14.61	0.28104	0.032
12.61	5	3342.102	2571.986	14.654	1.014577	-0.062
12.52	6.5	3642.379	2638.518	14.639	1.460322	0.028
12.45	9	3830.101	2874.339	14.669	2.188835	-0.041
12.39	10.8	3934.96	3056.203	14.667	2.6584	-0.045
12.32	12.7	4045.589	3129.525	14.718	3.101501	-0.012
12.25	14.5	4217.804	3240.087	14.73	3.584118	-0.091
12.03	23.9	4597.774	3778.629	14.538	5.704495	0.034
12.7	2.8	3022.984	2180.452	14.432	0.223749	-0.065
12.7	2.8	3132.975	2171.685	14.562	0.258668	0.023
12.64	4.7	3330.23	2507.171	14.606	0.908174	0.024
12.64	4.7	3378.041	2550.017	14.694	0.947272	-0.058
12.57	7.2	3528.452	2854.942	14.645	1.735624	-0.127
12.57	7.2	3649.259	2868.073	14.734	1.793991	-0.165
12.53	8.6	3657.939	3021.025	14.586	1.90385	-0.247
12.53	7.3	3507.391	2783.638	14.509	1.632256	-0.155
12.53	7.3	3548.692	2796.864	14.6	1.654876	-0.1
12.53	6.9	3611.363	2778.107	14.538	1.598495	-0.063
12.53	6.9	3544.465	2715.441	14.658	1.647446	-0.184
12.53	6.5	3583.431	2648.791	14.623	1.343855	0.107
12.53	6.5	3529.296	2617.481	14.633	1.442022	0.049
12.53	6.5	3555.106	2624.018	14.703	1.518232	0.001
12.46	8.9	3791.197	2909.016	14.609	2.096823	-0.093
12.46	8.9	3819.17	2876.905	14.623	2.105855	-0.128
12.46	8.9	3728.128	2874.322	14.705	2.182668	0.012
12.41	11	3915.276	3071.403	14.786	2.687361	-0.091
12.41	11	3937.184	3058.677	14.72	2.810937	-0.125
12.41	11	3961.621	3032.101	14.746	2.743612	-0.116
12.39	11.8	3985.785	3207.776	14.692	2.905837	-0.251
12.39	11.8	3894.983	3168.422	14.752	2.857922	-0.19
12.39	10.8	3875.116	3055.613	14.731	2.617336	-0.098
12.39	10.8	3904.36	3025.869	14.615	2.661122	-0.001
12.35	12.8	4070.613	3176.868	14.613	3.163689	-0.142
12.34	13.5	4082.165	3259.198	14.611	3.390256	-0.186
12.34	13.5	4111.221	3294.642	14.597	3.412227	-0.266
12.34	12.8	4070.183	3158.317	14.743	3.124956	-0.027
12.34	12.8	4095.414	3104.649	14.705	3.164668	-0.034
12.34	12.1	4064.083	2993.812	14.766	2.995983	0.029
12.34	12.1	4058.369	3049.085	14.814	3.0385	0.043
12.34	12.1	4158.727	2996.127	14.56	2.914237	0.143
12.28	14.6	4236.671	3226.578	14.7	3.602579	-0.006
12.28	14.8	4188.309	3324.734	14.68	3.659226	-0.12

12.25	15.6	4189.523	3426.389	14.689	3.848184	-0.216
12.22	16.8	4247.369	3395.633	14.68	4.083226	-0.102
12.22	16.8	4377.303	3386.934	14.751	4.107704	-0.086
12.21	16.4	4302.523	3367.867	14.655	3.993794	-0.069
12.21	15.7	4244.831	3190.037	14.505	3.770396	0.031
12.21	15.7	4301.449	3225.2	14.293	3.776175	0.084
12.16	18.4	4371.846	3340.965	14.538	4.372495	0.215
12.16	18.4	4345.412	3343.53	14.299	4.469446	0.143
12.15	18.2	4444.335	3301.742	14.488	4.495745	0.331
12.13	19.9	4444.882	3405.66	14.487	4.728531	0.1
12.13	19.9	4397.359	3454.912	14.606	4.770174	0.128
12.11	20.2	4410.25	3546.032	14.509	4.936256	-0.041
12.11	20.2	4421.617	3613.767	14.448	4.859172	0.053
12.1	20.8	4440.736	3582.5	14.474	4.982742	-0.159
12.1	19.9	4474.46	3539.293	14.547	4.795434	0.111
12.05	23.5	4538.969	3660.647	14.263	5.467829	0.203
12.04	24	4552.021	3749.37	14.465	5.727813	0.066
11.98	28.8	4658.259	3955.167	14.611	6.451256	-0.113
11.98	28.8	4765.561	3974.557	14.681	6.540444	-0.052
11.97	28.6	4694.883	3872.177	14.551	6.488296	0.121
11.97	28.6	4795.708	3910.439	14.756	6.655797	0.201
11.91	33.5	4799.491	4158.45	14.607	7.14539	-0.141
11.9	33.5	4753.184	4145.622	14.393	7.123423	-0.199
11.9	33.5	4729.189	4106.164	14.719	7.270719	-0.199
11.9	33.2	4755.315	4062.423	14.721	7.168155	-0.241

```
1462R & 2055
D1 & D2, respectively
0.203333
0.180833
Number of Point
32
32
53
Single Prop Data
-0.0475   1462R KT coeffs
0.2747
-0.834
0.6734
0.1821    1462R 10KQ coeffs
-0.3066
-0.7631
1.0905
-0.0092   2055 KT coeffs
0.0437
-0.4532
0.7347
```

0.0052 2055 10KQ coeffs
-0.0157
-1.1393
2.19
CR data

Voltage	Amperage	RPM 1	RPM 2	Velocity (ft/s)	Thrust	RM
12.45	3.2	1820.717	1613.283	3.915502	1.270558	0.034
12.38	5.3	2107.069	1989.553	4.161382	2.091268	-0.085
12.27	7.9	2420.61	2228.467	3.95375	2.938787	-0.028
12.19	10.7	2712.47	2476.285	4.014947	3.845399	0.034
12.13	13.4	2902.849	2768.598	3.769067	4.736223	0.018
12.06	18	3080.751	2967.847	3.781088	5.848895	-0.035
11.95	26.1	3441.752	3362.422	3.920966	7.608875	-0.092
12.46	5.3	2114.741	2066.08	3.856491	2.096161	-0.101
12.44	4.2	1885.896	1912.5	3.398608	1.618555	-0.095
12.46	3	1698.594	1625.087	4.128598	1.150255	-0.071
12.46	3	1698.27	1613.807	3.830264	1.121668	-0.018
12.46	3	1718.075	1604.626	4.306725	1.147384	-0.108
12.4	5.1	2132.966	1946.47	3.839006	2.027164	0.004
12.4	5.1	2096.475	1970.81	4.022597	2.052855	-0.01
12.33	6.9	2301.251	2246.399	4.184331	2.710686	-0.106
12.33	6.9	2291.456	2218.598	4.142805	2.708125	-0.065
12.3	8.1	2389.074	2347.524	3.775624	3.028589	-0.08
12.3	8.1	2418.327	2316.343	4.173403	3.03701	-0.094
12.28	7.9	2428.478	2247.694	4.088165	2.950794	0.001
12.28	7.9	2390.443	2293.716	3.99637	2.944297	0.004
12.23	10	2594.247	2447.741	4.014947	3.604399	-0.072
12.23	10	2588.166	2464.337	4.20728	3.650113	-0.087
12.23	10	2606.345	2510.4	3.741747	3.610703	-0.015
12.2	10.6	2718.008	2478.403	3.896925	3.829483	0.024
12.2	10.6	2673.168	2510.656	4.20728	3.852113	0.045
12.15	13.3	2872.311	2699.417	4.191981	4.603161	-0.009
12.15	13.3	2856.029	2703.029	3.747211	4.618006	0.027
12.08	17.1	3101.791	2928.034	4.146083	5.640326	-0.046
12.08	17.1	3137.514	2905.034	4.087072	5.756728	0.025
12	22.8	3344.786	3139.86	3.92643	6.854193	-0.007
12	22.8	3338.55	3256.862	3.902389	7.043799	0.001
11.97	24.3	3368.986	3244.378	3.87944	7.187476	-0.096
12.39	6.6	3024.993	2383.181	9.602918	1.900044	0.046
12.3	8.7	3226.615	2667.159	9.56528	2.6067	-0.05
12.21	10.8	3318.275	2780.778	9.370817	3.101425	-0.021
12.13	13.7	3550.815	2976.648	9.948978	3.969161	0.065
12.07	16.5	3708.062	3163.695	9.908204	4.671164	0.005
12.01	19.6	3884.572	3391.485	9.769152	5.491897	-0.037
11.89	25.7	4049.378	3608.638	9.940614	6.617929	-0.049
11.71	33.9	4209.027	3950.592	9.613373	7.963532	-0.042
12.44	6.4	3051.039	2290.9	9.671921	1.766896	0.006
12.44	6.4	3003.446	2353.818	9.730469	1.91231	0.061

12.33	8.5	3183.244	2597.622	9.452366	2.489795	-0.04
12.33	8.5	3196.64	2594.158	9.877884	2.55972	-0.01
12.24	10.7	3419.402	2821.188	9.822473	3.195635	-0.136
12.23	10.7	3312.846	2790.686	9.929114	3.151236	-0.038
12.17	12.7	3483.208	2967.654	9.689694	3.620444	-0.041
12.17	12.7	3492.879	2928.317	9.947933	3.664007	-0.088
12.17	12.7	3468.696	2960.386	9.613373	3.635532	-0.063
12.15	13.3	3526.753	2930.468	9.691785	3.861745	0.03
12.15	13.3	3593.422	2970.91	9.710604	3.845449	0.027
12.09	16.1	3705.697	3128.25	9.889385	4.552404	0.068
12.09	16.1	3699.205	3157.474	9.589326	4.548112	0.002
12.07	16.5	3747.689	3167.246	9.943751	4.580391	0.046
12.07	16.5	3715.845	3156.315	9.921795	4.62616	0.049
12.01	19.6	3834.616	3388.793	9.967797	5.501937	-0.013
12.01	19.6	3853.457	3413.866	9.914477	5.526085	0.014
11.92	24.9	4071.629	3572.887	9.643692	6.430857	0.025
11.92	24.9	4019.881	3615.794	10.01798	6.479366	-0.099
11.83	28.2	4143.814	3664.261	9.821427	7.090482	-0.049
11.83	28.2	4132.351	3725.689	9.818291	7.049026	-0.059
11.79	30	4237.884	3705.776	9.595599	7.482003	0.115
11.75	31.8	4228.262	3800.846	9.92075	7.647006	-0.043
11.75	31.8	4211.412	3862.624	9.968843	7.730092	-0.154
12.46	6.2	3520.679	2638.44	14.8191	1.47162	0
12.38	9.1	3828.841	2968.541	14.65094	2.315913	-0.036
12.34	11	3953.613	3106.102	14.72977	2.777067	-0.027
12.3	12.3	3958.28	3143.654	14.57001	3.021397	-0.09
12.26	14	4177.55	3298.752	14.80964	3.562544	0.021
12.22	16.1	4249.302	3427.349	14.7119	4.091171	-0.012
12.12	20.3	4398.101	3621.938	14.53323	4.938467	0.008
12.06	24.1	4593.259	3780.048	14.75394	5.705346	0.036
12.45	7.7	3660.29	2852.241	14.83802	1.914775	-0.138
12.46	6.2	3556.747	2564.257	14.98831	1.480979	0.016
12.46	6.2	3558.79	2548.911	14.88216	1.393493	-0.024
12.43	7.7	3617.572	2752.812	14.4008	1.696087	0.03
12.43	7.7	3670.863	2791.472	14.8706	1.931945	-0.092
12.42	7.8	3702.14	2804.753	14.83276	1.95062	-0.084
12.39	9.1	3796.414	2882.363	15.12389	2.366218	-0.006
12.39	9.1	3822.782	2933.435	14.82541	2.220004	0.021
12.35	11.1	3918.133	3056.765	14.82541	2.817004	-0.058
12.35	11.1	3916.878	3158.562	14.66145	2.793195	0.006
12.34	11.3	4038.113	3129.447	14.80439	2.853391	-0.12
12.31	12.9	4078.448	3220.961	14.89687	3.291738	-0.063
12.31	12.9	4122.205	3306.881	14.75604	3.260805	-0.148
12.31	12.9	4106.118	3285.913	14.8212	3.324081	-0.074
12.3	12.5	4050.179	3160.712	14.56161	3.134583	0.06
12.3	12.5	4074.139	3214.48	14.59734	3.150301	0.13
12.3	12.3	4064.406	3130.935	14.81805	3.054389	0.012
12.3	12.3	3956.908	3166.371	14.39134	3.065069	-0.02
12.27	14.2	4129.743	3359.141	14.82751	3.565466	-0.142

