Holistic Concepts for Fixed and Rotary Wing Aircraft Engines

Shyam V., Poinsatte P., Culley D., Snyder C., Kulkarni S., Schilling H., Wroblewski A., Zelek M.
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Ameri A.
The Ohio State University, Columbus, OH 43210

Thurman D.
U.S. Army Research Laboratory
Glenn Research Center
Cleveland, Ohio 44135

Raghu S.
Advanced Fluidics LLC.
Seattle, Washington

Eichele P.
Gilcrest Electric & Supply Company
Cleveland, Ohio 44135

Abstract

The purpose of the work presented in this report is to evaluate the feasibility of a holistic concept for turbine energy extraction by combining the benefits of bio-inspired concepts and flow control concepts that have until now only been studied in isolation. The term holistic is used in the sense that the entire engine and even the airframe may benefit from the concepts explored here. In addition, applications to particular parts of the engine such as the low pressure turbine, involve management of the secondary flows, endwall flows and blade performance and acoustics simultaneously. Two primary concepts are studied: **1.** Seal Vibrissae inspired airfoil profiles are studied at a Reynolds number of 100,000 based on inlet conditions and blade axial chord at various incidence angles and for various airfoil treatments. **2.** An ACFC (Autonomous Closed-Loop Flow Control) scheme was devised and a feasibility study of the system was conducted. There are also applications to non-aerospace areas. The Seal treated airfoils show drag reduction at all incidence angles compared to modern turbomachinery airfoil geometry while not sacrificing lift. The results of this work show that a fuel burn reduction of approximately 5% can be achieved while reducing noise. Fuel burn reduction is accomplished by improving Low Pressure Turbine (LPT) performance at cruise and enabling incidence tolerance for the turbine blade geometry. Weight reduction is also a contributor to the fuel burn reduction and is achieved through endwall flow management that increases the spanwise lift distribution. Noise reduction is achieved by spreading the energy in the wake to create many small peaks rather than one large peak at blade passing frequency and shear layer modes. The results are applicable to compressors, inlets, fans and even external aerodynamic features such as wings, landing gear and struts. It was found that the biomimetic 'Seal Blade' is relatively insensitive to incidence angle, reduces or eliminates flow separation and has the potential to reduce noise due to the modified wake spectrum. The components of the autonomous closed-loop flow

control system were assembled on a benchtop test and found to function as conceptually predicted. Simulations of boundary layer suction to provide flow for flow control showed an increased loading near the endwall and marked reduction in size of the horseshoe vortex. Trailing edge pulsed ejection was shown to fill the momentum deficit in the wake and simultaneously alter the acoustic signature of the wake to reduce broadband noise. The results presented herein are preliminary and further analysis and data collection is in progress.

Background

The Fixed Wing and Rotary Wing projects under the Fundamental Aeronautics Program face several challenges over the coming decades. The Fixed Wing program for example, aims to reduce fuel burn by 30% by 2015 and by 60% by 2025 simultaneously reducing noise by 32db and 71db respectively.
For the Rotary Wing project, the Variable Speed Power Turbine (VSPT) is a viable solution for vertical take-off and landing (VTOL) and high-speed cruise rotorcraft [1, 2]. It avoids the complexity and weight of variable transmission that is used on fixed speed power turbines. A key challenge for the VSPT is to maintain high turbine efficiency over main rotor speed variations from 54% to 100% speed to minimize overall fuel burn. The power turbine must therefore operate at a large range of incidences (60 degrees or greater,) and in low Reynolds number regimes.
For Fixed Wing aircraft, current low pressure turbine (LPT) designs are unable to maintain aerodynamic performance at altitude cruise conditions where the Reynolds number has dropped substantially from the sea level value [3]. These low Reynolds number issues prompted NASA to host a large international LPT aerodynamics workshop in 2010. Currently, boundary layer separation at low Reynolds number is either accepted, or the blading is designed for reduced loading to avoid the issue but penalizes the engine with additional LPT weight (and possibly reduced Sea Level Static performance). Figure 1 shows some of the problems experienced in a turbine stage. Figure 1 (left) shows the formation of the horseshoe vortex [1] that migrates into the passage and interacts with the passage vortex leading to significant losses. At the tip, leakage flows also interact with passage flows and cause aerodynamic loss. The loss can be assessed by measuring the total pressure loss coefficient in the wake.

LPT efficiency is known from system studies to have the strongest impact on engine specific fuel consumption reduction and is the heaviest component in the engine. Hollow LPT blades have been envisioned as a way to reduce engine weight. Reducing blade weight as well as reducing the cruise performance loss would doubly benefit future engine designs which are attempting to reduce weight by increasing the per stage work and thereby reducing stage count. Additional areas of improvement for the gas turbine and for propulsors in general are the fan blades for fixed wing aircraft or rotors and tails for rotorcraft.

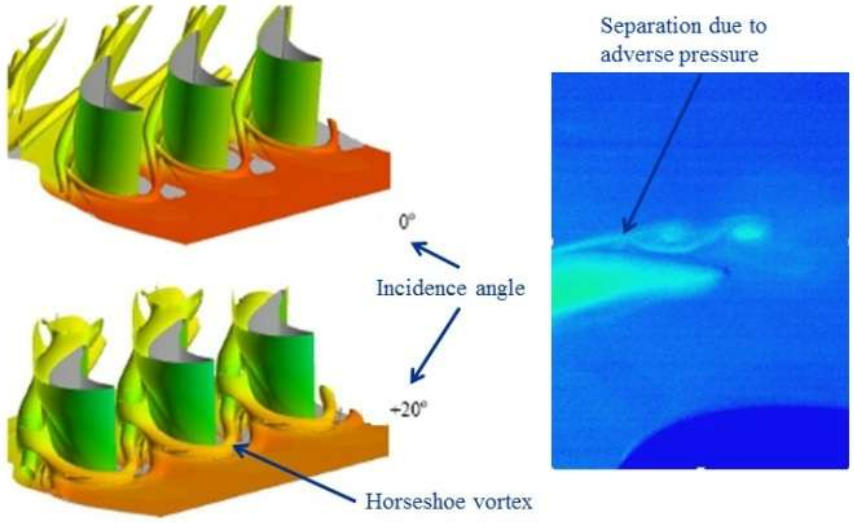


Figure 1. Effect of incidence change from CFD (left) and low Reynolds number from water table infrared measurement (right).

v2013.1

TECHNOLOGY BENEFITS*	TECHNOLOGY GENERATIONS (Technology Readiness Level = 4-6)		
	N+1 (2015)	N+2 (2020**)	N+3 (2025)
Noise (cum margin rel. to Stage 4)	-32 dB	-42 dB	-52 dB
LTO NOx Emissions (rel. to CAEP 6)	-60%	-75%	-80%
Cruise NOx Emissions (rel. to 2005 best in class)	-55%	-70%	-80%
Aircraft Fuel/Energy Consumption‡ (rel. to 2005 best in class)	-33%	-50%	-60%

* Projected benefits once technologies are matured and implemented by industry. Benefits vary by vehicle size and mission. N+1 and N+3 values are referenced to a 737-800 with CFM56-7B engines, N+2 values are referenced to a 777-200 with GE90 engines
** ERA's time-phased approach includes advancing "long-pole" technologies to TRL 6 by 2015
‡ CO2 emission benefits dependent on life-cycle CO2e per MJ for fuel and/or energy source used

Figure 2. NASA Fixed Wing System metrics for N+1 through N+3 generations.

The results of this work aim to simultaneously enable fuel burn reduction, noise reduction and NOx reduction by 1) Biomimetic geometric features such as Seal Vibrisae to sustain high performance over a wide range of incidences, and in low Reynolds number regimes, 2) An Autononomous Closed-Loop Flow Control (ACFC) system to self-regulate the flow to and from the suction and pressure side using internal circuits of fluidic diverters in a thick hollow blade and 3) manipulating the wake in a pulsed mode to reduce noise and to reduce aerodynamic losses. Flow for control can be suctioned through an under rotor tunnel or from strategic locations on the blade surface in the case of split-blade geometries.

A comprehensive literature review of biomimetic aerodynamic and acoustic applications has been conducted that includes a study of owls, harbor seals, whales, sharks, dragonflies, cacti and palm trees. In addition, flow control literature was also reviewed including separation control through pulsed jets,

plasma actuators, and passive methods such as ramps and steady blowing/suction trailing edge serrations, trailing edge blowing and tandem airfoil concepts.

Flow control has been investigated in the past on low pressure turbines and compressors using synthetic jets, vortex generating jets and dielectric barrier discharge plasma actuators [4-9]. Grooves and ribs on blades have also been investigated. For the trailing edge, chevrons and other forms of serrations have been attempted on fan blades but these are sensitive to local flow conditions. Halasz [10] investigated the use of steady trailing edge blowing for fan blades by unevenly distributing the ejection slots among the fan blades of a rotor/stator configuration. Figure 3 shows the sources of noise from a turbofan engine [10]. The fan noise and noise from the aft turbine components are the dominant sources that contribute to noise that is considered undesirable to humans. Halasz [10] found that the uneven distribution changed the spectral shape of the wake

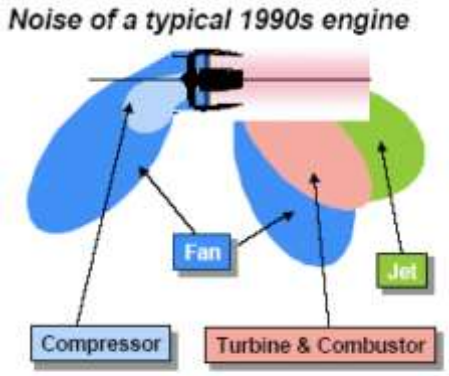


Figure 1.3: Turbofan Engine Noise Sources.

Figure 3. Noise sources form a turbofan engine. [10]

For external flow, whale tubercles [11-15] inspired wind turbine blades have had great success. These blades use the bumps on the leading edge to delay stall. The creation of downstream vorticity due to the channeling of flow between tubercles could also be detrimental to downstream components. Also, no reduction in drag is found at zero incidence. Similar conclusions are drawn for studies using sinusoidal leading edges [16]. In nature, dragonfly wings have serrations to mitigate separation at low Reynolds numbers [17-19]. For application to the low pressure turbine, one must ensure that the flow is laminar for this modification to work and the scale of serrations on the blade must be on the order of magnitude of blade thickness. Thus, unless a drag penalty is acceptable, such serrations are unlikely to be implemented on high Reynolds number turbomachinery components. Cactus trees [20-22] and palm trees [22, 23] are optimized by nature to reduce drag and therefore shear stresses on their trunks. Unfortunately one of the factors in their optimization is variable wind direction which results in a serrated circular trunk cross-section with no net lift force. Chevron-shaped protruberances on Sailfish skin [24, 25] possibly serve to keep flow attached during maneuvering at high speeds. This results in larger drag and is therefore not beneficial to propulsion applications except where it might serve to reduce separation and thus offset the drag penalty. In these cases, the chevrons would be highly sensitive to relative flow direction and Reynolds number and would only be suited for a particular design.