12.27	14.2	4217.077	3343.364	14.91054	3.567755	-0.148
12.27	13.3	4177.417	3220.835	14.96834	3.375548	0.088
12.27	13.3	4054.758	3126.182	14.58578	3.333802	0.123
12.26	14	4228.427	3323.602	14.86114	3.561862	0.017
12.23	16.1	4227.664	3413.019	14.34195	3.867556	0.038
12.23	16.1	4271.49	3477.135	14.59419	4.050619	0.009
12.18	19.1	4383.886	3615.249	14.73292	4.800755	0.033
12.18	19.1	4367.166	3588.362	14.611	4.835257	0.092
12.16	20.2	4389.865	3635.325	14.24841	4.882748	-0.124
12.16	20.2	4364.234	3669.007	14.74553	4.932509	-0.087
12.16	20.2	4374.154	3699.738	15.19326	5.024795	-0.237
12.14	20.9	4366.097	3754.644	14.9284	5.047704	-0.355
12.14	20.9	4353.134	3761.781	14.75394	5.114346	-0.195
12.14	20.9	4337.294	3781.672	14.57212	5.049851	-0.188
12.13	20.5	4405.548	3709.448	14.6625	4.951423	-0.089
12.13	20.5	4324.788	3702.565	14.38294	4.923277	-0.168
12.13	20	4420.561	3652.992	14.44074	4.785619	0.038
12.13	20	4332.761	3660.894	14.58473	4.863575	-0.096
12.08	23.5	4535.418	3804.743	14.60995	5.547029	0.123
12.08	23.5	4467.702	3805.455	14.32513	5.568986	0.011
12.08	23.5	4602.167	3768.735	14.88006	5.723029	0.031
12.01	28.2	4716.487	3967.238	14.67091	6.55725	0.112
12.01	28.2	4719.251	3911.152	14.68037	6.576306	0.096
11.99	30	4703.084	4105.685	14.74028	6.775361	-0.147
11.99	30	4689.607	4084.246	14.46071	6.814893	-0.133
11.96	32.4	4876.137	4136.24	14.67722	7.102621	-0.018

```
1462R & Z55
D1 & D2, respectively
0.203333
0.180833
Number of Points
47
52
54
Single Prop Data
-0.0475   1462R KT coeffs
0.2747
-0.834
0.6734
0.1821    1462R 10KQ coeffs
-0.3066
-0.7631
1.0905
0.1486    Z55 KT coeffs
-0.2762
-0.2836
```

0.5001
0.4171 Z55 10KQ coeffs
-0.9068
-0.2116
0.9436
CR data

Voltage	Amperage	RPM 1	RPM 2	Velocity (ft/s)	Thrust	RM
12.52	2.5	1534.03	1976.802	3.751	0.858216	-0.002
12.1	2.3	1539.022	1837.191	3.98	0.77233	0.06
12.1	2.4	1541.613	1835.366	3.869	0.765877	0.045
12.51	2.8	1543.806	2172.601	3.958	1.042036	-0.088
12.52	2.5	1564.044	1960.853	3.902	0.881776	-0.014
12.51	2.8	1564.217	2122.856	4.084	1.058542	-0.113
12.1	2.4	1588.24	1819.289	4.056	0.789854	0.018
12.52	2.5	1589.121	2033.635	3.94	0.881983	-0.032
12.43	4.2	1911.656	2371.012	3.822	1.571199	0.098
12.41	4.7	1947.657	2538.687	3.98	1.77133	-0.022
12.46	4.1	1953.597	2316.298	3.938	1.542867	0.036
12.46	4.1	1970.336	2310.87	4.026	1.551058	-0.003
12.37	6.8	2194.465	2967.163	4.021	2.63576	-0.002
12.37	6.8	2196.231	2939.986	4.06	2.610094	-0.014
12.34	7	2215.722	2902.221	3.943	2.616158	-0.016
12.34	7	2226.987	2967.728	4.023	2.686879	-0.025
12.34	7	2263.543	2934.656	4.016	2.665462	-0.057
12.32	7	2292.142	3013.915	3.959	2.761095	-0.029
12.28	8.9	2422.027	3223.419	3.864	3.28859	0.031
12.26	9.1	2428.276	3207.532	3.989	3.382861	-0.068
12.28	8.9	2445.678	3191.556	4.041	3.338954	-0.025
12.2	10.9	2590.188	3468.484	3.93	4.0354	-0.051
12.2	10.9	2602.761	3459.733	4.003	4.02569	-0.027
12.18	11.1	2620.07	3446.55	3.866	4.041705	-0.017
12.2	10.9	2646.442	3439.293	3.93	3.9874	-0.013
12.08	14	2783.617	3761.199	3.979	4.923271	-0.007
12.13	13.3	2791.503	3655.601	4.019	4.681641	-0.018
12.13	13.3	2792.771	3631.481	3.932	4.684517	0.081
12.13	13.3	2794.814	3612.251	4.023	4.743879	0.05
12.09	14.1	2807.217	3737.57	3.892	4.903199	-0.018
12.13	13.3	2810.451	3658.656	3.986	4.757684	0.062
12.09	14.1	2832.086	3713.483	3.79	4.870395	0.007
12.04	16.6	2978.198	3892.643	4.052	5.638613	0.111
12.02	17.4	2998.758	3968.161	3.951	5.810626	0.06
12.02	17.4	3020.911	3970.402	4.031	5.792356	0.005
12.04	16.6	3022.557	3880.833	3.984	5.658566	0.155
12.01	17.5	3024.442	4012.459	3.813	5.95069	0.069
11.96	21.3	3152.552	4149.429	3.94	6.756983	0.203
11.94	22.2	3186.2	4297.183	3.7	7.018402	0.088
11.94	22.2	3188.03	4303.671	4.036	7.053655	-0.042
11.96	21.3	3247.629	4205.981	4.082	6.831421	0.097

11.92	22.6	3256.413	4348.001	3.935	7.076692	-0.009
11.87	26.4	3295.709	4546.116	3.961	7.837212	-0.076
11.87	26.4	3301.914	4588.566	3.972	7.963859	-0.071
11.87	26.4	3339.699	4535.827	3.996	7.853275	0.004
11.88	26.9	3351.894	4486.956	4.049	7.947433	0.125
11.84	29.2	3393.778	4600.853	4.055	8.329793	0.009
12.53	2.9	2444.543	2403.693	9.784	0.482047	0.017
12.53	2.9	2446.773	2340.656	9.724	0.436378	0.077
12.46	3.4	2476.861	2595.089	9.882	0.590323	-0.018
12.48	3.2	2492.079	2553.503	9.715	0.557082	-0.004
12.53	2.9	2511.124	2469.778	9.769	0.442875	0.05
12.38	5.4	2742.515	3010.89	9.855	1.305375	0.023
12.38	5.4	2757.982	3133.445	9.846	1.337062	-0.01
12.34	5.6	2804.798	3129.037	9.844	1.29077	-0.018
12.38	5.4	2821.361	3052.883	9.48	1.19767	0.023
12.27	7.7	3009.678	3465.438	9.836	2.135604	-0.004
12.27	7.7	3013.157	3485.526	9.71	2.135362	0.022
12.23	7.9	3055.699	3508.269	9.735	2.084963	0.004
12.27	7.7	3085.448	3495.704	9.779	2.057323	0.026
12.18	9.7	3157.265	3711.135	9.867	2.583128	-0.006
12.18	9.7	3184.511	3716.46	9.883	2.680469	-0.008
12.16	9.7	3206.58	3656.883	9.848	2.698354	-0.006
12.11	11.4	3276.1	3913.591	9.741	3.229829	0.044
12.08	12.1	3313.488	4039.219	9.796	3.311788	-0.051
12.06	12.1	3326.854	4010.892	9.748	3.366839	-0.051
12.08	12.1	3327.513	3985.325	9.831	3.374876	-0.038
12.11	11.4	3371.176	3887.931	9.826	3.200148	0.022
12.11	11.4	3390.814	3947.821	9.931	3.166514	0.076
11.99	14.1	3516.182	4180.732	9.903	3.8634	-0.06
11.93	16.2	3601.43	4364.064	9.708	4.381075	0.027
11.93	16.2	3617.202	4315.152	9.865	4.504836	0.008
11.85	19	3709.471	4507.763	9.948	4.973017	0.095
3	19.4	3712.889	4536.671	9.816	5.149693	0.018
11.85	19	3733.64	4490.681	9.951	5.008459	0.096
3	19.4	3739.392	4580.335	9.871	5.143713	-0.066
11.83	19.5	3799.21	4572.405	9.749	5.191984	0.052
11.78	24.6	3865.297	4938.28	9.793	6.326352	-0.039
11.79	23.6	3876.914	4864.897	9.569	6.167228	0.148
11.79	24.8	3877.808	4917.502	9.697	6.416494	-0.013
11.79	23.6	3884.057	4829.458	9.845	6.065916	0.035
11.8	22.7	3885.738	4750.753	9.931	5.957514	0.102
11.78	24.6	3903.088	4934.096	9.885	6.345762	-0.063
11.8	22.7	3923.75	4740.036	9.768	5.96273	0.208
11.76	28.2	3953.635	5144.708	9.944	7.159427	-0.122
11.77	26.7	3963.049	5053.504	9.773	6.849454	0.048
11.77	26.7	4003.951	5051.641	9.806	7.01524	0.057
11.76	28.2	4014.239	5143.511	9.831	7.218876	-0.149
11.77	26.7	4029.862	5059.025	9.843	6.968625	0.057
11.72	31.4	4034.314	5255.222	9.709	7.599218	-0.074

11.74	29.4	4049.073	5162.837	9.689	7.230345	0.133
11.73	30.5	4060.823	5207.359	9.849	7.6155	-0.033
11.76	27.5	4060.951	5047.614	9.872	6.995859	0.006
11.73	30.5	4071.21	5214.564	9.843	7.611625	-0.097
11.74	29.4	4073.821	5115.327	9.969	7.249115	0.058
11.74	29.4	4100.997	5151.496	9.903	7.3484	0.079
11.69	33.6	4115.223	5344.965	9.741	8.038829	-0.073
11.69	33.6	4154.418	5368.868	9.757	8.068139	-0.042
11.69	33.6	4184.842	5337.571	9.898	8.054667	0.019
12.56	3.2	3053.524	3148.769	14.858	0.208171	0.033
12.58	2.8	3081.994	2961.479	14.885	0.169119	0.104
12.55	3.3	3151.778	3140.132	14.849	0.21419	0.013
12.58	2.8	3166.503	3007.003	14.907	0.135974	0.066
12.49	5.3	3342.454	3501.923	14.688	0.853967	0.019
12.5	5.2	3364.572	3533.228	14.824	0.816695	0.015
12.5	5.2	3428.451	3548.833	14.79	0.854238	0.016
12.44	7.3	3564.279	3843.963	14.9	1.468428	0.039
12.44	7.3	3581.493	3882.023	14.826	1.422135	-0.033
12.38	9	3707.452	4035.688	14.833	2.040673	-0.007
12.38	9	3715.709	4081.028	14.526	1.936912	-0.006
12.36	9.2	3732.448	4110.074	14.822	1.953256	-0.032
12.31	10.5	3833.059	4230.763	14.68	2.275226	-0.061
12.32	11	3852.124	4283.006	14.648	2.358275	-0.151
12.3	10.7	3887.865	4276.166	14.648	2.331275	0.012
12.31	10.5	3910.247	4216.187	14.828	2.329574	0.037
12.23	12.8	3928.048	4482.631	14.748	2.937048	0.052
12.21	13	3935.024	4461.043	14.712	2.987193	-0.001
12.26	12.5	3958.338	4449.411	14.792	2.850676	0.104
12.26	12.5	3996.06	4411.622	14.549	2.813865	0.156
12.26	12.5	4006.334	4444.719	14.818	2.764378	0.053
12.23	12.8	4016.767	4502.916	14.8	2.934429	0.001
12.15	15.3	4044.032	4689.925	14.562	3.451668	-0.025
12.17	15	4058.392	4628.753	14.213	3.367283	0.067
12.23	12.8	4059.074	4450.353	14.888	3.01578	0.001
12.17	15	4078.303	4593.671	14.627	3.405721	-0.042
12.17	15	4087.063	4640.629	14.605	3.495958	-0.047
12.11	16.9	4158.211	4791.659	14.646	3.845841	0.133
12.11	16.9	4199.146	4782.133	14.558	3.857805	0.11
12.11	16.9	4216.853	4799.496	14.626	3.924505	0.029
12.08	17.6	4227.369	4838.694	14.671	4.02627	0.052
12.07	17.6	4239.287	4821.07	14.666	4.104183	-0.016
12.01	20.7	4273.095	5084.356	14.697	4.776925	-0.044
12.01	20.7	4329.986	5100.56	14.787	4.85858	-0.009
12.03	20.1	4355.621	4965.722	14.623	4.659855	0.077
12.03	20.1	4372.195	4937.811	14.763	4.580327	0.137
12.01	20.7	4379.565	5021.361	14.812	4.584061	0.027
11.95	23.4	4380.037	5233.146	14.719	5.130719	-0.01
12	20.7	4414.368	5049.764	14.686	4.769531	0.04
11.92	25	4421.621	5280.703	14.726	5.582246	0.055

11.95	23.9	4431.577	5249.959	14.789	5.345018	0.113
11.96	23.2	4451.136	5231.308	14.631	5.103588	-0.038
11.96	23.2	4459.143	5200.353	14.778	5.262609	0.017
11.95	23.9	4477.779	5209.353	14.747	5.40883	0.089
11.89	27	4499.656	5418.596	14.859	5.962391	0.069
11.93	25	4513.812	5321.924	14.763	5.585327	-0.01
11.89	27	4538.73	5403.881	14.775	5.933953	0.11
11.87	28	4547.261	5498.871	14.777	6.204391	-0.008
11.89	27	4553.566	5373.473	14.887	5.90856	0.052
11.93	25	4556.334	5247.063	14.594	5.661579	0.087
11.85	29.9	4574.182	5534.769	14.786	6.476361	0.036
11.83	31.3	4584.06	5605.459	14.789	6.693018	-0.078
11.88	28	4608.142	5546.538	14.854	6.12529	-0.051
11.85	29.9	4685.742	5529.989	14.768	6.529421	0.122

Version 1

Version 1 of the iteraction model – CR Prop Opt.for

```
C This program solves for four propeller interaction coefficients - a wake
C fraction (w), an axial velocity factor (avf), an aft propeller torque
C modification factor (Qf2), and an aft propeller thrust modification
C factor (Tf2) – in order to use propeller performance data to predict the
C counter-rotating performance of two unmatched Octura propellers
C
C*************************************************************************
C*          Optimization of CR prop parameters
C*************************************************************************
C
C       MAIN PROGRAM
C
       COMMON/PR/D1,D2,v(200),t(200),q(200),RPM1(200),RPM2(200),NP,
       1volt(200),amp(200),coeff(20),NP4,NP9,NP14,speed
       REAL RPM1,RPM2,J1,J2,KT1,KQ1,KT2,KQ2V,t,q,volt,amp,coeff
       Character(80) description
       Dimension X(4),XL(4),XU(4),G(2),WK(900),IWK(250),RPRM(20),IPRM(20)
       NRWK=900
       NRIWK=250
       Do 1 I=1,20
       RPRM(i)=0
1      IPRM(i)=0
       Method = 1
       NDV = 4
       NCON = 0

       open(5, file='CRdat_1462R_Z55.csv', status='old')
       open(unit=6, file='4_13_07, CRout_1462R_Z55_4fpse.out')
       speed = 4

       read(5,*)
       read(5,*)
       read(5,*)D1
       read(5,*)D2
       read(5,*)
       read(5,*)NP4
```

```
      read(5,*)NP9
      read(5,*)NP14
      read(5,*)
      do 2 i=1,16
2     read(5,*)coeff(i)
      read(5,*)
      read(5,*)
      if (speed.EQ.4) then
      NP = NP4
      do 36 i=1,NP4
36    read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
      do 37 i=1,(NP9+NP14)
37    read(5,*)
      elseif (speed.EQ.9) then
      NP = NP9
      do 38 i=1,NP4
38    read(5,*)
      do 39 i=1,NP9
39    read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
      do 40 i=1,NP14
40    read(5,*)
      else
      NP = NP14
      do 41 i=1,(NP4+NP9)
41    read(5,*)
      do 42 i=1,NP14
42    read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
      endif
      close(5)

      avf = 1
      qf2 = 1.44
      tf2 = 1.44
      w = 0.3

C  DV1 Wake fraction
      X(1)=w
      XL(1)=w
      XU(1)=w
C  DV2 axial velocity factor
      X(2)=avf
      XL(2)=avf
      XU(2)=avf
C  DV3 Torque factor 2
      X(3)=qf2
      XL(3)= qf2
      XU(3)= qf2
C  DV4 Thrust factor 2
      X(4)=tf2
      XL(4)=tf2
      XU(4)=tf2

      Iprint = 1
      MinMax = -1
      INFO = 0
      RPRM(3) = 0.000000000000000000001
      RPRM(4) = 0.000000000000000000001
      IPRM(4) = 5
      IPRM(9) = 5

      Do 20 Idot=1,5
10    Call Dot(INFO,Method,Iprint,NDV,NCON,X,XL,XU,Diff,MinMax,
      1G,RPRM,IPRM,WK,NRWK,IWK,NRIWK)
```

```
      If(INFO.eq.0)go to 20
      CALL DatDiff(X(1),X(2),X(3),X(4),Diff,0)
      Go to 10
20    continue
      CALL DatDiff(X(1),X(2),X(3),X(4),Diff,1)
101   Format(a16,1x,f4.2,1x,a13,1x,f4.2,1x,a13,f4.2)
      END

      SUBROUTINE DatDiff(w,avf,qf2,tf2,Diff,IP)
      COMMON/PR/D1,D2,v(200),t(200),q(200),RPM1(200),RPM2(200),NP,
      1volt(200),amp(200),coeff(20)
      INTEGER NP
      REAL RPM1,RPM2,J1,J2,KT1,KQ1,KT2,KQ2,D1,D2,DENS,Diff,Tdif,Qdif,
      1Toff,Qoff
      REAL V,t,q,A1,Va1,Va2,w,avf,qf2,tf2,volt,amp
      4Peff,Meff,Oeff,Epwr
      Real Pout, Eta1, Eta2,coeff
      DENS=1.94

      Toff=0
      Qoff=0
      if(ip.eq.1)then
      Write (6,*)
      Write (6,*)'Vel J1  J2   RPM1  RPM2  T1    T2    Q1    Q2    Ttot
      1Qtot Tmea Qmea Toff Qoff'
      endif

      do 11 i=1,NP

      Va1=v(i)*(1-w)
      J1=Va1/(RPM1(i)/60.)/D1

      KT1=coeff(1)*J1**3 + coeff(2)*J1**2 + coeff(3)*J1 + coeff(4)
      KQ1=(coeff(5)*J1**3 + coeff(6)*J1**2 + coeff(7)*J1 + coeff(8))/10

      T1=KT1*DENS*(RPM1(i)/60.)**2*D1**4
      Q1=KQ1*DENS*(RPM1(i)/60.)**2*D1**5*12

      CT1=T1/.5/DENS/Va1**2/3.1415926/(D1/2.)**2
      Etai1=2/(1+(CT1+1)**.5)
      a1=1/Etai1-1

      Va2=Va1*(1+avf*a1)
      J2=Va2/(RPM2(i)/60.)/D2

      KT2=coeff(9)*J2**3 + coeff(10)*J2**2 + coeff(11)*J2 + coeff(12)
      KQ2=(coeff(13)*J2**3 + coeff(14)*J2**2+coeff(15)*J2+coeff(16))/10
      KT2=KT2*Tf2
      KQ2=KQ2*Qf2

      T2=KT2*DENS*(RPM2(i)/60.)**2*D2**4
      Q2=KQ2*DENS*(RPM2(i)/60.)**2*D2**5*12

      Tdif=T(i)-(T1+T2)
      Qdif=Q(i)-(Q1-Q2)
      Toff=Toff+Tdif**2
      Qoff=Qoff+Qdif**2

      if(ip.eq.1)then
      Write (6,103)v(i),J1,J2,RPM1(i),RPM2(i),T1,T2,Q1,Q2,
      1(T1+T2),Q1-Q2,T(i),Q(i),(T1+T2)-T(i),Q1-Q2-Q(i)
      endif
```

```
11     continue
       Diff=Toff+Qoff

       if(ip.eq.1)then
       Write(6,*)
       Write(6,100)'Toff Qoff', Toff,Qoff
       write(*,*)Diff
       Write(6,*)
       Write (6,*)'Vel  RPM1  RPM2  Ttot  Qtot  PropEff  ElecPwr
       1MotorEff  OverallEff  Eta1  Eta2'

       do 20 i=1,NP
       Va1=v(i)*(1-w)
       J1=Va1/(RPM1(i)/60.)/D1

       KT1=coeff(1)*J1**3 + coeff(2)*J1**2 + coeff(3)*J1 + coeff(4)
       KQ1=(coeff(5)*J1**3 + coeff(6)*J1**2 + coeff(7)*J1 + coeff(8))/10
       Eta1=KT1/KQ1*J1/2./3.1415926

       T1=KT1*DENS*(RPM1(i)/60.)**2*D1**4
       Q1=KQ1*DENS*(RPM1(i)/60.)**2*D1**5*12

       CT1=T1/.5/DENS/Va1**2/3.1415926/(D1/2.)**2
       Etai1=2/(1+(CT1+1)**.5)
       a1=1/Etai1-1

       Va2=Va1*(1+avf*a1)
       J2=Va2/(RPM2(i)/60.)/D2

       KT2=coeff(9)*J2**3 + coeff(10)*J2**2 + coeff(11)*J2 + coeff(12)
       KQ2=(coeff(13)*J2**3+coeff(14)*J2**2+coeff(15)*J2+coeff(16))/10
       KT2=KT2*Tf2
       KQ2=KQ2*Qf2
       Eta2=KT2/KQ2*J2/2./3.1415926

       T2=KT2*DENS*(RPM2(i)/60.)**2*D2**4
       Q2=KQ2*DENS*(RPM2(i)/60.)**2*D2**5*12

       Pout=2*3.1415926/60*(RPM1(i)*Q1+RPM2(i)*Q2)/12
       Peff=(T1+T2)*Va1/Pout
       Epwr=volt(i)*amp(i)*0.7376
       Meff=Pout/Epwr
       Oeff=(T1+T2)*Va1/Epwr

       Write (6,104)v(i),RPM1(i),RPM2(i),(T1+T2),Q1-Q2,Peff,Epwr/.7376,
       1Meff,Oeff, Eta1, Eta2
20     continue
       endif

100    Format(A10, 4(1x,F6.2))
103    Format(f5.2,x,2(f4.2,x),2(f5.0,x),6(f5.2,x),4(f5.2,x))
104    Format(f4.1,x,2(f5.0,x),f4.2,x,f5.2,3x,f4.2,5x,f4.0,5x,2(f4.2,8x),
       12(f4.2,4x))
       return
       end
```