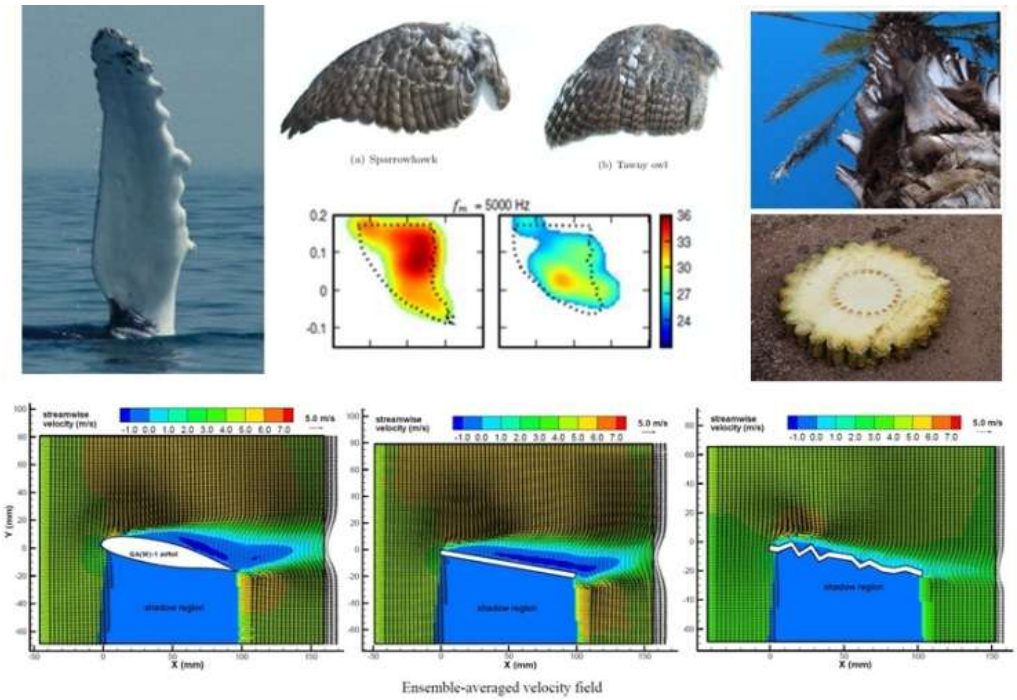


Figure 4. Some of Nature's answers to separation and noise. [11-26]

Figure 4 shows some biological features that have inspired solutions to aerodynamic problems. Owl wings [26] are shown (figure 4 top and center) here as an excellent acoustic damper. Figure 5 shows some of the regions of a hypothetical engine that can benefit from improvements in active and passive flow control both for fuel burn reduction and for noise reduction. The work in this report can benefit the areas labelled in blue.

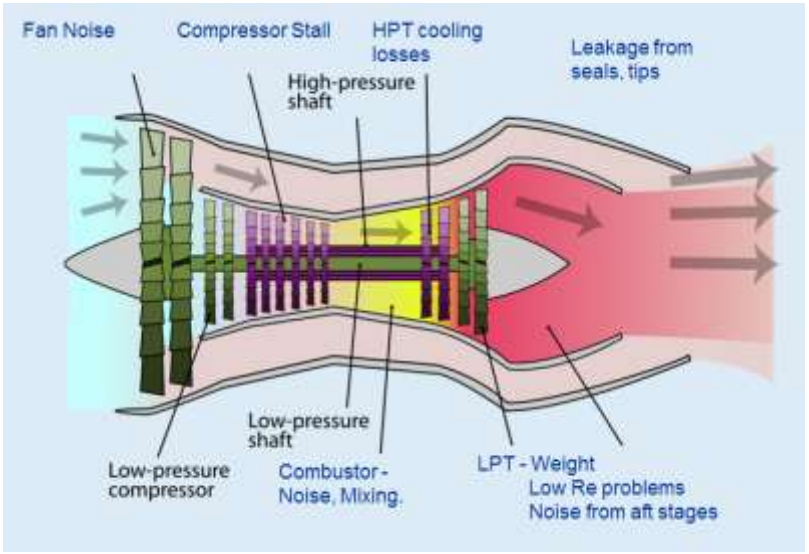


Figure 5. Areas of engine that could benefit from this work (blue).

Conceptual Approach

Biomimetics – Seal Vibrissae

The implementation of biomimicry on a turbine blade or a fan blade (see figure 5) is very different to that on a low speed wind turbine or even on the wing of an airplane. For example, whale tubercles [11-15] have been shown to improve incidence tolerance on average over a wide range of incidence angles by making the leading edge region more insensitive to incidence. This is accomplished by drawing the flow into valleys and the formation of counter rotating vortical structures. On a turbine blade, due to the high loading, range of incidence angles and large camber, separation can occur both near the crown region and along the suction side. Separated regions also exit on the pressure side for negative incidence. The wake from these internal flows has an impact on downstream blade rows and must be managed for both performance and acoustic reasons.

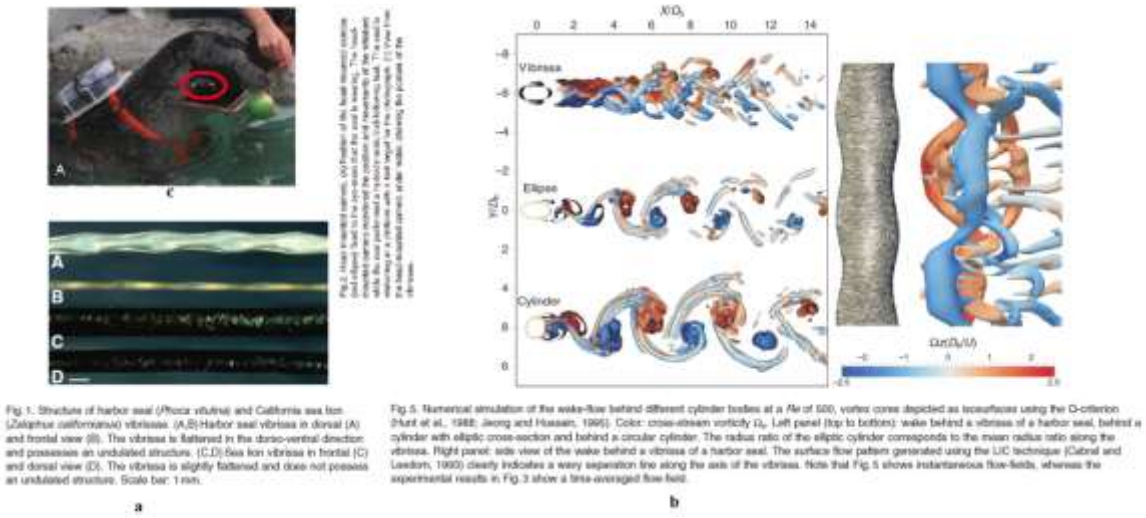


Figure 6. Figures from Hanke et al. [28] showing a) Whiskers of Seal (A and B) and Sea Lion (C and D) b)Vortex shedding from seal (top,) ellipse (center) and cylinder (bottom.) and c) an instrumented harbor seal. (Hanke et al. [28]).

Seal vibrissae allow the seal to detect slight disturbances in the water upstream of them by pushing the vortices from their whiskers downstream and away from them [27, 28]. It was noticed by Hanke et al. [28] that the unsteady forces on the seal vibrissae were reduced by 90% while the drag was reduced by 40%. This is markedly different from other sinusoidal treatments as noted by Hanke et al.[28] in that most sinusoidal treatments such as the Scruton helix, or sinusoidal leading edges on blades are unable to impact drag in a positive manner while maintaining incidence tolerance or reducing vortex induced vibrations (VIV). It appears that if a similar approach was used on a turbine blade we can achieve a thinning of the wake and a reduction or elimination of separation.

Observations of the seal whisker prompted the incorporation of undulations for an entire blade such that the leading edge undulations match the undulations of the seal vibrissae. The 3D Unsteady RANS code Glenn-HT was used to simulate the baseline Rolls Royce ‘incidence tolerant’ power turbine blade (airfoil section shown in figure 7 – right [1]) that was tested in NASA Glenn’s CW-22 facility at angles of attack of negative incidence, 0 degrees and high positive incidence. This blade showed separation at negative and at high positive incidence angles. Next, the blade profile was swept through profiles of varying pitch and amplitude. Figure 7 shows the parameters of interest for the present study. The pitch, P and amplitude, a, of the undulations were varied to match the pitch to amplitude ratio and pitch to leading edge radius of the seal vibrissae [27]. Table 1 shows the range of pitches and amplitudes

explored in this study. The highlighted cells contain parameters that are the same for the seal treated blades as they are for the seal vibrissa.

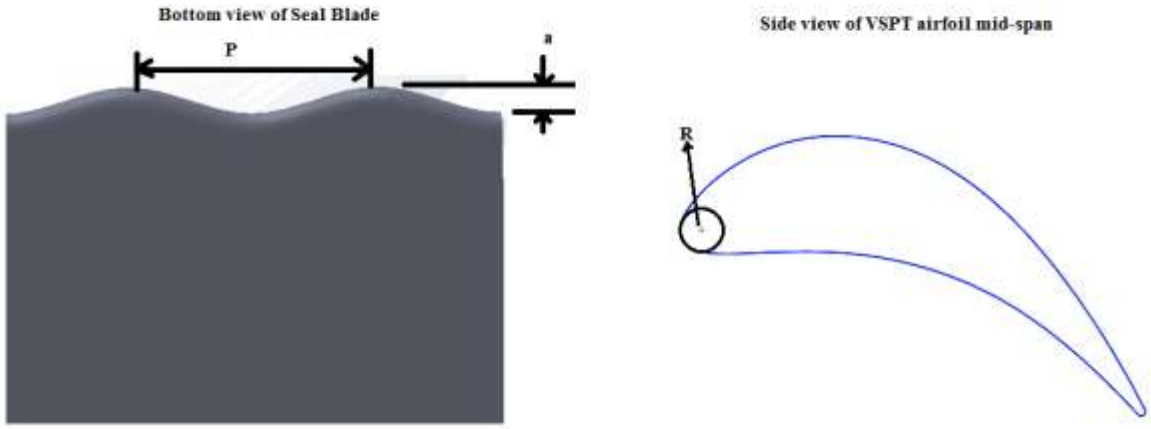


Figure 7. Parameters of undulations for the 'Seal Blade'.

Table 1. Parameters of undulations for the VSPT mid-span airfoil section [1].

Parameters	Seal	S1R0P5	S1R1P0	S2R0P5	S2R1P0
Reference Radius, R	0.54	0.46	0.46	0.46	0.46
Peak to Peak, P	1.82	1.55	1.55	3.00	3.00
LE amplitude, a	0.12	0.12	0.23	0.12	0.23
ratio R/P	0.29	0.30	0.30	0.15	0.15
ratio a/P	0.07	0.07	0.15	0.04	0.08

The seal vibrissae are comprised of elliptical cross sections that are inclined at 15-17° to the flow direction. This angle was not included in the present study as it is unclear whether or not the angle is due to the flexibility of the whisker or due to the flexibility of the skin. It was intended to add the inclination in the event that the geometry without the inclination did not show any significant changes from the baseline geometry. This was not necessary in the present study and due to the limited time and budget, variation of Reynolds number, Mach number, undulation inclination and endwall flows will be studied in future work. The pressure side is largely unaffected and does not suffer from undulations that would cause a reduction in lift. This preserves the 2D airfoil profile for simplified design. Figure 8 shows the rationale for this approach. Flow approaching Peak 1 encounters the blade first and sees a spherical surface. The 3D relief accelerates the flow and crown separation if it exists would be pushed further aft. The peak downstream of Peak 1 experiences the lowest pressure of any point along cut 1. A pressure gradient exists between Peak 3 and Peak 1 such that flow downstream of Peak 1 moves toward Peak 3 and continues along the peak that is formed downstream of Peak 3. This is a high momentum stream and is thus less prone to separation. The flow in the adjacent valleys however is lower momentum flow but is energized by the span-wise pressure gradient. This causes a delayed separation or a mitigation of separation. The advantage of this scheme would be that separation or severe adverse pressure gradient regions would be confined to valleys along the trailing edge of the blade. Active flow control can then be used sparingly in these regions. The distortion of the wake would also lead to a modified acoustic spectrum that could lead to noise reduction.

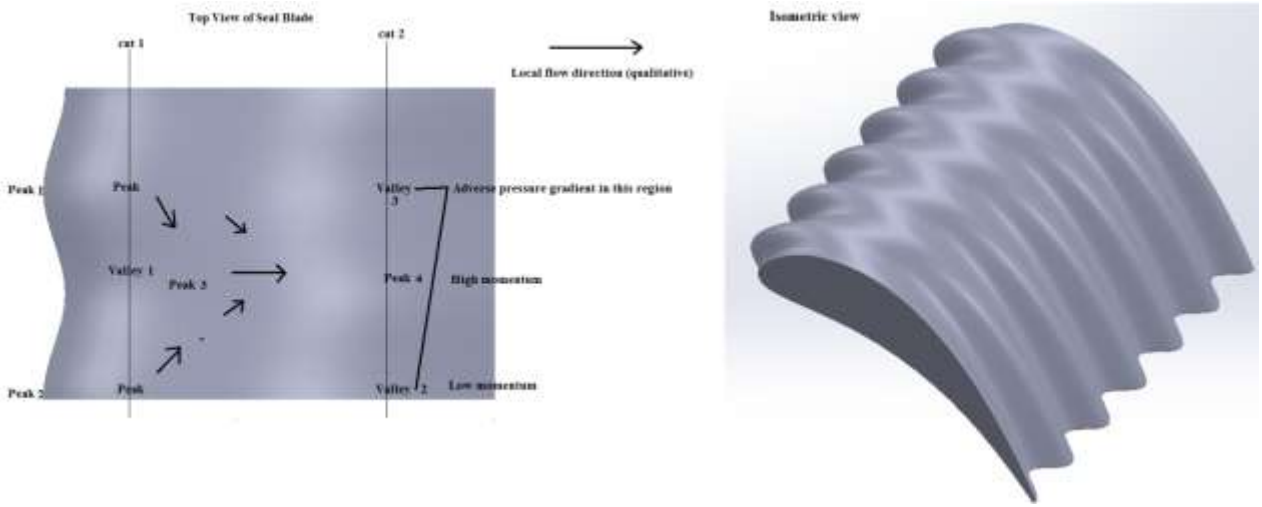


Figure 8 Seal blade concept showing undulations on suction side and on leading edge.