The Matlab thrust/torque surface plotting program for Version 1 – ErrorSurface1.m

```
function ErrorSurface1

plottype = 2; % 1 for thrust, 2 for roll
speed = 9;
[D1,D2,NP4,NP9,NP14,coeffs,DataMat] = Data145720;

% Set DVs

avf1 = 1;
Qf2 = 1.44;
Tf2 = 1.44;

if speed == 4;
    w = 0.0558;
elseif speed == 9;
    w = 0.126;
else % speed == 14;
    w = 0.186;
end

% Sort Data
VOLT = DataMat(:,1);
AMP = DataMat(:,2);
RPMS1 = DataMat(:,3);
RPMS2 = DataMat(:,4);
V = DataMat(:,5);
T = DataMat(:,6);
Q = DataMat(:,7);

if speed == 4
    NP = NP4;
    VOLT = DataMat(1:NP,1);
    AMP = DataMat(1:NP,2);
    RPMS1 = DataMat(1:NP,3);
    RPMS2 = DataMat(1:NP,4);
    V = DataMat(1:NP,5);
    T = DataMat(1:NP,6);
    Q = DataMat(1:NP,7);
elseif speed == 9
    NP = NP9;
    VOLT = DataMat(NP4+1:NP4+NP,1);
    AMP = DataMat(NP4+1:NP4+NP,2);
    RPMS1 = DataMat(NP4+1:NP4+NP,3);
    RPMS2 = DataMat(NP4+1:NP4+NP,4);
    V = DataMat(NP4+1:NP4+NP,5);
    T = DataMat(NP4+1:NP4+NP,6);
    Q = DataMat(NP4+1:NP4+NP,7);
else %speed == 14
    NP = NP14;
    VOLT = DataMat(NP4+NP9+1:NP4+NP9+NP,1);
    AMP = DataMat(NP4+NP9+1:NP4+NP9+NP,2);
    RPMS1 = DataMat(NP4+NP9+1:NP4+NP9+NP,3);
    RPMS2 = DataMat(NP4+NP9+1:NP4+NP9+NP,4);
    V = DataMat(NP4+NP9+1:NP4+NP9+NP,5);
    T = DataMat(NP4+NP9+1:NP4+NP9+NP,6);
    Q = DataMat(NP4+NP9+1:NP4+NP9+NP,7);
end

rpm1lo = fix(min(RPMS1/1000))*1000;
rpm1up = ceil(max(RPMS1/1000))*1000;
```

```
drpm1 = 100;

rpm2lo = fix(min(RPMS2/1000))*1000;
rpm2up = ceil(max(RPMS2/1000))*1000;
drpm2 = 100;

size1 = (rpm1up-rpm1lo)/drpm1+1;
size2 = (rpm2up-rpm2lo)/drpm2+1;

sum = 0;
for i=1:NP
    sum = sum + V(i);
end
vel = sum/NP;

dens=1.94;

Mat(1,1) = 0;

f = 0;
for rpm1 = rpm1lo:drpm1:rpm1up
    f = f+1;
      Mat(f+1,1) = rpm1;
       h = 0;

    for rpm2 = rpm2lo:drpm2:rpm2up
       h = h+1;
       Mat(1,h+1) = rpm2;

       Va1=vel*(1-w);
           J1=Va1/(rpm1/60)/D1;

           KT1=coeffs(1)*J1^3 + coeffs(2)*J1^2 + coeffs(3)*J1 + coeffs(4);
           KQ1=(coeffs(5)*J1^3 + coeffs(6)*J1^2 + coeffs(7)*J1 + coeffs(8))/10;

           T1=KT1*dens*(rpm1/60)^2*D1^4;
           Q1=KQ1*dens*(rpm1/60)^2*D1^5*12;

           CT1=T1/0.5/dens/Va1^2/pi/(D1/2)^2;
           Etai1=2/(1+(CT1+1)^0.5);
           a1=1/Etai1-1;

           Va2=Va1*(1+avf1*a1);
           J2=Va2/(rpm2/60)/D2;

           KT2=coeffs(9)*J2^3 + coeffs(10)*J2^2 + coeffs(11)*J2 + coeffs(12);
           KQ2=(coeffs(13)*J2^3 + coeffs(14)*J2^2+coeffs(15)*J2+coeffs(16))/10;
           KT2=KT2*Tf2;
           KQ2=KQ2*Qf2;

           T2=KT2*dens*(rpm2/60)^2*D2^4;
           Q2=KQ2*dens*(rpm2/60)^2*D2^5*12;

           Thrust = T1+T2;
           Roll = Q1-Q2;

           if plottype == 1
               Mat(f+1,h+1) = Thrust;
           else
               Mat(f+1,h+1) = Roll;
           end
       end
    end
```

```
[row,col] = size(Mat);

x = Mat(1,2:col);
y = Mat(2:row,1);
val = Mat(2:row,2:col);

contour3(x(1,:),y(:,1),val, 140);
surface(x(1,:),y(:,1),val,'EdgeColor',[.8 .8 .8],'FaceColor','none');
grid off;
colormap;

hold on
grid on

if plottype == 1
       hold on; xlabel('RPM2','FontSize',13); ylabel('RPM1','FontSize',13);
zlabel('Thrust, lbs','FontSize',13); title('Thrust','FontSize',14)
    plot3(RPMS2, RPMS1,T,'o','MarkerEdgeColor','k','MarkerFaceColor','k',
'MarkerSize',4)
else
    hold on; xlabel('RPM2','FontSize',13); ylabel('RPM1','FontSize',13); zlabel('Roll,
in-lbs','FontSize',13); %title('Roll','FontSize',14)
    plot3(RPMS2, RPMS1, Q,'o','MarkerEdgeColor','k','MarkerFaceColor','k',
'MarkerSize',4)
end

set(gca,'FontSize',12)

Tmax = max(T);
Qmax = max(Q);

% Calculate Errors

Toff = 0;
Qoff = 0;
Toffs = 0;
Qoffs = 0;

for i=1:NP

    Va1=V(i)*(1-w);
    J1=Va1/(RPMS1(i)/60)/D1;

    KT1=coeffs(1)*J1^3 + coeffs(2)*J1^2 + coeffs(3)*J1 + coeffs(4);
    KQ1=(coeffs(5)*J1^3 + coeffs(6)*J1^2 + coeffs(7)*J1 + coeffs(8))/10;

    T1=KT1*dens*(RPMS1(i)/60)^2*D1^4;
    Q1=KQ1*dens*(RPMS1(i)/60)^2*D1^5*12;

    CT1=T1/0.5/dens/Va1^2/pi/(D1/2)^2;
    Etai1=2/(1+(CT1+1)^0.5);
    a1=1/Etai1-1;

    Va2=Va1*(1+avf1*a1);
    J2=Va2/(RPMS2(i)/60)/D2;

    KT2=coeffs(9)*J2^3 + coeffs(10)*J2^2 + coeffs(11)*J2 + coeffs(12);
    KQ2=(coeffs(13)*J2^3 + coeffs(14)*J2^2+coeffs(15)*J2+coeffs(16))/10;
    KT2=KT2*Tf2;
    KQ2=KQ2*Qf2;

    T2=KT2*dens*(RPMS2(i)/60)^2*D2^4;
    Q2=KQ2*dens*(RPMS2(i)/60)^2*D2^5*12;
```

```
    Tdif=T(i)-(T1+T2);
    Qdif=Q(i)-(Q1-Q2);
    Toff=Toff+Tdif;
    Qoff=Qoff+Qdif;
    Toffs=Toffs+Tdif^2;
    Qoffs=Qoffs+Qdif^2;
end

Diff=Toff+Qoff;
Epp = Diff/NP;

Diffs=Toffs+Qoffs;
Epps = Diffs/NP;

if plottype == 1
    text(rpm2up-1200,rpm1lo+600,Tmax+Tmax/2+7.5,'F(x)/NP = ','FontSize',12);
    text(rpm2up-900,rpm1lo+600,Tmax+Tmax/2+5.5,'(Thrust Error)/NP = ','FontSize',12);

    text(rpm2up-600,rpm1lo+140,Tmax+Tmax/2+7.5,sprintf(' %.3f',Epps),'FontSize',12);
    text(rpm2up-400,rpm1lo+140,Tmax+Tmax/2+5.75,sprintf(' %.3f',Toff/NP),
'FontSize',12);

else % plottype == 2
    text(rpm2up-1000,rpm1lo+600,8.5,'F(x)/NP = ','FontSize',12);
    text(rpm2up-1000,rpm1lo+600,7.5,'(Torque Error)/NP = ','FontSize',12);

    text(rpm2up-400,rpm1lo+140,8.5,sprintf(' %.3f',Epps),'FontSize',12);
    text(rpm2up-400,rpm1lo+140,7.45,sprintf(' %.3f',Qoff/NP),'FontSize',12);
end
```

Version 2

Version 2 of the interaction model – CR Prop Opt2.for

```
C This program solves for five propeller interaction coefficients - a wake
C fraction (w), a front propeller axial velocity factor (avf1), an aft
C propeller axial velocity factor (avf2), an aft propeller RPM factor (rpmf2),
C and an aft propeller thrust modification factor (Tf2) – in order to use
C propeller performance data to predict the counter-rotating performance of
C two unmatched Octura propellers
C
C*************************************************************************
C*          Optimization of CR prop parameters
C*************************************************************************
C
C       MAIN PROGRAM
C
       COMMON/PR/D1,D2,v(200),t(200),q(200),RPM1(200),RPM2(200),NP,
       1ITmax,volt(200),amp(200),coeff(20),NP4,NP9,NP14,speed
       REAL RPM1,RPM2,J1,J2,KT1,KQ1,KT2,KQ2V,t,q,volt,amp,coeff
       Character(80) description
       Dimension X(5),XL(5),XU(5),G(2),WK(900),IWK(250),RPRM(20),IPRM(20)
       NRWK=900
       NRIWK=250
       Do 1 I=1,20
       RPRM(i)=0
1      IPRM(i)=0
       Method = 1
       NDV = 5
```

```
      NCON = 0

      open(5, file='CRdat_1462R_Z55.csv', status='old')
      open(unit=6, file='4_14_07, CRout3_1462R_Z55_14fps.out')
      speed = 14

      w = 0.1
      avf1 = 1.001
      avf2 = 0.999
      rpm2f = 0.5
      tf2 = 1.5

      read(5,*)
      read(5,*)
      read(5,*)D1
      read(5,*)D2
      read(5,*)
      read(5,*)NP4
      read(5,*)NP9
      read(5,*)NP14
      read(5,*)
      do 2 i=1,16
2     read(5,*)coeff(i)
      read(5,*)
      read(5,*)
      if (speed.EQ.4) then
      NP = NP4
      do 36 i=1,NP4
36    read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
      do 37 i=1,(NP9+NP14)
37    read(5,*)
      elseif (speed.EQ.9) then
      NP = NP9
      do 38 i=1,NP4
38    read(5,*)
      do 39 i=1,NP9
39    read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
      do 40 i=1,NP14
40    read(5,*)
      else
      NP = NP14
      do 41 i=1,(NP4+NP9)
41    read(5,*)
      do 42 i=1,NP14
42    read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
      endif
      close(5)

C  DV1 Wake fraction
      X(1)=w
      XL(1)=0
      XU(1)=0.4
C  DV2 axial velocity factor 1
      X(2)=avf1
      XL(2)=1
      XU(2)=2
C  DV3 axial velocity factor 2
      X(3)=avf2
      XL(3)=0
      XU(3)=1
C  DV4 RPM2 factor
      X(4)=rpm2f
      XL(4)= 0
```

```
      XU(4)= 2
C  DV5 Thrust factor 2
      X(5)=tf2
      XL(5)=0.5
      XU(5)=2

      Iprint = 0
      MinMax = -1
      INFO = 0
      RPRM(3) = 0.000000000000000000001
      RPRM(4) = 0.000000000000000000001
      IPRM(4) = 5
      IPRM(9) = 5

      ITmax = 200

      Do 20 Idot=1,5
      if (Idot.eq.5) Iprint=1
10    Call Dot(INFO,Method,Iprint,NDV,NCON,X,XL,XU,Diff,MinMax,
      1G,RPRM,IPRM,WK,NRWK,IWK,NRIWK)
      If(INFO.eq.0)go to 20
      CALL DatDiff(X(1),X(2),X(3),X(4),X(5),Diff,0)
      Go to 10
20    continue
      CALL DatDiff(X(1),X(2),X(3),X(4),X(5),Diff,1)
101   Format(a16,1x,f4.2,1x,a13,1x,f4.2,1x,a13,f4.2)
      END

      SUBROUTINE DatDiff(w,avf1,avf2,rpm2f,tf2,Diff,IP)
      COMMON/PR/D1,D2,v(200),t(200),q(200),RPM1(200),RPM2(200),NP,
      1ITmax,volt(200),amp(200),coeff(20),a2(100)
      INTEGER NP,ITmax
      REAL RPM1,RPM2,J1,J2,KT1,KQ1,KT2,KQ2,D1,D2,DENS,Diff,Tdif,Qdif,
      1Toff,Qoff
      REAL V,t,q,A1,a2guess,Va1,Va2,w,avf1,avf2,rpm2f,tf2,volt,amp,a2
      4Peff,Meff,Oeff,Epwr
      Real Pout, Eta1, Eta2, coeff
      DENS=1.94

      Toff=0
      Qoff=0
      if(ip.eq.1)then
      Write (6,*)
      Write (6,*)'Vel J1  J2   RPM1  RPM2  T1    T2    Q1    Q2    Ttot
      1Qtot Tmea Qmea Toff Qoff'
      endif
      do 11 i=1,NP

      a2guess = 0.1

      do 12 j=1,ITmax

      Va1=v(i)*(1-w)*(1+avf2*a2guess)
      J1=Va1/(RPM1(i)/60.)/D1

      KT1=coeff(1)*J1**3 + coeff(2)*J1**2 + coeff(3)*J1 + coeff(4)
      KQ1=(coeff(5)*J1**3 + coeff(6)*J1**2 + coeff(7)*J1 + coeff(8))/10

      T1=KT1*DENS*(RPM1(i)/60.)**2*D1**4
      Q1=KQ1*DENS*(RPM1(i)/60.)**2*D1**5*12

      CT1=T1/.5/DENS/Va1**2/3.1415926/(D1/2.)**2
      Etai1=2/(1+(CT1+1)**.5)
```

```
      a1=1/Etai1-1

      RPM2corr = RPM2(i) + RPM1(i)*a1*rpm2f

      Va2=Va1*(1+avf1*a1)
      J2=Va2/(RPM2corr/60.)/D2

      KT2=coeff(9)*J2**3 + coeff(10)*J2**2 + coeff(11)*J2 + coeff(12)
      KQ2=(coeff(13)*J2**3 + coeff(14)*J2**2+coeff(15)*J2+coeff(16))/10
      KT2=KT2*Tf2

      T2=KT2*DENS*(RPM2corr/60.)**2*D2**4
      Q2=KQ2*DENS*(RPM2corr/60.)**2*D2**5*12

      CT2=T2/.5/DENS/Va2**2/3.1415926/(D2/2.)**2
      Etai2=2/(1+(CT2+1)**.5)
      a2(i)=1/Etai2-1

      if ((abs(a2(i)-a2guess)/a2(i)).LT.0.001) go to 13

12    a2guess = a2(i)
      if (J.eq.ITmax) then
      write(*,*)'a2 not converging'
      stop
      endif

13    continue

      Tdif=T(i)-(T1+T2)
      Qdif=Q(i)-(Q1-Q2)
      Toff=Toff+Tdif**2
      Qoff=Qoff+Qdif**2

      if(ip.eq.1)then
      Write (6,103)v(i),J1,J2,RPM1(i),RPM2(i),T1,T2,Q1,Q2,
      1(T1+T2),Q1-Q2,T(i),Q(i),(T1+T2)-T(i),Q1-Q2-Q(i)
      endif

11    continue
      Diff=Toff+Qoff

      if(ip.eq.1)then
      Write(6,*)
      Write(6,100)'Toff Qoff', Toff,Qoff
      Write(6,*)
      Write (6,*)'Vel  RPM1  RPM2  Ttot  Qtot  PropEff  ElecPwr
      1MotorEff  OverallEff  Eta1  Eta2'

      do 20 i=1,NP
      Va1=v(i)*(1-w)*(1+avf2*a2(i))
      J1=Va1/(RPM1(i)/60.)/D1

      KT1=coeff(1)*J1**3 + coeff(2)*J1**2 + coeff(3)*J1 + coeff(4)
      KQ1=(coeff(5)*J1**3 + coeff(6)*J1**2 + coeff(7)*J1 + coeff(8))/10
      Eta1=KT1/KQ1*J1/2./3.1415926

      T1=KT1*DENS*(RPM1(i)/60.)**2*D1**4
      Q1=KQ1*DENS*(RPM1(i)/60.)**2*D1**5*12

      CT1=T1/.5/DENS/Va1**2/3.1415926/(D1/2.)**2
      Etai1=2/(1+(CT1+1)**.5)
      a1=1/Etai1-1
```

```
      RPM2corr = RPM2(i) + RPM1(i)*a1*rpm2f

      Va2=Va1*(1+avf1*a1)
      J2=Va2/(RPM2corr/60.)/D2

      KT2=coeff(9)*J2**3 + coeff(10)*J2**2 + coeff(11)*J2 + coeff(12)
      KQ2=(coeff(13)*J2**3+coeff(14)*J2**2+coeff(15)*J2+coeff(16))/10
      KT2=KT2*Tf2
      Eta2=KT2/KQ2*J2/2./3.1415926

      T2=KT2*DENS*(RPM2(i)/60.)**2*D2**4
      Q2=KQ2*DENS*(RPM2corr/60.)**2*D2**5*12

      Pout=2*3.1415926/60*(RPM1(i)*Q1+RPM2corr*Q2)/12
      Peff=(T1+T2)*Va1/Pout
      Epwr=volt(i)*amp(i)*0.7376
      Meff=Pout/Epwr
      Oeff=(T1+T2)*Va1/Epwr

      Write (6,104)v(i),RPM1(i),RPM2(i),(T1+T2),Q1-Q2,Peff,Epwr/.7376,
      1Meff,Oeff, Eta1, Eta2
20    continue
      endif

100   Format(A10, 4(1x,F6.2))
103   Format(f5.2,x,2(f4.2,x),2(f5.0,x),6(f5.2,x),4(f5.2,x))
104   Format(f4.1,x,2(f5.0,x),f4.2,x,f5.2,3x,f4.2,5x,f4.0,5x,2(f4.2,8x),
      12(f4.2,4x))
      return
      end
```

The Fortran program used to create the coefficient interaction matrices for Version 2 – CR Prop Opt 2b.for. The output matrices were then used in conjunction with speed2b.m to produce the interaction plots.

```
C This program is an extension of CR Prop Opt 2.for. This program iterates
C any 2 interaction coefficients to give the error associated with different
C combinations of the coefficients. This program outputs a matrix to be used in
C speed2b.m
C
C**********************************************************************
C*          Optimization of CR prop parameters
C**********************************************************************
C
C       MAIN PROGRAM
C
      COMMON/PR/D1,D2,v(200),t(200),q(200),RPM1(200),RPM2(200),NP,f,h,
      1sizew,sizerpm2f,ITmax,volt(200),amp(200),coeff(20),
      2Mat(2000,2000),NP4,NP9,NP14,speed
      INTEGER NP,f,h,sizew,sizerpm2f
      REAL RPM1,RPM2,J1,J2,KT1,KQ1,KT2,KQ2V,t,q,volt,amp,coeff,Mat
      Character(80) description
      Dimension X(5),XL(5),XU(5),G(2),WK(900),IWK(250),RPRM(20),IPRM(20)
      NRWK=900
      NRIWK=250
      Do 1 I=1,20
      RPRM(i)=0
1     IPRM(i)=0
```

```
      Method = 1
      NDV = 5
      NCON = 0

      open(5, file='CRdat_1457R_Z55.csv', status='old')
      open(unit=6, file='4_15_07, CRout4_1457R_Z55_14fps.out')
      speed = 14

      wlo = 0
      wup = 0.4
      dw = 0.01

      rpm2flo = 0
      rpm2fup = 2
      drpm2f = 0.1

      avf1 = 1.001
      avf2 = 0.999
      tf2 = 1.2

      read(5,*)
      read(5,*)
      read(5,*)D1
      read(5,*)D2
      read(5,*)
      read(5,*)NP4
      read(5,*)NP9
      read(5,*)NP14
      read(5,*)
      do 2 i=1,16
2     read(5,*)coeff(i)
      read(5,*)
      read(5,*)
      if (speed.EQ.4) then
      NP = NP4
      do 36 i=1,NP4
36    read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
      do 37 i=1,(NP9+NP14)
37    read(5,*)
      elseif (speed.EQ.9) then
      NP = NP9
      do 38 i=1,NP4
38    read(5,*)
      do 39 i=1,NP9
39    read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
      do 40 i=1,NP14
40    read(5,*)
      else
      NP = NP14
      do 41 i=1,(NP4+NP9)
41    read(5,*)
      do 42 i=1,NP14
42    read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
      endif
      close(5)

      sizew = (wup-wlo)/dw + 1
      sizerpm2f = (rpm2fup-rpm2flo)/drpm2f + 1

      Mat(1,1) = 0

      f = 0
      do 113 w = wlo,wup,dw
```

```
       f = f+1
       Mat(f+1,1) = w
       write(*,*)w
       h = 0
       do 112 rpm2f = rpm2flo,rpm2fup,drpm2f
       h = h+1
       Mat(1,h+1) = rpm2f

C  DV1 Wake fraction
       X(1)=w
       XL(1)=w
       XU(1)=w
C  DV2 axial velocity factor 1
       X(2)=avf1
       XL(2)=avf1
       XU(2)=avf1
C  DV3 axial velocity factor 2
       X(3)=avf2
       XL(3)=avf2
       XU(3)=avf2
C  DV4 RPM2 factor
       X(4)=rpm2f
       XL(4)= rpm2f
       XU(4)= rpm2f
C  DV5 Thrust factor 2
       X(5)=tf2
       XL(5)=tf2
       XU(5)=tf2

       Iprint = 0
       MinMax = -1
       INFO = 0
       RPRM(3) = 0.000000000000000000001
       RPRM(4) = 0.000000000000000000001
       IPRM(4) = 5
       IPRM(9) = 5

       ITmax = 200

       Do 20 Idot=1,5
10     Call Dot(INFO,Method,Iprint,NDV,NCON,X,XL,XU,Diff,MinMax,
       1G,RPRM,IPRM,WK,NRWK,IWK,NRIWK)
       If(INFO.eq.0)go to 20
       CALL DatDiff(X(1),X(2),X(3),X(4),X(5),Diff,0)
       Go to 10
20     continue
       CALL DatDiff(X(1),X(2),X(3),X(4),X(5),Diff,1)

112    continue
113    continue

       END

       SUBROUTINE DatDiff(w,avf1,avf2,rpm2f,tf2,Diff,IP)
       COMMON/PR/D1,D2,v(200),t(200),q(200),RPM1(200),RPM2(200),NP,
       1f,h,sizew,sizerpm2f,ITmax,volt(200),amp(200),coeff(20),
       2Mat(2000,2000),a2(100)
       INTEGER NP,f,h,sizew,sizerpm2f,ITmax
       REAL RPM1,RPM2,J1,J2,KT1,KQ1,KT2,KQ2,D1,D2,DENS,Diff,Tdif,Qdif,
       1Toff,Qoff
       REAL V,t,q,A1,a2guess,Va1,Va2,w,avf1,avf2,rpm2f,tf2,volt,amp,a2
       4Peff,Meff,Oeff,Epwr
       Real Pout, Eta1, Eta2, coeff,Mat
```

```
      DENS=1.94

      Toff=0
      Qoff=0

      do 11 i=1,NP

      a2guess = 0.1

      do 12 j=1,ITmax

      Va1=v(i)*(1-w)*(1+avf2*a2guess)
      J1=Va1/(RPM1(i)/60.)/D1

      KT1=coeff(1)*J1**3 + coeff(2)*J1**2 + coeff(3)*J1 + coeff(4)
      KQ1=(coeff(5)*J1**3 + coeff(6)*J1**2 + coeff(7)*J1 + coeff(8))/10

      T1=KT1*DENS*(RPM1(i)/60.)**2*D1**4
      Q1=KQ1*DENS*(RPM1(i)/60.)**2*D1**5*12

      CT1=T1/.5/DENS/Va1**2/3.1415926/(D1/2.)**2
      Etai1=2/(1+(CT1+1)**.5)
      a1=1/Etai1-1

      RPM2corr = RPM2(i) + RPM1(i)*a1*rpm2f

      Va2=Va1*(1+avf1*a1)
      J2=Va2/(RPM2corr/60.)/D2

      KT2=coeff(9)*J2**3 + coeff(10)*J2**2 + coeff(11)*J2 + coeff(12)
      KQ2=(coeff(13)*J2**3 + coeff(14)*J2**2+coeff(15)*J2+coeff(16))/10
      KT2=KT2*Tf2

      T2=KT2*DENS*(RPM2corr/60.)**2*D2**4
      Q2=KQ2*DENS*(RPM2corr/60.)**2*D2**5*12

      CT2=T2/.5/DENS/Va2**2/3.1415926/(D2/2.)**2
      Etai2=2/(1+(CT2+1)**.5)
      a2(i)=1/Etai2-1

      if ((abs(a2(i)-a2guess)/a2(i)).LT.0.001) go to 13

12    a2guess = a2(i)
      if (J.eq.ITmax) then
      write(*,*)'a2 not converging'
      stop
      endif

13    continue

      Tdif=T(i)-(T1+T2)
      Qdif=Q(i)-(Q1-Q2)
      Toff=Toff+Tdif**2
      Qoff=Qoff+Qdif**2

11    continue
      Diff=Toff+Qoff

      if(ip.eq.1)then

      do 20 i=1,NP
      Va1=v(i)*(1-w)*(1+avf2*a2(i))
      J1=Va1/(RPM1(i)/60.)/D1
```

```
      KT1=coeff(1)*J1**3 + coeff(2)*J1**2 + coeff(3)*J1 + coeff(4)
      KQ1=(coeff(5)*J1**3 + coeff(6)*J1**2 + coeff(7)*J1 + coeff(8))/10
      Eta1=KT1/KQ1*J1/2./3.1415926

      T1=KT1*DENS*(RPM1(i)/60.)**2*D1**4
      Q1=KQ1*DENS*(RPM1(i)/60.)**2*D1**5*12

      CT1=T1/.5/DENS/Va1**2/3.1415926/(D1/2.)**2
      Etai1=2/(1+(CT1+1)**.5)
      a1=1/Etai1-1

      RPM2corr = RPM2(i) + RPM1(i)*a1*rpm2f

      Va2=Va1*(1+avf1*a1)
      J2=Va2/(RPM2corr/60.)/D2

      KT2=coeff(9)*J2**3 + coeff(10)*J2**2 + coeff(11)*J2 + coeff(12)
      KQ2=(coeff(13)*J2**3+coeff(14)*J2**2+coeff(15)*J2+coeff(16))/10
      KT2=KT2*Tf2
      Eta2=KT2/KQ2*J2/2./3.1415926

      T2=KT2*DENS*(RPM2(i)/60.)**2*D2**4
      Q2=KQ2*DENS*(RPM2corr/60.)**2*D2**5*12

      Pout=2*3.1415926/60*(RPM1(i)*Q1+RPM2corr*Q2)/12
      Peff=(T1+T2)*Va1/Pout
      Epwr=volt(i)*amp(i)*0.7376
      Meff=Pout/Epwr
      Oeff=(T1+T2)*Va1/Epwr

      if (i.eq.NP) Mat(f+1,h+1) = Diff/NP

      if (i.eq.NP .and. f.eq.sizew .and. h.eq.sizerpm2f) then
      do 18 o=1,sizerpm2f+1
      write(6,19)(Mat(n,o),n=1,sizew+1)
18    continue
      endif

19    format(1000f20.6)

20    continue
      endif

      return
      end
```

The Matlab file used to create the coefficient interaction plots – speed2b.m

```
function speed2b
clc

Mat=[INSERT];

[row,col] = size(Mat);

x = Mat(1,2:col);
y = Mat(2:row,1);
val = Mat(2:row,2:col);
step1 = max(val);
```

```
valmax = max(step1);
valstep = valmax/5;

errstep1 = min(val);
minerr = min(errstep1);

for i = 1:row-1
    for j = 1:col-1
        if val(i,j) == minerr
            w = Mat(1,j+1);
            rpmf2 = Mat(i+1,1);
        end
    end
end

contour3(x(1,:),y(:,1),val, 140);
surface(x(1,:),y(:,1),val,'EdgeColor',[.8 .8 .8],'FaceColor','none');
grid off;
colormap;

hold on; xlabel('w','FontSize',13); ylabel('rpmf_2','FontSize',13); title('F(x)/point
- 14 ft/s','FontSize',14)
set(gca,'FontSize',12)
```