Autononomous Closed-Loop Flow Control (ACFC) system

It is expected that even if fully successful, the secondary flows (figure 1) will still contribute to a significant loss and must therefore be managed as part of a holistic approach. In addition, separation on low pressure turbines with more aggressive loadings would still need to be managed. Noise due to rotor stator interaction would also benefit from wake management. Fan noise is a major contributor to noise pollution and it would be beneficial to find ways to manage the wake with less energy.

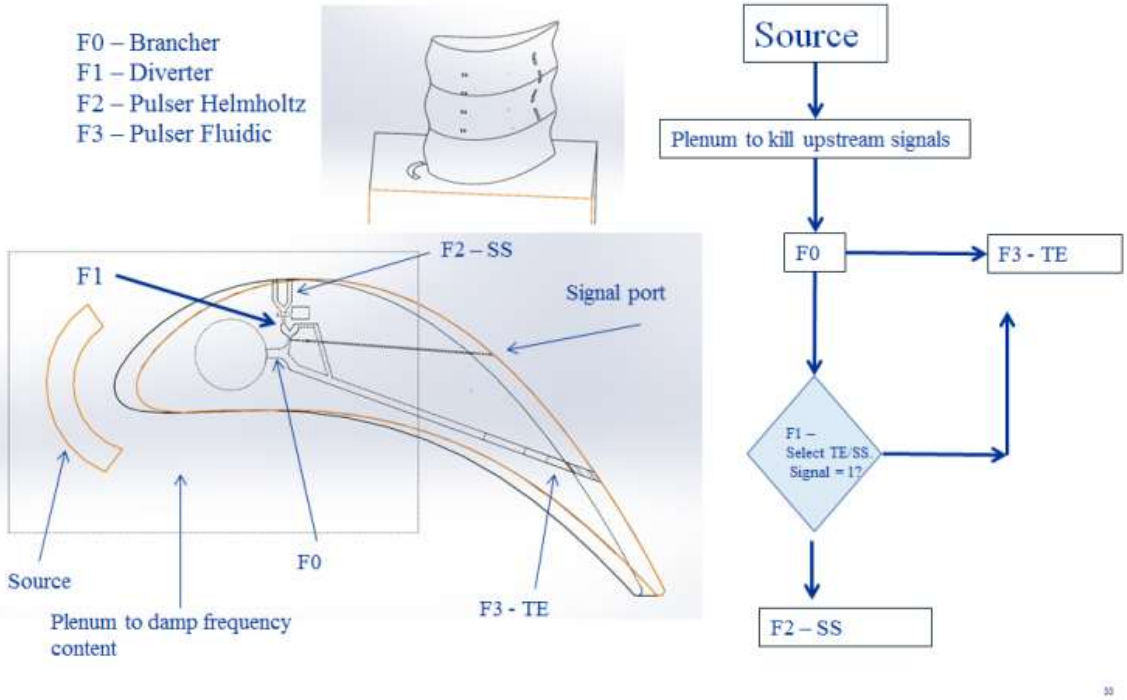


Figure 9. Concept diagram for ACFC (Autonomous Closed-Loop Flow Control.)

Figure 9 shows a conceptual diagram for the ACFC system. The system is comprised of 3 main components.

1. It comprises a source for flow control. This would be either a location upstream of a turbine blade or a location downstream of a control point for a compressor or fan. In this embodiment, the source flow is obtained from the hub boundary layer and fed through the fluidic network system. The objective of this work was to simulate the effect that such boundary layer suction would have on the wake of a representative turbine blade. While boundary layer suction or blowing have been attempted before [29] the suction or blowing have been located in the passage region on the endwall after the formation of the horseshoe vortex. The suction or blowing strength is also allowed to vary. In this study, the suction strength is determined by the pressure ratio across the suction slot location and the blade surface location where flow control is used. The slot is then sized to allow sufficient mass flow to provide a blowing ratio [4-9] that has been shown to be effective in controlling separation. The location of the suction slot is upstream of the horseshoe vortex saddle point to enable maximum effect on the horseshoe vortex. This location was determined using computational fluid dynamic simulation of the flow for multiple incidence angles.

2. The fluidic network consists of a series of fluidic diverters and actuators. Figure 10 shows the functioning and schematic of one such fluidic diverter labelled F1 (also shown in figure 9). The diverter controls the direction of the flow exiting a particular fluidic diverter via signals received from control ports that are connected to strategically located ports. These ports are similar to pressure ports that could be located in the turbine flowpath. For example, control port 1 could be located on the suction surface of a low pressure turbine blade, control port 2 could be located at a reference location, Exit 1 diverts flow to the trailing edge actuator, F3 and Exit 2 diverts flow to the actuator F2 on the suction side. The diverters themselves have been extensively studied as fluidic switches When a separation occurs for example, a local high pressure region is detected, the lower pressure from the other port then causes the jet to switch to the side of lower pressure. In figure 9, source flow enters a plenum or a series of plena. The plena can either be used to damp out incoming frequency content or can be tailored to amplify the frequency content using F2 (Fluidic pulser using Helmholtz resonator that is shown below).

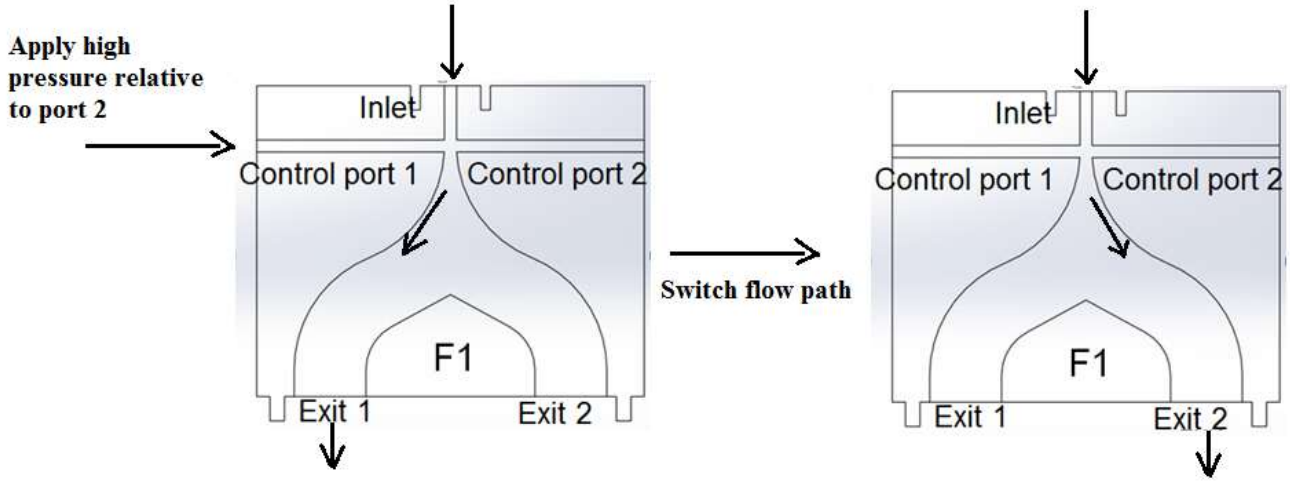


Figure 10. Functioning and schematic of a fluidic diverter.

3. Flow control actuators (F2 or F3 in figure 9) along the surface of the blade for managing pressure loading and at the trailing edge. The objective of trailing edge pulsed blowing is twofold. It should be effective in mixing out the blade wake, which should reduce tone noise generated by the interaction of

rotor wakes with downstream stators. It should also cause the breakdown of large turbulent structures in the wake into smaller ones with the attendant benefit of shifting the rotor-stator interaction broadband noise content from low to high frequencies. This is beneficial since high frequency noise is more easily absorbed by the atmosphere so there should be a net reduction in broadband noise. Actuator F2 is a vortex generating jet produced by a fluidic pulsing device that could either be F3 described below or a Helmholtz resonator driven fluidic pulsing device that is shown in figure 11. The Helmholtz resonator would be designed for a particular frequency based on the literature [9]. Miller et al. [9] found that Helmholtz resonators embedded in the blade might not have sufficient control authority to directly control the main gas path flow. However there might be enough control authority to provide a perturbation for the control channel of a fluidic device.

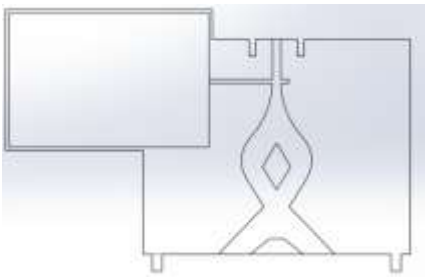


Figure 11. Schematic of a Helmholtz driven fluidic oscillator.

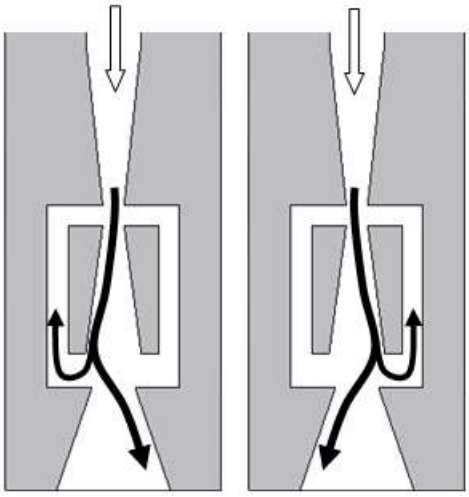


Figure 12. Schematic and functioning of a fluidic sweeping actuator.

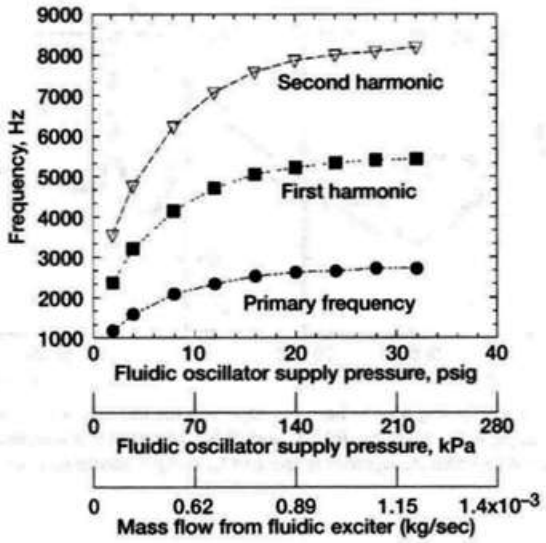


Figure 13. Frequency and mass flow characteristics of a sweeping jet. [30, 31]

Fluidic sweeping jet actuators (such as F3 in figure 9) with no moving parts are based on bi-stable states of a jet of fluid in a cavity caused by a specially designed feedback path as shown in the schematic of figure 12. A jet of fluid attaches to one of the two sides of a surface due to the "wall attachment"; commonly known as the "Coanda" effect. The pressure distribution in the cavity is accordingly changed and the feedback channel transmits this pressure differential back to the point of the jet separation thus deflecting the jet to the other side. This cycle is repeated on the other side of the cavity through the feedback channel on the right thus producing an oscillating jet at the exit of the cavity. A slight modification at the exit of the actuator (a splitter plate) is required to produce alternately pulsing jets at the exit instead of a single sweeping jet. Thus, these devices do not need external signals or actuation to produce oscillating jets. Frequencies from 1-10 kHz have been obtained with meso-scale (nozzle sizes in the range of 200 microns – 1mm) fluidic actuators with very low mass flow rates of the order of (10-3 Kg/sec) [30]. Figure 13 shows the frequency as a function of mass flow characteristics of a fluidic actuator (1.69 mm x 0.95 mm exit area) used by Raman et al [31] in their experiments on cavity noise control. Figure 14 shows the footprint of the sweeping jets in contrast to that of steady jets visualized on a liquid crystal surface.

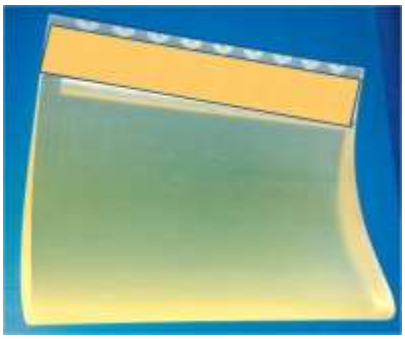


Figure 14. VSPT blade with trailing edge ejection oscillators.

Advanced Fluidics has recently developed actuator designs that can be integrally fabricated into airfoil sections using 3-D rapid prototyping technologies. An example of an integrated pulsing jet fluidic

actuator array fabricated for phase I experiments is shown in figure 14 with the proprietary designs of the actuators masked. Alternate methods of CNC milling in parts and assembly will also be explored if needed. Although the pulsing and sweeping jets can work even in supersonic regimes, in the present work, we plan to test these till sonic exit conditions.

Sweeping actuators have been used for separation control on multi-flap airfoils [32] and on compressor blades [33]. It has been shown by Culley et al. [33] that pulsed ejection at the trailing edge of a stator vane can reduce the separation on a compressor stage. Both solenoid driven oscillations and fluidic diverters were used for the study. Woszildo et al. [32] conducted a parametric study of sweeping jet fluidic actuators for a multi-flap airfoil. They found that actuator spacing, frequency and amplitude had a significant role in reducing separation.

Setup of Computations

Biomimetics – Seal Vibrissae

The numerical investigations were conducted using a NASA, in-house, Reynolds-Averaged Navier-Stokes code, Glenn-HT [33-35]. For the work done with seal-whisker inspired treatment of the blade a two-equation model, namely the SST- transition enabled two-equation model [34] was used. For the work on the endwall boundary layer suction a three-equation transitional turbulence model [35] was used. The geometry considered is the Rolls Royce Variable Speed Power Turbine Blade Cascade [1]. The cascade was deemed a suitable case for the initial testing of the concept due to its large thickness to chord ratio, design for power turbine application and due to the existence of separation. The setup of the models will be briefly described below. The cascade geometry cases shown in Table 1 were simulated. Surface meshes for 3 of the cases are shown in Figure 15. For these cases, the baseline grid had 7 million grid points per span-wise pitch and was generated as a multi-block grid with 567 blocks per period using GridPro. Cases were run with a once-coarsened grid with 1.04 million grid points per period after comparison of results from both grid resolutions for the baseline case.

Figure 15. A "clean" blade, S1R1P0, S2R0P5 whisker type of treatments from left to right.

ACFC – Boundary Layer Ingestion

Figure 16. VSPT blade, baseline geometry.

The endwall suction aspect of the ACFC was evaluated numerically using Glenn-HT using the VSPT geometry. The grid in figure 16, was generated to contain over 1.8 million grid points. The grid is clustered to the blade and to the endwall to finely resolve a thick boundary layer (20%span). The grid was set up such that four separate zones may be used for suction. For the present computations all four zones were used for suction when suction was on to simulate a blowing ratio of 2.0 for flow control ports. The zones were located using preliminary CFD computations of the baseline geometry at negative and positive angels of attack.

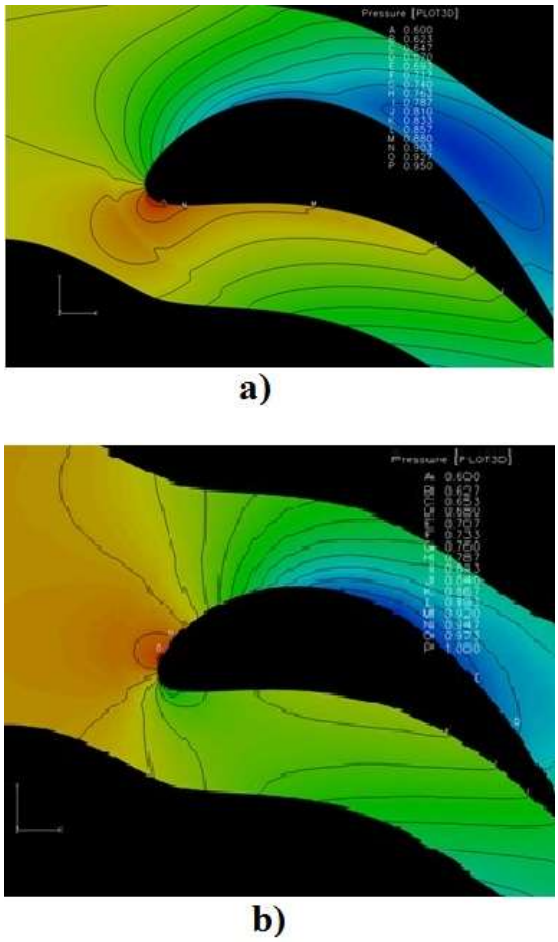


Figure 17. Hub pressure distribution for a) 10° incidence angle, b) -37° incidence angle

The pressure profile on the hub at negative incidence and high positive incidence were studied for the baseline Rolls Royce blade geometry. These profiles serve as the basis for the location of the under-rotor tunnel that draws in the boundary layer and ejects it into the wake and suction side in a pulsed mode. Figure 17a shows the pressure distribution on the hub at 10° incidence angle wile figure 17b shows the pressure distribution at -37° incidence. The positive incidence case was selected for study of suction to limit the number of cases. The effect on negative incidence should not differ markedly but will be studied in future work. Figure 18 shows the grid generated by GridPro on the hub of the VSPT geometry.

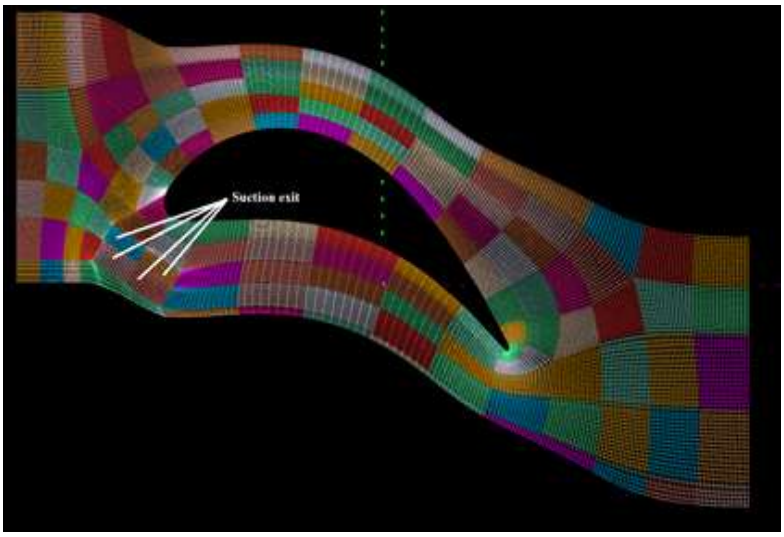


Figure 18. Grid for boundary layer ingestion upstream of baseline VSPT blade.

Setup of Experiments

SW-2 Cascade facility

The SW-2 facility test section shown in figure 19 (left) was redesigned in Phase 1 to accommodate testing of biomimetic and flow control test articles. The new test section shown in figure 19 (right) incorporates a modular tunnel and multiple slot locations for wake pressure, hotwire and thermocouple surveys. A modular lid allows optical access for IR or PIV. The inlet was modified to allow a wide range of incidence angles and the test section was modified to allow for blades with high turning angles. A removable ceiling plate allows for rapid exchange of test articles. Hotwire anemometry, total and static pressure and temperature surveys are made possible using a 3-axis actuator system. The system is controlled using LabView. The tunnel is to be operated in the Mach 0.1 – Mach 0.7 range. Thurman et al. [36] describe the operation of this facility.

Figure 19. SW-2 wind tunnel facility before modifications. [36]

Water table facility

A water table rig has been set up in test cell SE-1 at NASA GRC to enable both quantitative and qualitative analysis of new aeropropulsion concepts. A picture of the rig is shown in Figure 20. This is a low cost facility used to complement data taken in the SW-2 wind tunnel facility Tests on the water table rig visualize the complex flow structures associated with several biomimetic concepts.

Figure 20. Water table in test cell SE-1 at NASA GRC.

Figure 21 shows the VSPT baseline cascade used in this Phase 1 Seedling Fund work. The test section walls are made of acrylic glass with modular connections to enable changes in length and width of the test section. Gears are used to connect the inlet and exit to the test article to allow for a wide range of incidence angles (55° range). Baffles in the inlet water reservoir ensure steady inlet flow.
Sliding doors are used at the inlet to channel flow into the test section. This increases the range of flow rates through the test section. A weir is used downstream of the test section to control the water height and flow rate is controlled using a valve and flow meter. Temperature and depth measurements are available.

Figure 21. VSPT baseline cascade in test section of water table showing four blades.

A FLIR SC650 series infrared camera (figure 22) is used to visualize and quantify the flow. Heated commercial dye is injected through pores in the test article. This heated dye is easily recorded by the IR camera and provides a valuable tool for imaging flow separation and other flow features such as horseshoe vortices, tip leakage and film cooling. By recording multiple frames at high speed, it is possible to convert the thermal maps into velocity and pressure by following vortices captured by the image. The processing can be improved through the use of specialized optics to focus the camera on a plane and by using different fluids for injection.

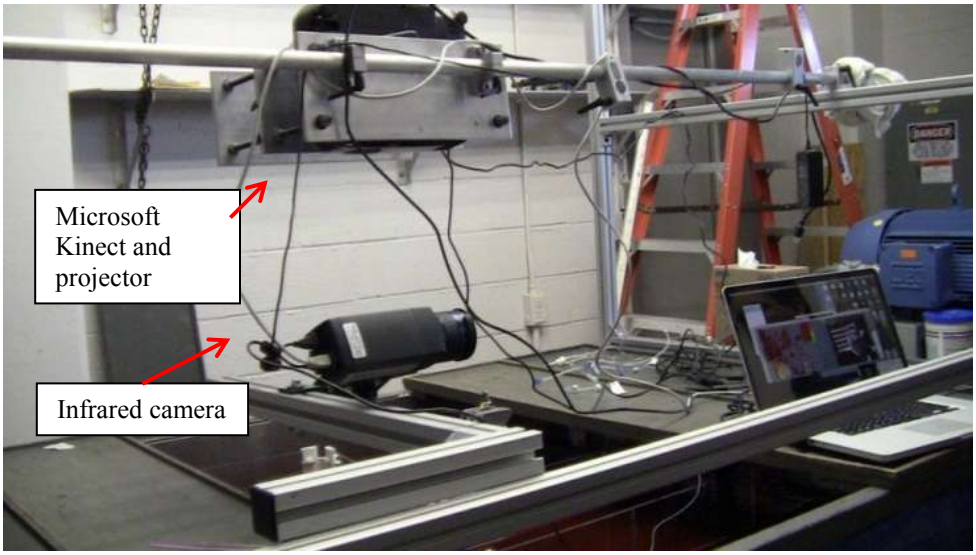


Figure 22. Microsoft Kinect, projection system, and infrared camera mounted above water table rig in SE-1.

A Microsoft Kinect is used to visualize the surface of the water flowing across the water table and through the test articles. The Kinect detects the distance of the water surface from its sensors. A computer processes this information to obtain real-time pressure ratios in the flowfield and simultaneously visualizes the flow by outputting the processed video to a projector. The projector displays the video back onto the water surface or onto another suitable viewing area. The location of the Kinect and the projection system is shown in figure 22.