Version 3

Version 3 of the interaction model – CR Prop Opt3.for

```
C This program solves for six interaction coefficients - a wake fraction (w),
C a front propeller axial velocity factor (avf1), an aft propeller axial
C velocity factor (avf2), an aft propeller RPM correction factor (rpmf2), an
C aft propeller thrust modification factor (Tf2) and an aft prop torque
C modification factor (Qf2) – in order to use propeller performance data to
C predict the counter-rotating performance of two unmatched Octura propellers
C
C*************************************************************************
C*          Optimization of CR prop parameters
C*************************************************************************
C
C       MAIN PROGRAM
C
       COMMON/PR/D1,D2,v(200),t(200),q(200),RPM1(200),RPM2(200),NP,
       1ITmax,volt(200),amp(200),coeff(20),NP4,NP9,NP14,speed
       REAL RPM1,RPM2,J1,J2,KT1,KQ1,KT2,KQ2V,t,q,volt,amp,coeff
       Character(80) description
       Dimension X(6),XL(6),XU(6),G(2),WK(900),IWK(250),RPRM(20),IPRM(20)
       NRWK=900
       NRIWK=250
       Do 1 I=1,20
       RPRM(i)=0
1      IPRM(i)=0
       Method = 1
       NDV = 6
       NCON = 0

       open(5, file='CRdat_1457R_2.1 3.csv', status='old')
       open(unit=6, file='4_02_07, CRout11_1457R_2.1 3_4fps.out')
       speed = 4
```

```
       if (speed.eq.4) then
       w = 0.014
       avf2 = 0.93
       endif
       if (speed.eq.9) then
       w = 0.065
       avf2 = 0.735
       endif
       if (speed.eq.14) then
       w = 0.111
       avf2 = 0.54
       endif

       avf1 = 1
       rpm2f = 0.474
       tf2 = 1.1
       qf2 = 0.894

C  DV1 Wake fraction
       X(1)=w
       XL(1)=w
       XU(1)=w
C  DV2 axial velocity factor 1
       X(2)=avf1
       XL(2)=avf1
       XU(2)=avf1
C  DV3 axial velocity factor 2
       X(3)=avf2
       XL(3)=avf2
       XU(3)=avf2
C  DV4 RPM2 factor
       X(4)=rpm2f
       XL(4)= rpm2f
       XU(4)= rpm2f
C  DV5 Thrust factor 2
       X(5)=tf2
       XL(5)=tf2
       XU(5)=tf2
C  DV6 Torque factor 2
       X(6) = qf2
       XL(6) = qf2
       XU(6) = qf2

       read(5,*)
       read(5,*)
       read(5,*)D1
       read(5,*)D2
       read(5,*)
       read(5,*)NP4
       read(5,*)NP9
       read(5,*)NP14
       read(5,*)
       do 2 i=1,16
2      read(5,*)coeff(i)
       read(5,*)
       read(5,*)
       if (speed.EQ.4) then
       NP = NP4
       do 36 i=1,NP4
36     read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
       do 37 i=1,(NP9+NP14)
37     read(5,*)
       elseif (speed.EQ.9) then
```

```
      NP = NP9
      do 38 i=1,NP4
38    read(5,*)
      do 39 i=1,NP9
39    read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
      do 40 i=1,NP14
40    read(5,*)
      else
      NP = NP14
      do 41 i=1,(NP4+NP9)
41    read(5,*)
      do 42 i=1,NP14
42    read(5,*)volt(i),amp(i),RPM1(i),RPM2(i),v(i),t(i),q(i)
      endif
      close(5)

      Iprint = 0
      MinMax = -1
      INFO = 0
      RPRM(3) = 0.000000000000000000001
      RPRM(4) = 0.000000000000000000001
      IPRM(4) = 5
      IPRM(9) = 5

      ITmax = 200

      Do 20 Idot=1,5
10    Call Dot(INFO,Method,Iprint,NDV,NCON,X,XL,XU,Diff,MinMax,
      1G,RPRM,IPRM,WK,NRWK,IWK,NRIWK)
      IF(Idot.eq.5) Iprint = 1
      If(INFO.eq.0)go to 20
      CALL DatDiff(X(1),X(2),X(3),X(4),X(5),X(6),Diff,0)
      Go to 10
20    continue
      CALL DatDiff(X(1),X(2),X(3),X(4),X(5),X(6),Diff,1)
101   Format(a16,1x,f4.2,1x,a13,1x,f4.2,1x,a13,f4.2)
      END

      SUBROUTINE DatDiff(w,avf1,avf2,rpm2f,tf2,qf2,Diff,IP)
      COMMON/PR/D1,D2,v(200),t(200),q(200),RPM1(200),RPM2(200),NP,
      1ITmax,volt(200),amp(200),coeff(20),a2(100)
      INTEGER NP,ITmax
      REAL RPM1,RPM2,J1,J2,KT1,KQ1,KT2,KQ2,D1,D2,DENS,Diff,Tdif,Qdif,
      1Toff,Qoff
      REAL V,t,q,A1,a2guess,Va1,Va2,w,avf1,avf2,rpm2f,tf2,qf2,volt,amp,
      4a2,Peff,Meff,Oeff,Epwr
      Real Pout, Eta1, Eta2, coeff
      DENS=1.94

      Toff=0
      Qoff=0
      if(ip.eq.1)then
      Write (6,*)
      Write (6,*)'Vel J1  J2   RPM1  RPM2  T1    T2    Q1    Q2    Ttot
      1Qtot Tmea Qmea Toff Qoff'
      endif
      do 11 i=1,NP

      a2guess = 0.1

      do 12 j=1,ITmax

      Va1=v(i)*(1-w)*(1+avf2*a2guess)
```

```
      J1=Va1/(RPM1(i)/60.)/D1

      KT1=coeff(1)*J1**3 + coeff(2)*J1**2 + coeff(3)*J1 + coeff(4)
      KQ1=(coeff(5)*J1**3 + coeff(6)*J1**2 + coeff(7)*J1 + coeff(8))/10

      T1=KT1*DENS*(RPM1(i)/60.)**2*D1**4
      Q1=KQ1*DENS*(RPM1(i)/60.)**2*D1**5*12

      CT1=T1/.5/DENS/Va1**2/3.1415926/(D1/2.)**2
      Etai1=2/(1+(CT1+1)**.5)
      a1=1/Etai1-1

      RPM2corr=RPM2(i)+Va1*a1*rpm2f/(0.7*D1/2)*J1*(2*3.1415926)**2*120
      1/100

      Va2=Va1*(1+avf1*a1)
      J2=Va2/(RPM2corr/60.)/D2

      KT2=coeff(9)*J2**3 + coeff(10)*J2**2 + coeff(11)*J2 + coeff(12)
      KQ2=(coeff(13)*J2**3 + coeff(14)*J2**2+coeff(15)*J2+coeff(16))/10
      KT2=KT2*Tf2
      KQ2=KQ2*Qf2

      T2=KT2*DENS*(RPM2corr/60.)**2*D2**4
      Q2=KQ2*DENS*(RPM2corr/60.)**2*D2**5*12

      CT2=T2/.5/DENS/Va2**2/3.1415926/(D2/2.)**2
      Etai2=2/(1+(CT2+1)**.5)
      a2(i)=1/Etai2-1

      if ((abs(a2(i)-a2guess)/a2(i)).LT.0.001) go to 13

12    a2guess = a2(i)
      if (J.eq.ITmax) then
      write(*,*)'a2 not converging'
      stop
      endif

13    continue

      Tdif=T(i)-(T1+T2)
      Qdif=Q(i)-(Q1-Q2)
      Toff=Toff+Tdif**2
      Qoff=Qoff+Qdif**2

      if(ip.eq.1)then
      Write (6,103)v(i),J1,J2,RPM1(i),RPM2(i),T1,T2,Q1,Q2,
      1(T1+T2),Q1-Q2,T(i),Q(i),(T1+T2)-T(i),Q1-Q2-Q(i)
      endif

11    continue
      Diff=Toff+Qoff

      if(ip.eq.1)then
      Write(6,*)
      Write(6,100)'Toff Qoff', Toff,Qoff
      Write(6,*)
      Write (6,*)'Vel  RPM1  RPM2  Ttot  Qtot  PropEff  ElecPwr
      1MotorEff  OverallEff  Eta1  Eta2'

      do 20 i=1,NP

      Va1=v(i)*(1-w)*(1+avf2*a2(i))
```

```
      J1=Va1/(RPM1(i)/60.)/D1

      KT1=coeff(1)*J1**3 + coeff(2)*J1**2 + coeff(3)*J1 + coeff(4)
      KQ1=(coeff(5)*J1**3 + coeff(6)*J1**2 + coeff(7)*J1 + coeff(8))/10
      Eta1=KT1/KQ1*J1/2./3.1415926

      T1=KT1*DENS*(RPM1(i)/60.)**2*D1**4
      Q1=KQ1*DENS*(RPM1(i)/60.)**2*D1**5*12

      CT1=T1/.5/DENS/Va1**2/3.1415926/(D1/2.)**2
      Etai1=2/(1+(CT1+1)**.5)
      a1=1/Etai1-1

      RPM2corr=RPM2(i)+Va1*a1*rpm2f/(0.7*D1/2)*J1*(2*3.1415926)**2*120
      1/100

      Va2=Va1*(1+avf1*a1)
      J2=Va2/(RPM2corr/60.)/D2

      KT2=coeff(9)*J2**3 + coeff(10)*J2**2 + coeff(11)*J2 + coeff(12)
      KQ2=(coeff(13)*J2**3+coeff(14)*J2**2+coeff(15)*J2+coeff(16))/10
      KT2=KT2*Tf2
      KQ2=KQ2*Qf2
      Eta2=KT2/KQ2*J2/2./3.1415926

      T2=KT2*DENS*(RPM2(i)/60.)**2*D2**4
      Q2=KQ2*DENS*(RPM2corr/60.)**2*D2**5*12

      Pout=2*3.1415926/60*(RPM1(i)*Q1+RPM2corr*Q2)/12
      Peff=(T1+T2)*Va1/Pout
      Epwr=volt(i)*amp(i)*0.7376
      Meff=Pout/Epwr
      Oeff=(T1+T2)*Va1/Epwr

      Write (6,104)v(i),RPM1(i),RPM2(i),(T1+T2),Q1-Q2,Peff,Epwr/.7376,
      1Meff,Oeff, Eta1, Eta2
20    continue
      endif

100   Format(A10, 4(1x,F6.2))
103   Format(f5.2,x,2(f4.2,x),2(f5.0,x),6(f5.2,x),4(f5.2,x))
104   Format(f4.1,x,2(f5.0,x),f4.2,x,f5.2,3x,f4.2,5x,f4.0,5x,2(f4.2,8x),
      12(f4.2,4x))

      if (ip.eq.1) write(*,*)Diff

      return
      end
```