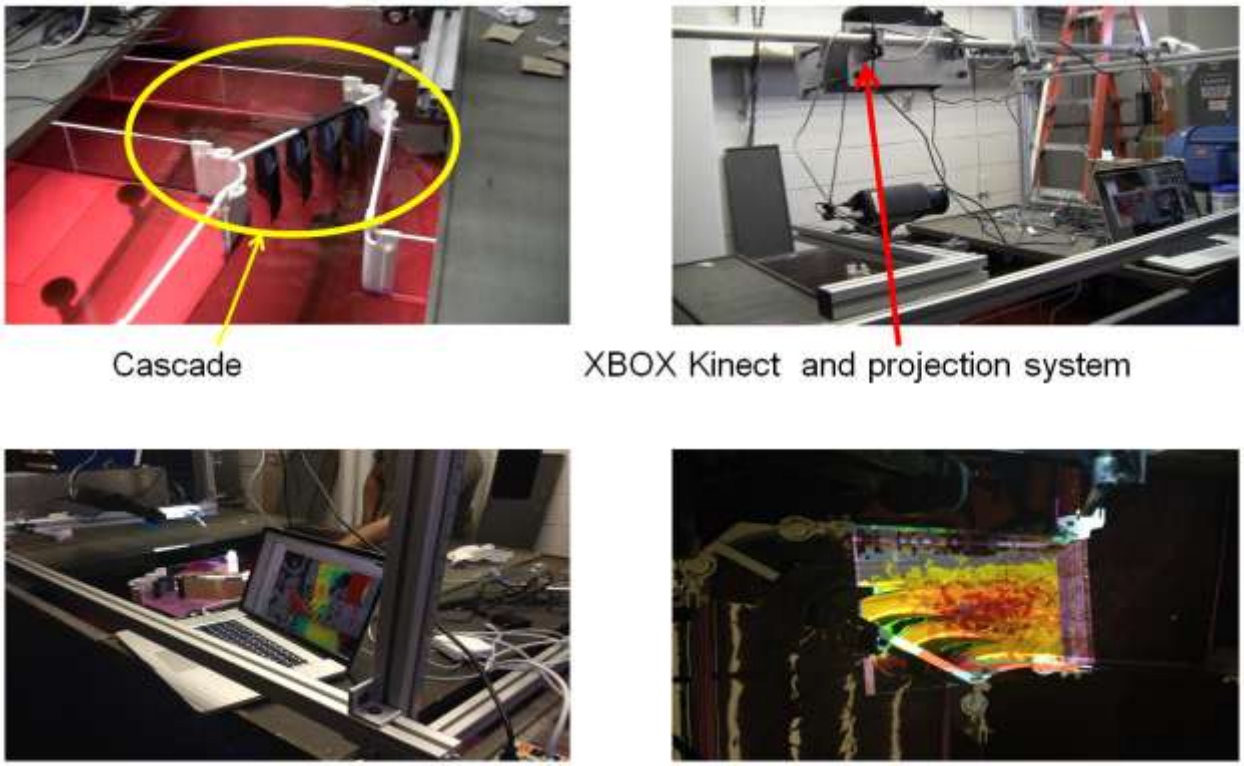


Figure 23. Prototype flow measurement using Microsoft Kinect V1.

Figure 23 shows the implementation of a Microsoft Kinect system for real-time flow measurements. Figure 23 (top left) shows the VSPT cascade in a low Reynolds number flow on the water table. Red commercial dye is used to make the water opaque to the infrared Kinect sensor. The Microsoft Kinect detects the water height and relays this information to a computer (shown at the bottom left of figure 23). The computer processes the height data into pressure ratio and plots pressure contours that are then displayed on the water surface using a projection system (shown at the bottom right of figure 22).

As part of the Phase 1 effort, an agreement was signed with Microsoft that enables NASA to use a pre-release version of Microsoft's Kinect V2 and the associated software development kit. This agreement was the result of a competitive award process. The Kinect V2 is currently being tested against standard depth measurement tools and with various surfactants and methods for achieving infrared opacity.

Biomimetics – Seal Vibrissae Results

The CFD was validated in a previous study [1] using the same baseline geometry. Figure 24 shows the pressure distribution around the mid-span airfoil section. Figure 24 shows that the numerical simulations (solid lines) match the data extremely well and more importantly show the separation location at approximately x/C of 0.8. Since this is a feasibility study, the objective of the work presented in this report is not to validate CFD or to obtain accurate flow data but to observe meaningful trends and to lay the foundation for more detailed studies.

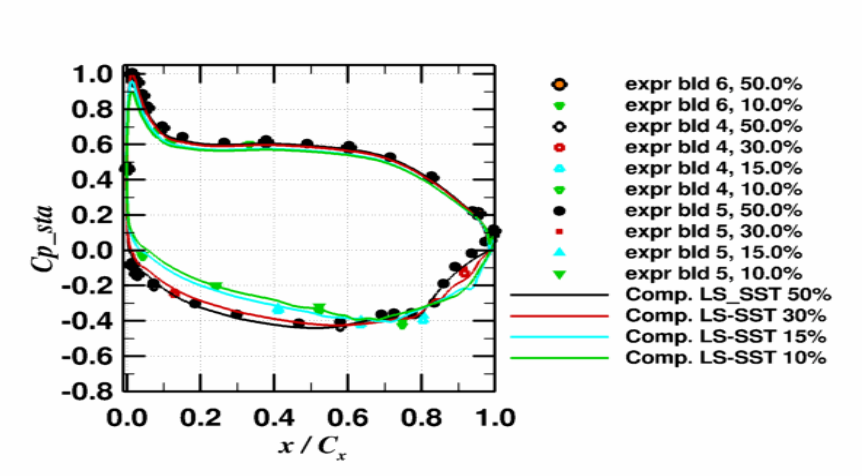


Figure 24. Comparison of CFD to data from NASA CW-22 facility. [1]

In the figures to follow, blade shapes, velocity distribution over the blade at two locations of peak and valley (except for the plane blade) with their pressure loadings and the total pressure loss distribution at a station x/axial chord = 10% downstream of the trailing edge will be shown. In addition, the average total pressure loss at that location will also be provided. Pressure loss is measured as a loss coefficient, $C_{pt} = (1.-Pt)/(Pt_{in}-Ps_{out})$. Here, Pt is the non-dimensional exit total pressure, Pt_{in}, is the non-dimensional inlet total pressure (equal to 1) and Ps_{out}, is the non-dimensional exit static pressure measured at 10% axial chord downstream of the trailing edge of the geometry.

All cases shown are for blades that are operated at a Reynolds number of approximately 100,000 based on inlet conditions which makes them transitional. Pressure ratio across the passages is 0.71. The incidence angle used for the following cases is +5°. After the results for the +5° are shown for the various blade profiles, a comparison between the VSPT baseline geometry and the S1R0P5 geometry is made for incidence angles of -37°, 0°, 5° and 10°.

Untreated VSPT Blade at 5° incidence

The time-averaged results for the untreated (baseline VSPT) blades are shown first. The blade shape picture is reproduced in figure 25a and 25b. The domain (not shown) is similar to that shown in figure 16 but a symmetry boundary condition is applied in the span-wise direction.

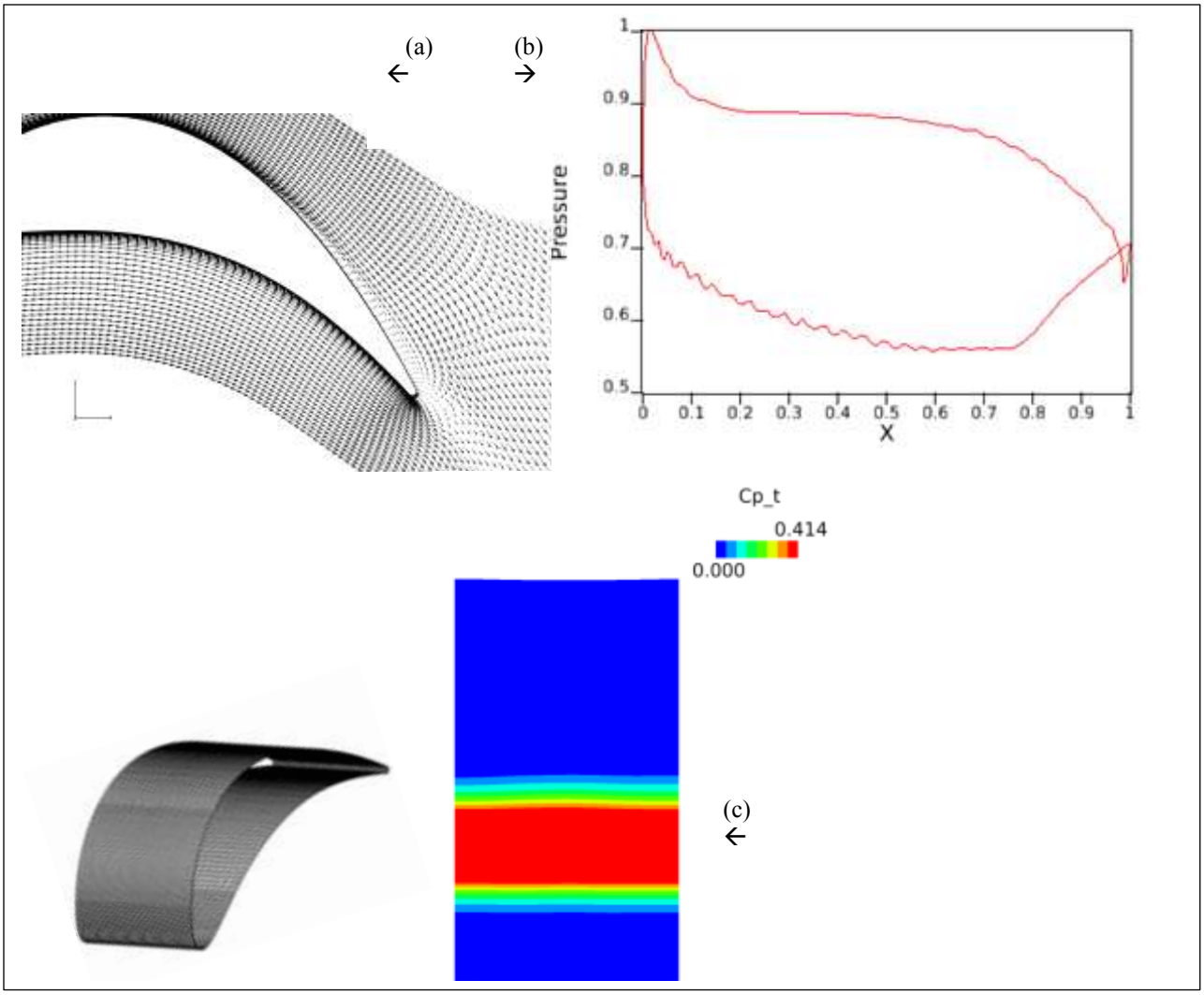


Figure 25. Untreated VSPT blade characterization, a) Velocity vectors at mid-span showing separation, b) pressure normalized by inlet total pressure, c) pressure loss coefficient, C_{pt}, in the wake.

Figure 25a shows the velocity distribution. The velocity vectors clearly show the area of separation which constitutes a large portion on the suction side of the blade. The pressure distribution shown in figure 25b also shows the effect of separation on the pressure loading which is in the form of an abrupt pressure rise and inflection in the curvature on the suction side instead of a smooth recovery. The distribution of the total pressure loss at an axial location x/c = 1.1 is shown in 14(c). As can be seen, the line of highest total pressure loss is aligned with the trailing edge as it should be. The distribution is spanwise uniform with the steady simulation as it should be. The average value at the plane downstream of the trailing edge (shown in figure 25d) was computed to be $C_{pt} = (1.-Pt)/(Pt_{in}-Ps_{out}) = 0.12$. This value needs to be reduced by the modifications to the blade.

Figure 26 shows an image from the water table. Dye injection shows flow at low Reynolds number moving from the pressure side to the suction side within the passage. Figure 27 shows a vortex shed from the suction side of the untreated VSPT blade.