The Matlab thrust/torque surface plotting program for Version 3 – ErrorSurface3.m

```
function ErrorSurface3
clc

plottype = 1; %1 for thrust, 2 for roll
speed = 14;
[D1,D2,NP4,NP9,NP14,coeffs,DataMat] = Data146220;

% Set DVs
```

```
mw = 0.0129122;
bw = -0.0231159;
avf1 = 1.00019;
mavf2 = -0.0468778;
bavf2 = 0.828232;
rpm2f = 0.243667;
Tf2 = 1.18446;
Qf2 = 1.07092;

% Sort Data
VOLT = DataMat(:,1);
AMP = DataMat(:,2);
RPMS1 = DataMat(:,3);
RPMS2 = DataMat(:,4);
V = DataMat(:,5);
T = DataMat(:,6);
Q = DataMat(:,7);

if speed == 4
    NP = NP4;
    VOLT = DataMat(1:NP,1);
    AMP = DataMat(1:NP,2);
    RPMS1 = DataMat(1:NP,3);
    RPMS2 = DataMat(1:NP,4);
    V = DataMat(1:NP,5);
    T = DataMat(1:NP,6);
    Q = DataMat(1:NP,7);
elseif speed == 9
    NP = NP9;
    VOLT = DataMat(NP4+1:NP4+NP,1);
    AMP = DataMat(NP4+1:NP4+NP,2);
    RPMS1 = DataMat(NP4+1:NP4+NP,3);
    RPMS2 = DataMat(NP4+1:NP4+NP,4);
    V = DataMat(NP4+1:NP4+NP,5);
    T = DataMat(NP4+1:NP4+NP,6);
    Q = DataMat(NP4+1:NP4+NP,7);
elseif speed == 14
    NP = NP14;
    VOLT = DataMat(NP4+NP9+1:NP4+NP9+NP,1);
    AMP = DataMat(NP4+NP9+1:NP4+NP9+NP,2);
    RPMS1 = DataMat(NP4+NP9+1:NP4+NP9+NP,3);
    RPMS2 = DataMat(NP4+NP9+1:NP4+NP9+NP,4);
    V = DataMat(NP4+NP9+1:NP4+NP9+NP,5);
    T = DataMat(NP4+NP9+1:NP4+NP9+NP,6);
    Q = DataMat(NP4+NP9+1:NP4+NP9+NP,7);
else
end

rpm1lo = fix(min(RPMS1/1000))*1000;
rpm1up = ceil(max(RPMS1/1000))*1000;
drpm1 = 100;

rpm2lo = fix(min(RPMS2/1000))*1000;
rpm2up = ceil(max(RPMS2/1000))*1000;
drpm2 = 100;

size1 = (rpm1up-rpm1lo)/drpm1+1;
size2 = (rpm2up-rpm2lo)/drpm2+1;

ITmax = 20;

Tmax = max(T);
Qmax = max(Q);
```

```
sum = 0;
for i=1:NP
    sum = sum + V(i);
end
vel = sum/NP;

dens=1.94;

Mat(1,1) = 0;

f = 0;
for rpm1 = rpm1lo:drpm1:rpm1up
    f = f+1;
       Mat(f+1,1) = rpm1;

    h = 0;
    for rpm2 = rpm2lo:drpm2:rpm2up
        h = h+1;
        Mat(1,h+1) = rpm2;
        a2guess = 1;
        flag = 0;

        for j=1:ITmax

           w = mw*vel + bw;
           avf2 = mavf2*vel + bavf2;

           Va1=vel*(1-w)*(1+avf2*a2guess);
           J1=Va1/(rpm1/60)/D1;

           KT1=coeffs(1)*J1^3 + coeffs(2)*J1^2 + coeffs(3)*J1 + coeffs(4);
           KQ1=(coeffs(5)*J1^3 + coeffs(6)*J1^2 + coeffs(7)*J1 + coeffs(8))/10;

           T1=KT1*dens*(rpm1/60)^2*D1^4;
           Q1=KQ1*dens*(rpm1/60)^2*D1^5*12;

           CT1=T1/0.5/dens/Va1^2/pi/(D1/2)^2;
           Etai1=2/(1+(CT1+1)^0.5);
           a1=1/Etai1-1;

           RPM2corr = rpm2 + Va1*a1*rpm2f/(0.7*D1/2)*J1*(2*pi)^2*120/100;

           Va2=Va1*(1+avf1*a1);
           J2=Va2/(RPM2corr/60)/D2;

           KT2=coeffs(9)*J2^3 + coeffs(10)*J2^2 + coeffs(11)*J2 + coeffs(12);
           KQ2=(coeffs(13)*J2^3 + coeffs(14)*J2^2+coeffs(15)*J2+coeffs(16))/10;
           KT2=KT2*Tf2;
           KQ2=KQ2*Qf2;

           T2=KT2*dens*(RPM2corr/60)^2*D2^4;
           Q2=KQ2*dens*(RPM2corr/60)^2*D2^5*12;

           CT2=T2/0.5/dens/Va2^2/pi/(D2/2)^2;
           Etai2=2/(1+(CT2+1)^0.5);
           a2(j)=1/Etai2-1;

           if ((abs(a2(j)-a2guess)/a2(j))<0.001) %& (a1>0) & (a2(j)>0)
                break
           end

           a2guess = a2(j);
```

```
        end

        w = mw*vel + bw;
        avf2 = mavf2*vel + bavf2;

        Va1=vel*(1-w)*(1+avf2*a2(j));
        J1=Va1/(rpm1/60)/D1;

        KT1=coeffs(1)*J1^3 + coeffs(2)*J1^2 + coeffs(3)*J1 + coeffs(4);
        KQ1=(coeffs(5)*J1^3 + coeffs(6)*J1^2 + coeffs(7)*J1 + coeffs(8))/10;

        T1=KT1*dens*(rpm1/60)^2*D1^4;
        Q1=KQ1*dens*(rpm1/60)^2*D1^5*12;

        CT1=T1/0.5/dens/Va1^2/pi/(D1/2)^2;
        Etai1=2/(1+(CT1+1)^0.5);
        a1=1/Etai1-1;

        RPM2corr = rpm2 + Va1*a1*rpm2f/(0.7*D1/2)*J1*(2*pi)^2*120/100;

        Va2=Va1*(1+avf1*a1);
        J2=Va2/(RPM2corr/60)/D2;

        KT2=coeffs(9)*J2^3 + coeffs(10)*J2^2 + coeffs(11)*J2 + coeffs(12);
        KQ2=(coeffs(13)*J2^3 + coeffs(14)*J2^2+coeffs(15)*J2+coeffs(16))/10;
        KT2=KT2*Tf2;
        KQ2=KQ2*Qf2;

        T2=KT2*dens*(RPM2corr/60)^2*D2^4;
        Q2=KQ2*dens*(RPM2corr/60)^2*D2^5*12;

        Thrust = T1+T2;
        Roll = Q1-Q2;

        if (plottype == 2) & (imag(Roll) == 0)
           Mat(f+1,h+1) = Roll;
        else
           Mat(f+1,h+1) = Thrust;
        end
    end
end

[row,col] = size(Mat);

x = Mat(1,2:col);
y = Mat(2:row,1);
val = Mat(2:row,2:col);

contour3(x(1,:),y(:,1),val, 140);
surface(x(1,:),y(:,1),val,'EdgeColor',[.8 .8 .8],'FaceColor','none');
grid off;
colormap;

hold on
grid on
if plottype == 2
    hold on; xlabel('RPM2','FontSize',13); ylabel('RPM1','FontSize',13); zlabel('Roll,
in-lbs','Fontsize',13)% title('Roll','FontSize',14)
    plot3(RPMS2, RPMS1, Q,'o','MarkerEdgeColor','k','MarkerFaceColor','k',
'MarkerSize',4)
else
    hold on; xlabel('RPM2','FontSize',13); ylabel('RPM1','FontSize',13);
zlabel('Thrust, lbs','FontSize',13); %title('Thrust','FontSize',14)
```

```
    plot3(RPMS2, RPMS1, T,'o','MarkerEdgeColor','k','MarkerFaceColor',
'k','MarkerSize',4)
end

set(gca,'FontSize',12)

% Calculate Errors

clear a2

Toff = 0;
Qoff = 0;
Toffs = 0;
Qoffs = 0;

for i=1:NP
    a2guess = 0.1;

    for j=1:ITmax

        w = mw*V(i) + bw;
        avf2 = mavf2*V(i) + bavf2;

        Va1=V(i)*(1-w)*(1+avf2*a2guess);
        J1=Va1/(RPMS1(i)/60)/D1;

        KT1=coeffs(1)*J1^3 + coeffs(2)*J1^2 + coeffs(3)*J1 + coeffs(4);
        KQ1=(coeffs(5)*J1^3 + coeffs(6)*J1^2 + coeffs(7)*J1 + coeffs(8))/10;

        T1=KT1*dens*(RPMS1(i)/60)^2*D1^4;
        Q1=KQ1*dens*(RPMS1(i)/60)^2*D1^5*12;

        CT1=T1/0.5/dens/Va1^2/pi/(D1/2)^2;
        Etai1=2/(1+(CT1+1)^0.5);
        a1=1/Etai1-1;

        RPM2corr = RPMS2(i) + Va1*a1*rpm2f/(0.7*D1/2)*J1*(2*pi)^2*120/100;

        Va2=Va1*(1+avf1*a1);
        J2=Va2/(RPM2corr/60)/D2;

        KT2=coeffs(9)*J2^3 + coeffs(10)*J2^2 + coeffs(11)*J2 + coeffs(12);
        KQ2=(coeffs(13)*J2^3 + coeffs(14)*J2^2+coeffs(15)*J2+coeffs(16))/10;
        KT2=KT2*Tf2;
        KQ2=KQ2*Qf2;

        T2=KT2*dens*(RPM2corr/60)^2*D2^4;
        Q2=KQ2*dens*(RPM2corr/60)^2*D2^5*12;

        CT2=T2/0.5/dens/Va2^2/pi/(D2/2)^2;
        Etai2=2/(1+(CT2+1)^0.5);
        a2(i)=1/Etai2-1;

        if ((abs(a2(i)-a2guess)/a2(i))<0.001)
            break
        end

        if (j==ITmax)
            disp('a2 not converging')
            break
        end

        a2guess = a2(i);
```

```
    end

    w = mw*V(i) + bw;
    avf2 = mavf2*V(i) + bavf2;

    Va1=V(i)*(1-w)*(1+avf2*a2(i));
    J1=Va1/(RPMS1(i)/60)/D1;

    KT1=coeffs(1)*J1^3 + coeffs(2)*J1^2 + coeffs(3)*J1 + coeffs(4);
    KQ1=(coeffs(5)*J1^3 + coeffs(6)*J1^2 + coeffs(7)*J1 + coeffs(8))/10;

    T1=KT1*dens*(RPMS1(i)/60)^2*D1^4;
    Q1=KQ1*dens*(RPMS1(i)/60)^2*D1^5*12;

    CT1=T1/0.5/dens/Va1^2/pi/(D1/2)^2;
    Etai1=2/(1+(CT1+1)^0.5);
    a1=1/Etai1-1;

    RPM2corr = RPMS2(i) + Va1*a1*rpm2f/(0.7*D1/2)*J1*(2*pi)^2*120/100;

    Va2=Va1*(1+avf1*a1);
    J2=Va2/(RPM2corr/60)/D2;

    KT2=coeffs(9)*J2^3 + coeffs(10)*J2^2 + coeffs(11)*J2 + coeffs(12);
    KQ2=(coeffs(13)*J2^3 + coeffs(14)*J2^2+coeffs(15)*J2+coeffs(16))/10;
    KT2=KT2*Tf2;
    KQ2=KQ2*Qf2;

    T2=KT2*dens*(RPM2corr/60)^2*D2^4;
    Q2=KQ2*dens*(RPM2corr/60)^2*D2^5*12;

    Tdif=T(i)-(T1+T2);
    Qdif=Q(i)-(Q1-Q2);
    Toff=Toff+Tdif;
    Qoff=Qoff+Qdif;
    Toffs=Toffs+Tdif^2;
    Qoffs=Qoffs+Qdif^2;

end

Diff=Toff+Qoff;
Epp = Diff/NP;

Diffs=Toffs+Qoffs;
Epps = Diffs/NP;

text(rpm2up-600,rpm1lo+600,Tmax+Tmax/2+1.5,'F(x)/NP = ','FontSize',12);
text(rpm2up-100,rpm1lo+140,Tmax+Tmax/2+1.65,sprintf(' %.3f',Epps),'FontSize',12);
```

The final propeller prediction program – Cr Prop Predictor.for

```
C This program calculates the predicted thrust and torque of a counter-
C rotating propeller set at a given velocity, RPM1, and RPM2. The
C diameters and single propeller curves (as functions of J) for both props
C are needed.
C
C The single propeller equations should be put into the four functions at
C the end of the program. KT1 and KQ1 are the KT and 10KQ curves of the
C front propeller, respectively. KT2 and KQ2 are the KT and 10KQ curves of
C the aft propeller, respectively. --J Jacobson 4/05/07
```

```
C
C*************************************************************************
C*                                                                       *
C*                     CR Prop Performance Predictor                     *
C*                                                                       *
C*************************************************************************
C
C
      COMMON/PR/flag,mw,bw,avf1,mavf2,bavf2,rpm2f,tf2,qf2,ITmax
      REAL D1,D2,V,RPM1,RPM2,mw,bw,avf1,mavf2,bavf2,rpm2f,tf2,qf2
      INTEGER flag,ITmax

      WRITE(*,*)'Please Enter Description of Run'
      READ(*,'(a80)') description
      write(*,*)

      WRITE(*,*)'Please Enter D1 and D2 (in)'
      READ(*,*)D1,D2
      WRITE(*,*)

      WRITE(*,*)'Use the default wake fraction value (y/n)?'
      WRITE(*,*)'(Default value for cylindrical body 3 ft long with 3 in
      1 diameter)'
      READ(*,'(A1)')ans
      WRITE(*,*)

      if (ans.eq.'y'.or.ans.eq.'Y') goto 1

      if (ans.eq.'n'.or.ans.eq.'N') then
      WRITE(*,*)'Please Enter the Velocity (ft/s)'
      WRITE(*,*)'Velocity = V*(1-w)'
      READ(*,*)V
      WRITE(*,*)
      flag = 1
      goto 2
      endif

1     WRITE(*,*)'Please Enter V (ft/s)'
      READ(*,*) V
      WRITE(*,*)

2     WRITE(*,*)'Please Enter the RPMs of the Front and Aft Props'
      READ(*,*)RPM1,RPM2
      WRITE(*,*)

      mw = 0.0129122
      bw = -0.0231159
      avf1 = 1.00019
      mavf2 = -0.0468778
      bavf2 = 0.828232
      rpm2f = 0.243667
      tf2 = 1.18446
      qf2 = 1.07092

      ITmax = 200

      CALL CRprops(D1,D2,V,RPM1,RPM2)

      END


      SUBROUTINE CRprops(D1,D2,V,RPM1,RPM2)
      COMMON/PR/flag,mw,bw,avf1,mavf2,bavf2,rpm2f,tf2,qf2,ITmax
```

```
      REAL D1,D2,V,RPM1,RPM2,mw,bw,avf1,mavf2,bavf2,rpm2f,tf2,qf2,dens,
      1a2guess,w,avf2,Va1,J1,KT1,T1,KQ1,Q1,CT1,Etai1,a1,RPM2corr,Va2,J2,
       2KT2,T2,KQ2,Q2,CT2,Etai2,a2,Ttot,Qtot,Eta
      INTEGER flag,ITmax

10    dens = 1.94
      a2guess = 0.1

      do 11 j=1,ITmax

            if (flag.eq.1) then
                  Va1 = V*(1+avf2*a2guess)
            endif

            if (flag.eq.0) then
                  w = mw*V + bw
                  avf2 = mavf2*V + bavf2

                  Va1 = V*(1-w)*(1+avf2*a2guess)
            endif

            J1 = Va1/(RPM1/60)/D1

            T1=KT1(J1)*dens*(RPM1/60)**2*D1**4
            Q1=KQ1(J1)*dens*(RPM1/60)**2*D1**5*12

            CT1=T1/0.5/dens/Va1**2/3.1415926/(D1/2)**2
            Etai1=2/(1+(CT1+1)**0.5)
            a1=1/Etai1-1

            RPM2corr=RPM2+Va1*a1*rpm2f/(0.7*D1/2)*J1*(2*3.1415926)**2*120
      1     /100

            Va2=Va1*(1+avf1*a1)
            J2=Va2/(RPM2corr/60.)/D2

            T2=Tf2*KT2(J2)*dens*(RPM2corr/60)**2*D2**4
            Q2=Qf2*KQ2(J2)*dens*(RPM2corr/60)**2*D2**5*12

            CT2=T2/0.5/dens/Va2**2/3.1415926/(D2/2)**2
            Etai2=2/(1+(CT2+1)**0.5)
            a2=1/Etai2-1

            if ((abs(a2-a2guess)/a2).lt.0.001) go to 12

11          a2guess = a2

      if (j.eq.ITmax) then
            write(*,*)'a2 not converging'
            stop
      endif


12    Ttot = T1 + T2
      Qtot = Q1 - Q2

      Eta = Ttot*V/(2*3.1415926*(RPM1/60)*Q1/12 + 2*3.1415926*(RPM2/60)
      1*Q2/12)

      write(*,*)'Vel(ft/s)   RPM1   RPM2   Thrust(lbs)   Torque(ft-lbs)
      1    Eta'
      write(*,100)V,RPM1,RPM2,Ttot,Qtot,Eta
```

```
      write(*,*)
      WRITE(*,*)'Another point (y/n)?'
      READ(*,'(A1)')ans

      if (ans.eq.'y'.or.ans.eq.'Y') then
      write(*,*)

      if (flag.eq.1) then
      WRITE(*,*)'Please Enter D1, D2, V*(1-w), RPM1 and RPM2'
      endif

      if (flag.eq.0) then
      WRITE(*,*)'Please Enter D1, D2, V, RPM1 and RPM2'
      endif

      READ(*,*)D1,D2,V,RPM1,RPM2
      go to 10
      endif

100   Format(f7.2,x,f10.0,x,f6.0,x,f9.3,x,f15.3,x,f11.3)

      return
      end

      real function KT1(J)
      real J
c 1457
            KT1 = -0.7475*J**3 + 1.9283*J**2 - 2.0641*J + 0.9681
c 1462
c           KT1 = -0.0475*J**3 + 0.2747*J**2 - 0.834*J + 0.6734
      return
      end

      real function KT2(J)
      real J
c 2.0
            KT2 = 0.0328*J**3 - 0.0108*J**2 - 0.5155*J + 0.7327
c 2352
c           KT2 = -0.0291*J**3 + 0.1371*J**2 - 0.493*J + 0.8437
c 2.1/3
c           KT2 = 0.1152*J**3 - 0.4397*J**2 + 0.0464*J + 0.6971
c 1455
c           KT2 = 0.4103*J**3 - 1.0053*J**2 + 0.3261*J + 0.3703
c 1755
c           KT2 = 0.0155*J**3 - 0.0213*J**2 - 0.3528*J + 0.4955
c 2055
c           KT2 = -0.0092*J**3 + 0.0437*J**2 - 0.4532*J + 0.7347
c Z55
c           KT2 = 0.1486*J**3 - 0.2762*J**2 - 0.2836*J + 0.5001
      return
      end

      real function KQ1(J)
      real J
c 1457
            KQ1 = (-0.9236*J**3 + 2.4553*J**2 - 3.0937*J + 1.7767)/10
c 1462
c           KQ1 = (0.1821*J**3 - 0.3066*J**2 - 0.7631*J + 1.0905)/10
      return
      end

      real function KQ2(J)
      real J
```

```
c 2.0
              KQ2 = (0.2391*J**3 - 0.6129*J**2 - 0.7371*J + 1.8599)/10
c 2352
c             KQ2 = (0.0015*J**3 - 0.0824*J**2 - 0.4723*J + 2.4157)/10
c 2.1/3
c             KQ2 = (0.4168*J**3 - 1.7006*J**2 + 0.9717*J + 1.6318)/10
c 1455
c             KQ2 = (1.2519*J**3 - 3.1696*J**2 + 1.6308*J + 0.5714)/10
c 1755
c             KQ2 = (0.088*J**3 - 0.0844*J**2 - 0.803*J + 1.3008)/10
c 2055
c             KQ2 = (0.0052*J**3 - 0.0157*J**2 - 1.1393*J + 2.19)/10
c Z55
c             KQ2 = (0.4171*J**3 - 0.9068*J**2 - 0.2116*J + 0.9436)/10
      return
      end
```