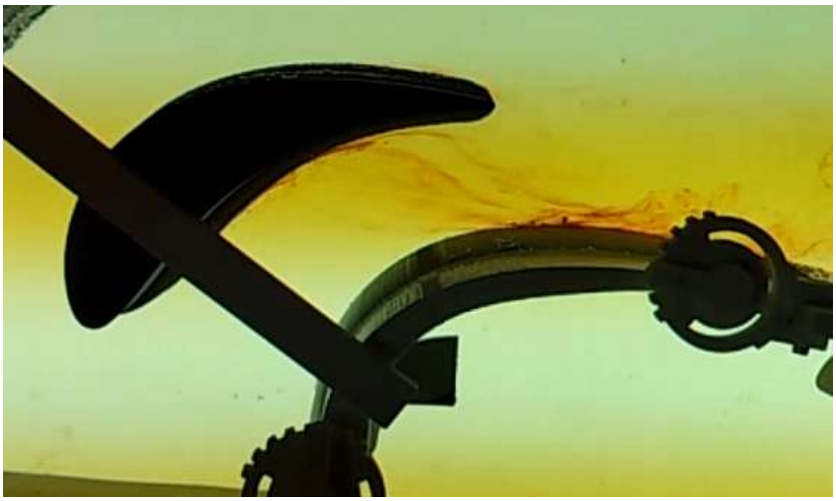


Figure 26. Photograph showing flow from the pressure side to the suction side of the VSPT turbine cascade in the water table facility, SE-1.

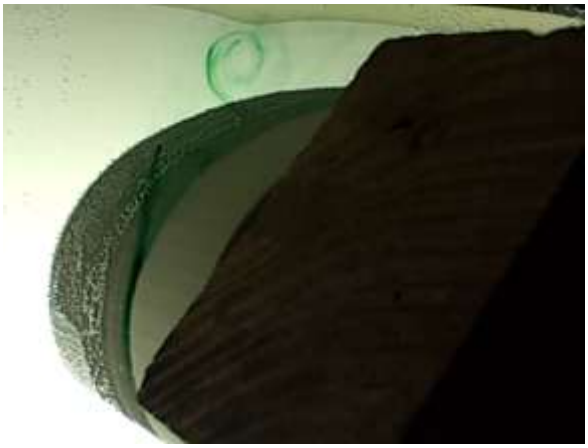


Figure 27. A shed vortex from the VSPT baseline blade in the water table at very low Reynolds number.

Figure 28 shows the horseshoe vortex pattern on the endwall of the water tunnel test section. Thin (0.05" thick) commercial dye pellets were placed on the floor of the tunnel and allowed to dissolve gradually. The powdered dye tends to accumulate near the saddle point of the horseshoe vortex.

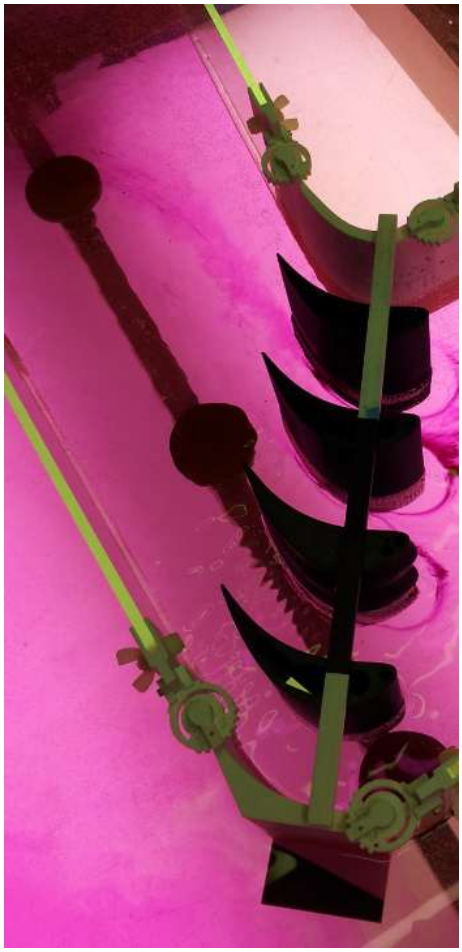


Figure 28. Visualization of Horseshoe pattern on endwall of water table using powdered dye. Powder accumulates in the 'dead zone'.

Figure 29 shows a comparison of IR thermography to CFD for the untreated blade at 0 incidence. There is good agreement between CFD and the experiment, however as the IR technique is a new method to characterize separation, more testing is required to validate these results. The figure is merely presented to show available capabilities and to confirm separation.

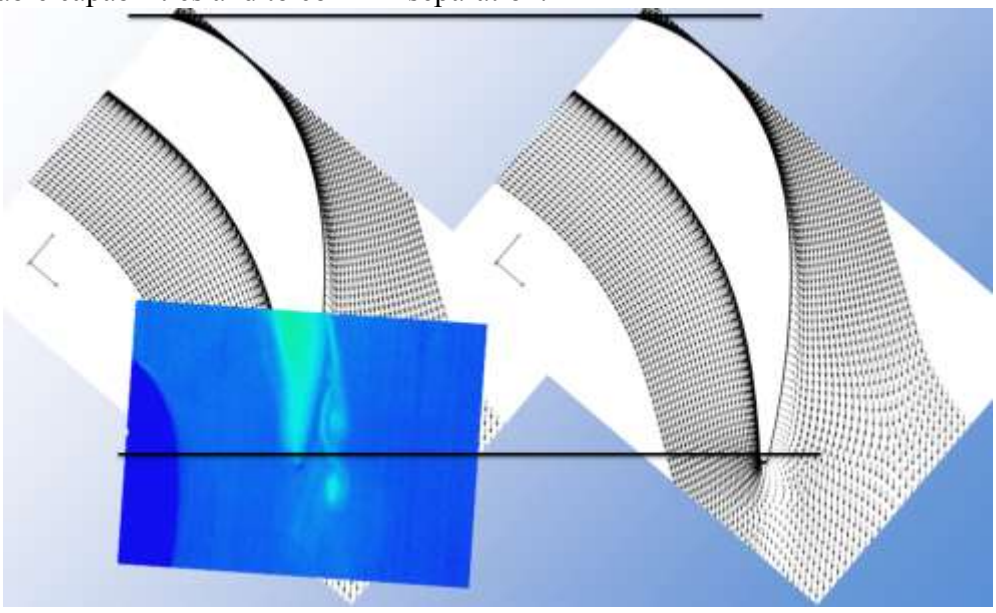


Figure 29 Comparison of water table IR measurement of separation with CFD for baseline geometry

Seal Vibrissae Treatments at 5° incidence

As described earlier, arrangements were tried by varying the amplitudes and the pitches of these whiskers over a spanwise domain which for the sake of ease of computations was assumed symmetrical. The cases (table 1) and the results are shown in figures below. The first case presented is for S1R0P5. The blade shape is such that at the leading edge, 3D relief causes the flow to accelerate around the peaks and into the valleys. The peak near the leading edge accelerates to the point of minimum pressure first and flow begins to experience an adverse pressure gradient. Adjacent to this location in the span-wise direction, the flow is still accelerating and the pressure is dropping. This creates a favorable span-wise pressure gradient. The flow that originated at the peak thus flows toward the valley location thus locally increasing the Reynolds number and reducing the adverse pressure gradient. Once the valley flow navigates past the crown minimum pressure location, it too starts to experience a chordwise adverse pressure gradient. However, due to the momentum added from the peak regions, there is enough energy to push any possible separation region to the trailing edge. This would only occur for aggressive geometry. The 3D relief provided by the peaks and valleys on the suction side, allows more flow to enter the valleys. The pressure side stays relatively unchanged. It is possible using various extrusion methods to create undulations only on the suction side and leading to tailor the undulations to the type of blade loading (aft, mid, front). The resulting waves on the trailing edge serve to mix the wake in the span-wise direction and thin the wake in the pitch-wise direction.

S1R0P5

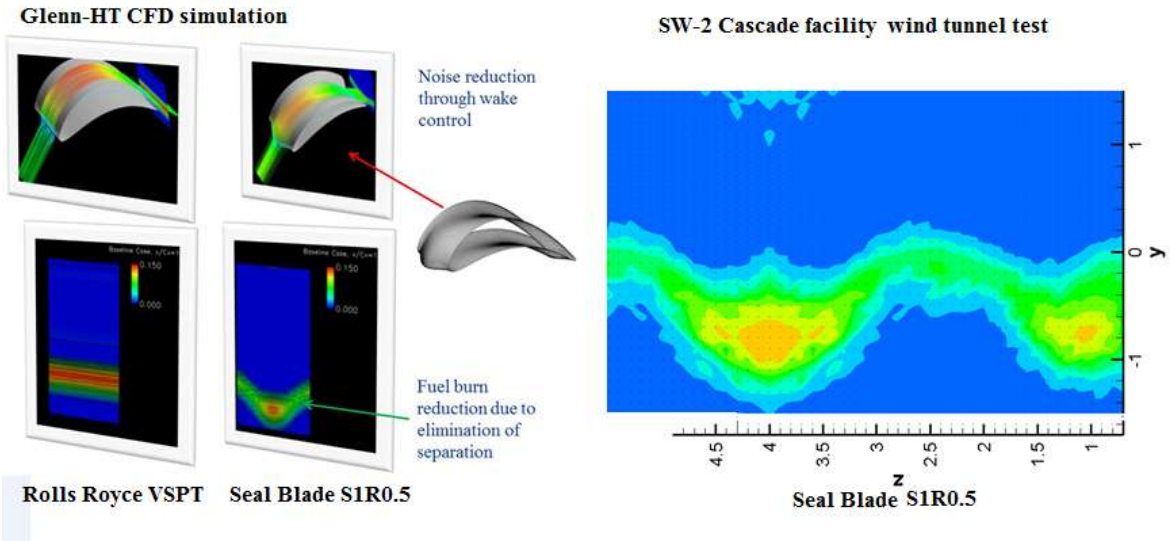


Figure 30. Instantaneous loss coefficient for a) baseline geometry and b) Seal blade. Streamlines are shown in the top contours while the loss coefficient downstream of the blade is shown in the bottom contours.

Figure 30 (left) shows a comparison of the instantaneous pressure loss coefficient in the wake for the baseline blade and for the seal blade S1R0P5 from the simulations. Figure 30 (right) shows the results of a pitot static survey at the same relative location as the CFD downstream of the S1R0P5 blade. The shape and extent of the wake is well captured by the CFD.

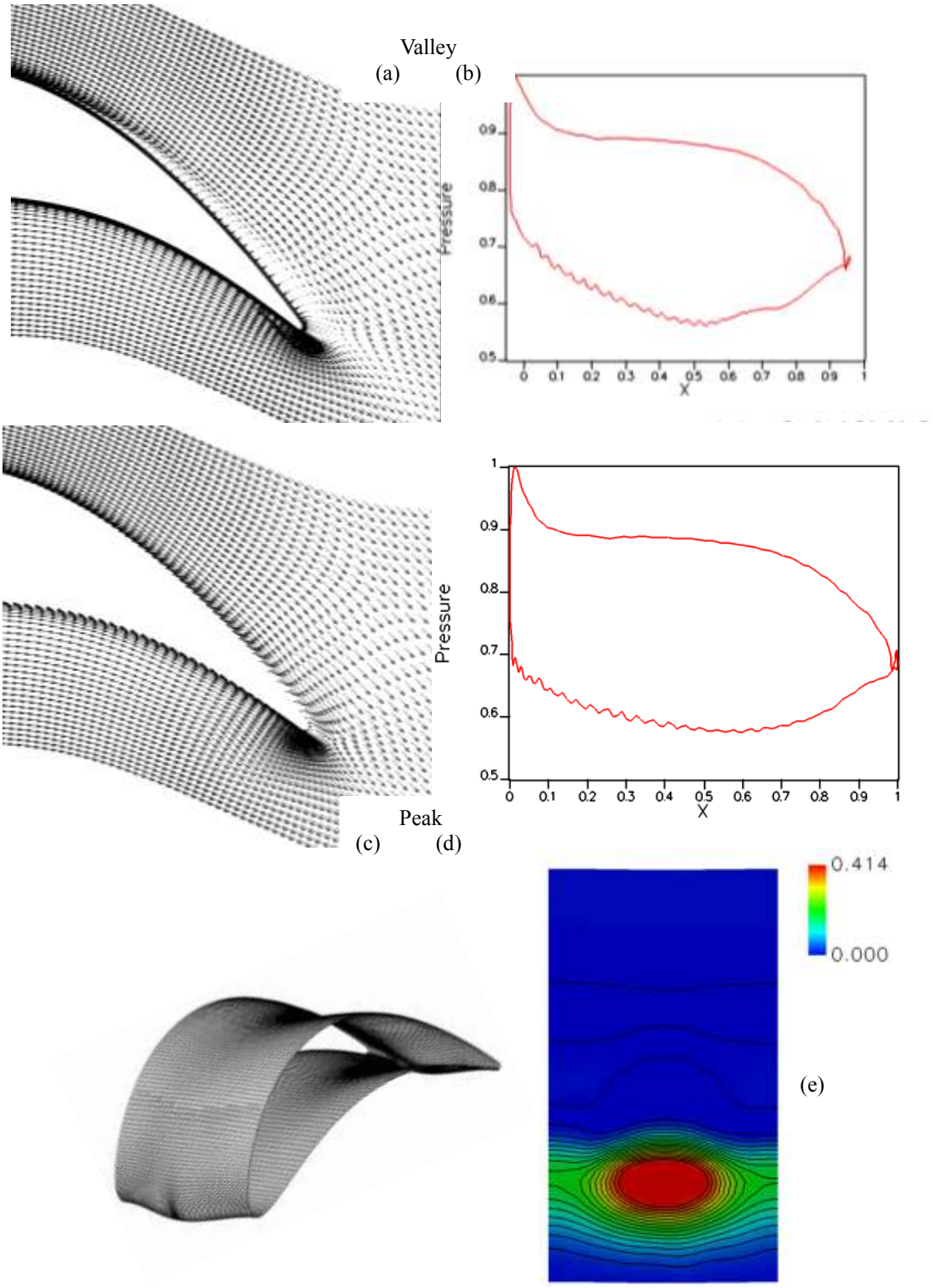


Figure 31. Characterization of the flow over the VSPT blade with S1R0P5 treatment at +5° incidence angle.