The RPM optimizing program – CR RPM Optimizer.for

```
C This program optimizes for the RPM1 and RPM2 that will produce zero
C torque given a desired thrust and flow speed. The diameters and single
C propeller curves (as functions of J) for both propellers are needed.
C
C The single propeller equations should be put into the four functions at
C the end of the program. KT1 and KQ1 are the KT and 10KQ curves of the
C front propeller, respectively. KT2 and KQ2 are the KT and 10KQ curves of
C the aft propeller, respectively. --J Jacobson 4/05/07
C
C**************************************************************************
C*                                                                        *
C*                            CR RPM Optimizer                            *
C*                                                                        *
C**************************************************************************
C
C
      COMMON/PR/D1,D2,V,flag,Tdes,ITmax
      REAL D1,D2,V,Tdes
      INTEGER flag,ITmax

      Dimension X(2),XL(2),XU(2),G(2),WK(900),IWK(250),RPRM(20),IPRM(20)
      NRWK=900
      NRIWK=250
      Do 1 I=1,20
      RPRM(i)=0
1     IPRM(i)=0
      Method = 1
      NDV = 2
      NCON = 0

      WRITE(*,*)'Please Enter Description of Run'
      READ(*,'(a80)') description
      write(*,*)

      WRITE(*,*)'Please Enter D1 and D2 (in)'
      READ(*,*)D1,D2
      WRITE(*,*)

      WRITE(*,*)'Use the default wake fraction value (y/n)?'
      WRITE(*,*)'(Default value for cylindrical body 3 ft long with 3 in
      1 diameter)'
```

```
      READ(*,'(A1)')ans
      WRITE(*,*)

      if (ans.eq.'y'.or.ans.eq.'Y') goto 2

      if (ans.eq.'n'.or.ans.eq.'N') then
      WRITE(*,*)'Please Enter the Velocity (ft/s)'
      WRITE(*,*)'Velocity = V*(1-w)'
      READ(*,*)V
      WRITE(*,*)
      flag = 1
      goto 3
      endif

2     WRITE(*,*)'Please Enter V (ft/s)'
      READ(*,*) V
      WRITE(*,*)

3     WRITE(*,*)'Please Enter the Desired Thrust (lbs)'
      READ(*,*)Tdes
      WRITE(*,*)

      RPM1 = 5000
      RPM2 = 5000
C  DV1 RPM1
      X(1)=RPM1
      XL(1)=0
      XU(1)=10000
C  DV2 RPM2
      X(2)=RPM2
      XL(2)=0
      XU(2)=10000

      ITmax = 200

      Iprint = 0
      MinMax = -1
      INFO = 0
      RPRM(3) = 0.000000000000000000001
      RPRM(4) = 0.000000000000000000001
      IPRM(4) = 5
      IPRM(9) = 5

9     Do 20 Idot=1,5
10    Call Dot(INFO,Method,Iprint,NDV,NCON,X,XL,XU,Diff,MinMax,G,RPRM,
      1IPRM,WK,NRWK,IWK,NRIWK)
      If(INFO.eq.0)go to 20
      CALL DatDiff(X(1),X(2),Diff,0)
      Go to 10
20    continue
      CALL DatDiff(X(1),X(2),Diff,1)
      WRITE(*,*)'Another point (y/n)?'
      READ(*,'(A1)')ans

      if (ans.eq.'y'.or.ans.eq.'Y') then
      write(*,*)

      if (flag.eq.1) then
      WRITE(*,*)'Please Enter D1, D2, V*(1-w), Tdes'
      endif

      if (flag.eq.0) then
      WRITE(*,*)'Please Enter D1, D2, V, Tdes'
```

```
      endif

      READ(*,*)D1,D2,V,Tdes
      WRITE(*,*)
      go to 9
      endif

      WRITE(*,*)

      Diff = 0

      END

      SUBROUTINE DatDiff(RPM1,RPM2,Diff,IP)
      COMMON/PR/D1,D2,V,flag,Tdes,ITmax
      REAL D1,D2,V,Tdes,dens,mw,bw,avf1,mavf2,bavf2,rpm2f,tf2,qf2,
      1a2guess,Va1,w,avf2,J1,KT1,T1,KQ1,Q1,CT1,Etai1,a1,RPM2corr,Va2,
       2J2,KT2,T2,KQ2,Q2,CT2,Etai2,a2,Ttot,Qtot,Toff,Qoff,Eta
      INTEGER flag,ITmax

      Toff = 0
      Qoff = 0

      dens=1.94

      mw = 0.0129122
      bw = -0.0231159
      avf1 = 1.00019
      mavf2 = -0.0468778
      bavf2 = 0.828232
      rpm2f = 0.243667
      tf2 = 1.18446
      qf2 = 1.07092


      a2guess = 0.1

      do 12 j=1,ITmax

      if (flag.eq.1) then
            Va1 = V*(1+avf2*a2guess)
      endif

      if (flag.eq.0) then
            w = mw*V + bw
            avf2 = mavf2*V + bavf2

            Va1 = V*(1-w)*(1+avf2*a2guess)
      endif

      J1=Va1/(RPM1/60.)/D1

      T1=KT1(J1)*dens*(RPM1/60)**2*D1**4
      Q1=KQ1(J1)*dens*(RPM1/60)**2*D1**5*12

      CT1=T1/0.5/dens/Va1**2/3.1415926/(D1/2)**2
      Etai1=2/(1+(CT1+1)**0.5)
      a1=1/Etai1-1

      RPM2corr=RPM2+Va1*a1*rpm2f/(0.7*D1/2)*J1*(2*3.1415926)**2*120/100

      Va2=Va1*(1+avf1*a1)
      J2=Va2/(RPM2corr/60.*D2)
```

```
      T2=Tf2*KT2(J2)*dens*(RPM2corr/60)**2*D2**4
      Q2=Qf2*KQ2(J2)*dens*(RPM2corr/60)**2*D2**5*12

      CT2=T2/0.5/dens/Va2**2/3.1415926/(D2/2)**2
      Etai2=2/(1+(CT2+1)**0.5)
      a2=1/Etai2-1

      if ((abs(a2-a2guess)/a2).lt.0.001) go to 12

11    a2guess = a2

      if (j.eq.ITmax) then
            write(*,*)'a2 not converging'
            stop
      endif

12    Ttot = T1 + T2
      Qtot = Q1 - Q2

      Eta = Ttot*V/(2*3.1415926*(RPM1/60)*Q1/12 + 2*3.1415926*(RPM2/60)
      1*Q2/12)

      Toff = Toff + (Tdes - Ttot)**2
      Qoff = Qoff + Qtot**2

      Diff = Toff + Qoff

      if (ip.eq.1) then
      write(*,*)'Vel(ft/s)   Thrust(lbs)   Torque(ft-lbs)   RPM1   RPM2
      1    Eta'
      write(*,100)V,Ttot,Qtot,RPM1,RPM2,Eta
      write(*,*)
      endif

100   Format(f8.2,x,f12.3,x,f14.3,x,f12.0,x,f6.0,x,f6.3,x)

      return
      end


      real function KT1(J)
      real J
c 1457
            KT1 = -0.7475*J**3 + 1.9283*J**2 - 2.0641*J + 0.9681
c 1462
c           KT1 = -0.0475*J**3 + 0.2747*J**2 - 0.834*J + 0.6734
      return
      end

      real function KT2(J)
      real J
c 2.0
            KT2 = 0.0328*J**3 - 0.0108*J**2 - 0.5155*J + 0.7327
c 2352
c           KT2 = -0.0291*J**3 + 0.1371*J**2 - 0.493*J + 0.8437
c 2.1/3
c           KT2 = 0.1152*J**3 - 0.4397*J**2 + 0.0464*J + 0.6971
c 1455
c           KT2 = 0.4103*J**3 - 1.0053*J**2 + 0.3261*J + 0.3703
c 1755
c           KT2 = 0.0155*J**3 - 0.0213*J**2 - 0.3528*J + 0.4955
c 2055
```

```
c             KT2 = -0.0092*J**3 + 0.0437*J**2 - 0.4532*J + 0.7347
c Z55
c             KT2 = 0.1486*J**3 - 0.2762*J**2 - 0.2836*J + 0.5001
      return
      end

      real function KQ1(J)
      real J
c 1457
             KQ1 = (-0.9236*J**3 + 2.4553*J**2 - 3.0937*J + 1.7767)/10
c 1462
c             KQ1 = (0.1821*J**3 - 0.3066*J**2 - 0.7631*J + 1.0905)/10
      return
      end

      real function KQ2(J)
      real J
c 2.0
             KQ2 = (0.2391*J**3 - 0.6129*J**2 - 0.7371*J + 1.8599)/10
c 2352
c             KQ2 = (0.0015*J**3 - 0.0824*J**2 - 0.4723*J + 2.4157)/10
c 2.1/3
c             KQ2 = (0.4168*J**3 - 1.7006*J**2 + 0.9717*J + 1.6318)/10
c 1455
c             KQ2 = (1.2519*J**3 - 3.1696*J**2 + 1.6308*J + 0.5714)/10
c 1755
c             KQ2 = (0.088*J**3 - 0.0844*J**2 - 0.803*J + 1.3008)/10
c 2055
c             KQ2 = (0.0052*J**3 - 0.0157*J**2 - 1.1393*J + 2.19)/10
c Z55
c             KQ2 = (0.4171*J**3 - 0.9068*J**2 - 0.2116*J + 0.9436)/10
      return
      end
```