Figure 31a shows the blade shape used for this "experiment." Figures 31b and 31c show the velocity distribution and the pressure distribution for the same sections. Two sections are identified as "Peak" and "Valley". These are span-wise cuts across the blade that pass through Peak 3 and Valley 3 respectively (see figure 8.) These Peak and Valley locations are referred to in the remainder of this report. Figures 31b and 31c show the velocity distribution and loading for the Valley. They show the

trailing edge separation and the abrupt change in pressure on the suction side and thus the reduced loading of the blade. Along the peak, the separation is almost eliminated and pressure distribution on the suction side is quite benign. The wake loss contours at 10% chord downstream of the blade is most interesting where the low total pressure regions are concentrated behind the valley. The average total pressure loss coefficient for this case was computed to be $C_{pt} = 0.069$ which is approximately 30 percent lower than the baseline case. This is very promising. The particular conclusion that the treatment leads to boundary layer reattachment and a reduction of losses has been repeatedly verified. The pressure coefficient plots at the valley and peak locations show that the lift on the airfoil is significantly increased for the 'Peak' location while not changing much for the 'Valley' location. Note the 'filled in' region on the suction side of the profile in figure 31d (lower half of curve.)

Figure 32 shows the velocity deficit in the wake at a span-wise location corresponding to the 'Valley' from a partial hotwire survey in the SW-2 facility at an axial distance of 10% chord downstream of the trailing edge. Complete surveys of the wake are in progress but require further modifications to the tunnel to eliminate endwall effects. The baseline VSPT geometry clearly has a higher velocity deficit than the S1R0P5 blade.

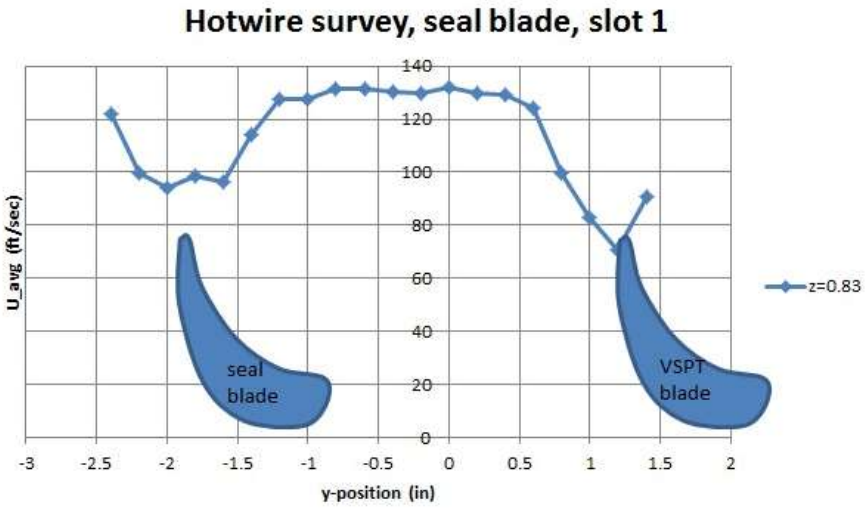


Figure 32. Hotwire survey downstream of VSPT blade (right) and Seal blade S1R0P5 (left).

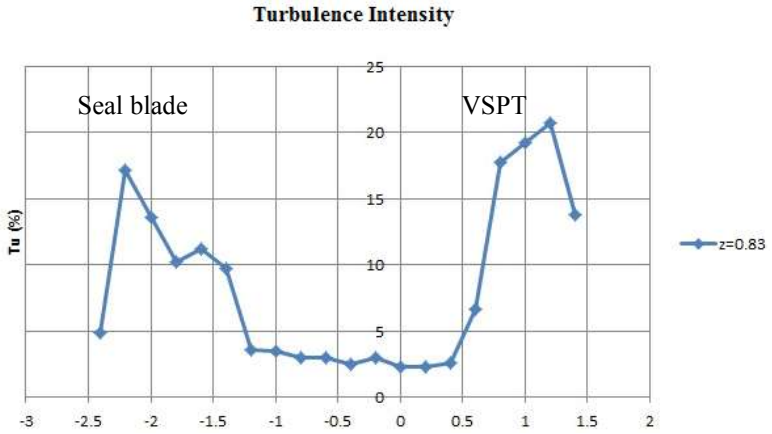


Figure 33. Turbulence intensity in the wake for S1R0P5 (left) and VSPT baseline (right).

Figure 33 shows a comparison of wake turbulence intensity downstream of the blades at a spanwise location corresponding to the 'Valley'. The turbulence intensity in the S1R0P5 blade is lower but appears to have a double peak shape to it. This requires further investigation.

The flow for the baseline blade is unsteady due to separation. The seal blade S1R0P5 also exhibits unsteadiness but mainly in the valleys where there is an unsteady separation. This is shown in figure 34 (bottom).

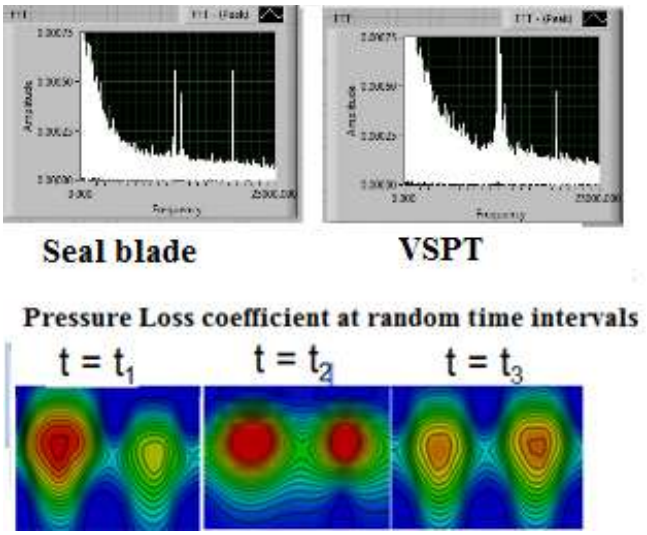


Figure 34. Comparison of S1R0P5 blade to baseline blade (top) and instantaneous snapshots of the wake immediately downstream of trailing edge of the seal blade S1R0P5

S1R1P0

Results for the S1R1P0 case are shown in figure 35 that is similar to the S1R0P5 case considered in figure 31. The level of integrated total pressure loss over the plane at x/c = 0.1 downstream of the trailing edge was computed to be $Cp_t = 0.066$ similar to the previous case.

S2R0P5 and S2R1P0

Figure 37 shows results for the S2R0P5 case that is similar to the S2R1P0 case considered. The level of integrated total pressure loss over the plane at x/c = 0.1 downstream of the trailing edge was computed to be Cpt = 0.066 similar to the previous case. The treatment again results in boundary layer stability in the "Peak" locations. The level of total pressure losses as computed was slightly higher than a single peak case. But, as described earlier, the computations were not always convergent and final convergence was not achieved using the 2-equation turbulence model. The issue was addressed for the case of S2R1P0 with the use of a three-equation model. The value so computed was Cpt=0.085 which is a significant improvement over the base case. Figure 38 shows CFD results for the case of high positive incidence (+10°). Figure 2a shows the baseline geometry while figure 2b shows one biomimetic case. Symmetry boundaries were used in the span-wise direction to speed up computations. The loss coefficient in figure 2b (bottom) is significantly reduced compared to that in figure 2a (bottom) and the wake thickness is also markedly reduced. Close examination of flow on the blade shows that separation has been removed. It was observed that the performance deteriorates with reduced pitch of the sinusoid and also with increased radius. There is therefore the potential to optimize this design.

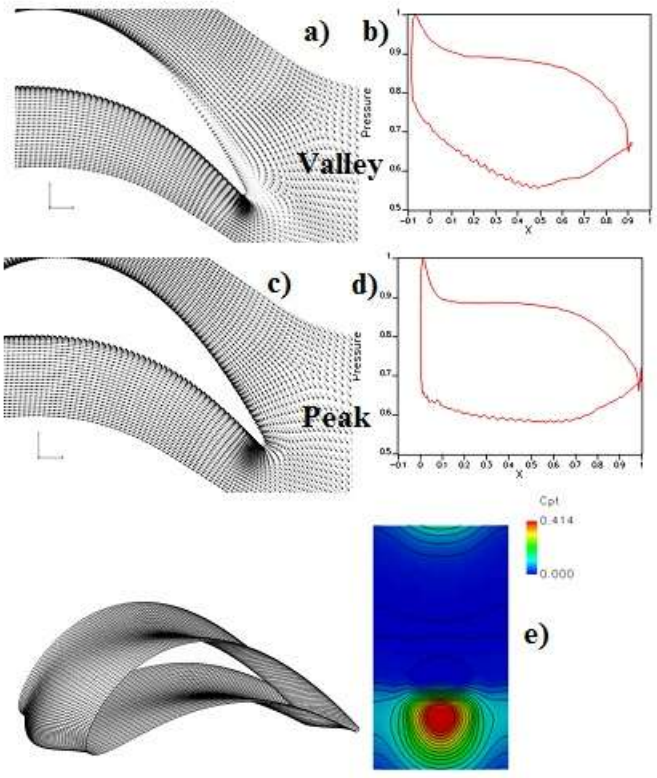


Figure 35. Characterization of the flow over the VSPT blade with S1R1P0 treatment

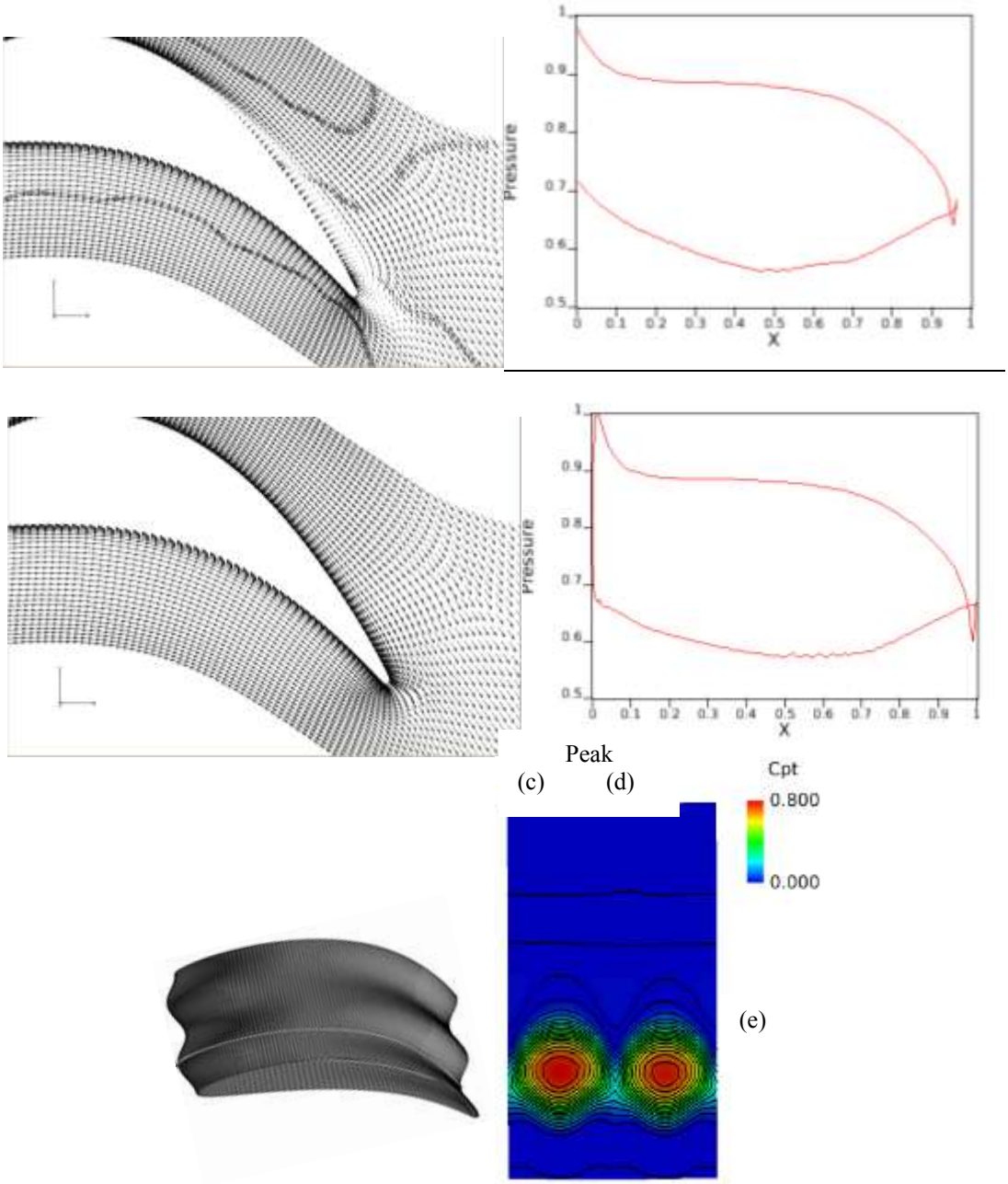


Figure 36. Characterization of the flow over the VSPT blade with S2R0P5 treatment

Effect of incidence angle

The same geometries were tested at 0°, 10° and -37° incidence to check whether performance had been compromised at other incidence angles. The seal treated blade showed improvements over the untreated

blades over all incidence angles. Figure 37a shows velocity vectors for the untreated VSPT blade at 10° incidence angle and figure 37b shows pressure distribution at mid-span. Figures 38 and 39 shows results for the S1R0P5 and S1R1P0 blades respectively at 10° incidence angle. Here, at the peak location, separation is again seen to be removed and the wake structure is similar to that at lower incidence angles.

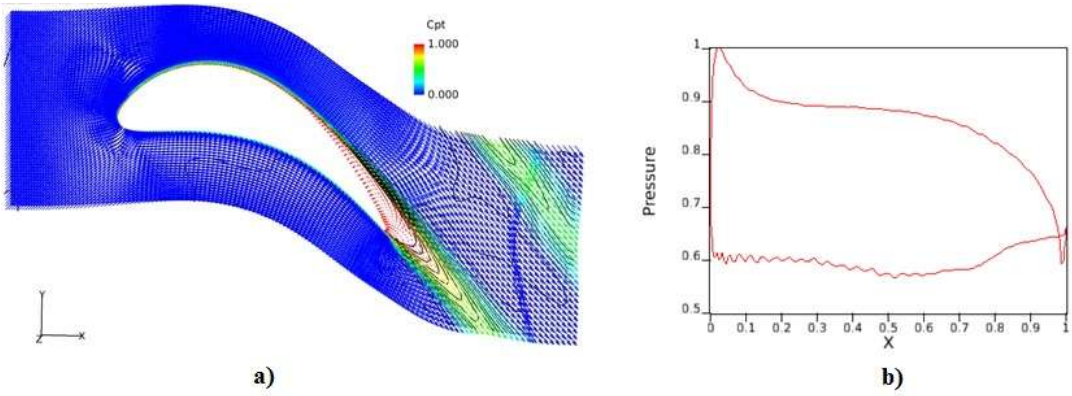


Figure 37. Untreated VSPT blade at 10° incidence angle.

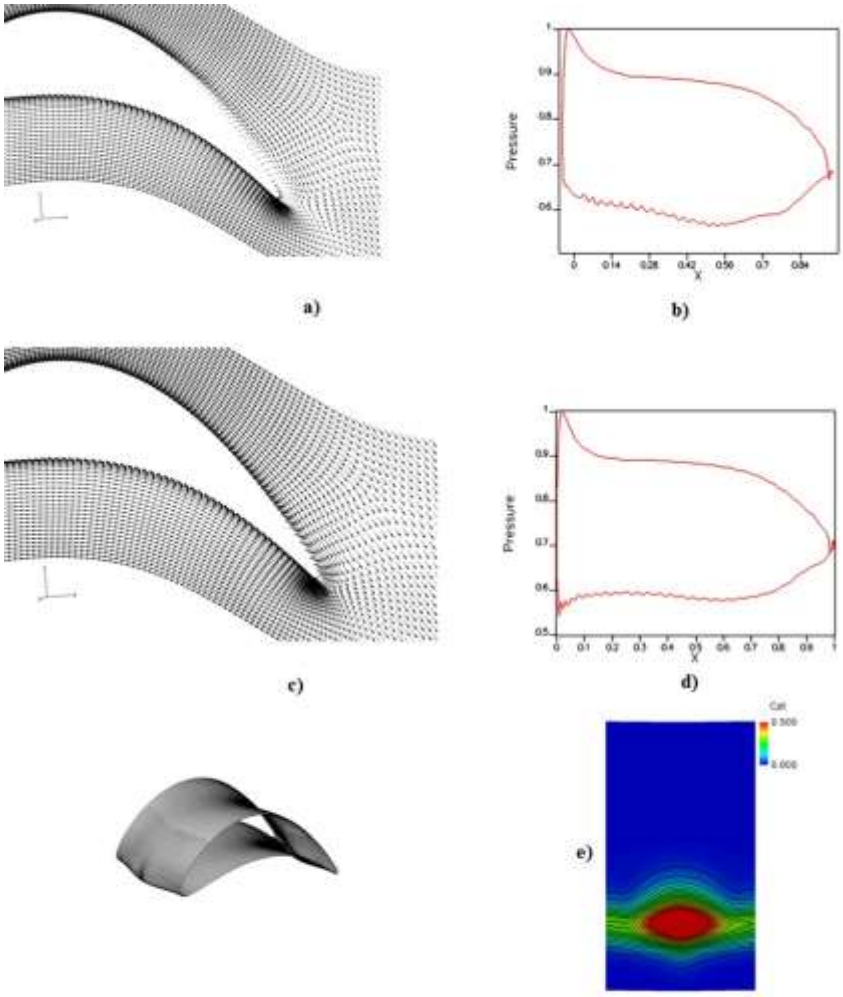


Figure 38. S1R0P5 at 10° incidence angle.

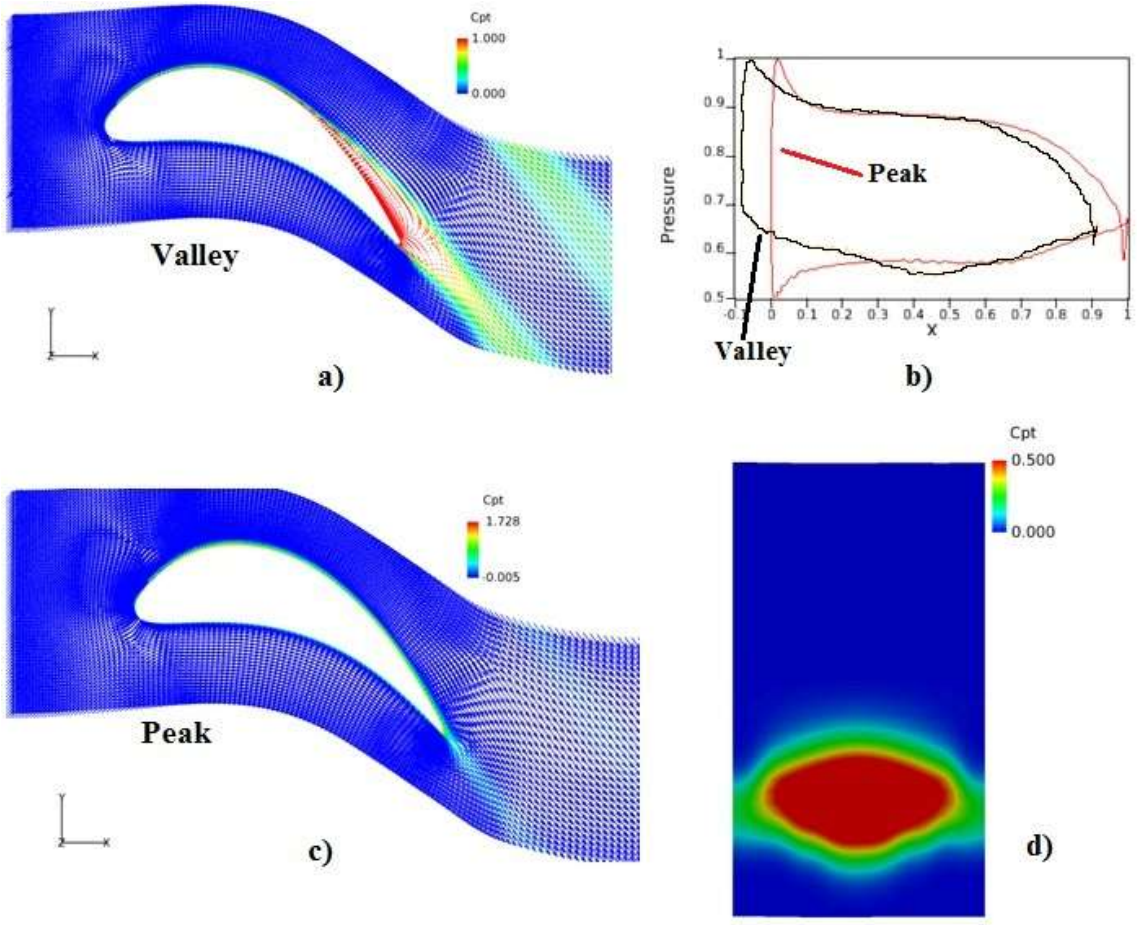


Figure 39. S1R1P0 at 10° incidence angle.

Figure 40 shows results for the untreated VSPT blade at high negative incidence angle. Figure 41 and 42 show results for the S1R0P5 and S1R1P0 cases respectively. An interesting result was noted at the negative incidence angle. While the untreated blade showed separation on the pressure side and on the suction side at high negative incidence, the Valley location of the Seal blades did not show separation on the pressure side while the Peak location showed no separation on the suction side. This is again a hint that optimization of the undulations along the chord could be beneficial to the overall performance of the blade. For example, the undulation amplitude could be reduced as the flow progresses toward the trailing edge. Alternatively, undulation sweep could be introduced along the trailing edge or alone the entire blade if endwall flows are considered.

Summary of results

Figure 43 (Left) shows a plot of total pressure loss coefficient that is an indication of drag for a wide range of incidence angles. The blue line is the Rolls Royce VSPT Power Turbine blade that was optimized by Rolls Royce for the range of incidences shown. The red line shows the loss associated with the Seal Blade. The Seal Blade is largely insensitive to incidence angles and figure 43 (Right) shows averaged wake loss coefficients from the unsteady simulations of the multiple Seal Blade variations that were studied using 3D unsteady Computational Fluid Dynamics (CFD). Figure 43 (Right) shows that increased pitch of the undulations causes a decrease in benefit from the undulations. For a given pitch, it appears that a larger undulation amplitude has lower loss (refer to table 1 for pitch and amplitude specifications for each case.)

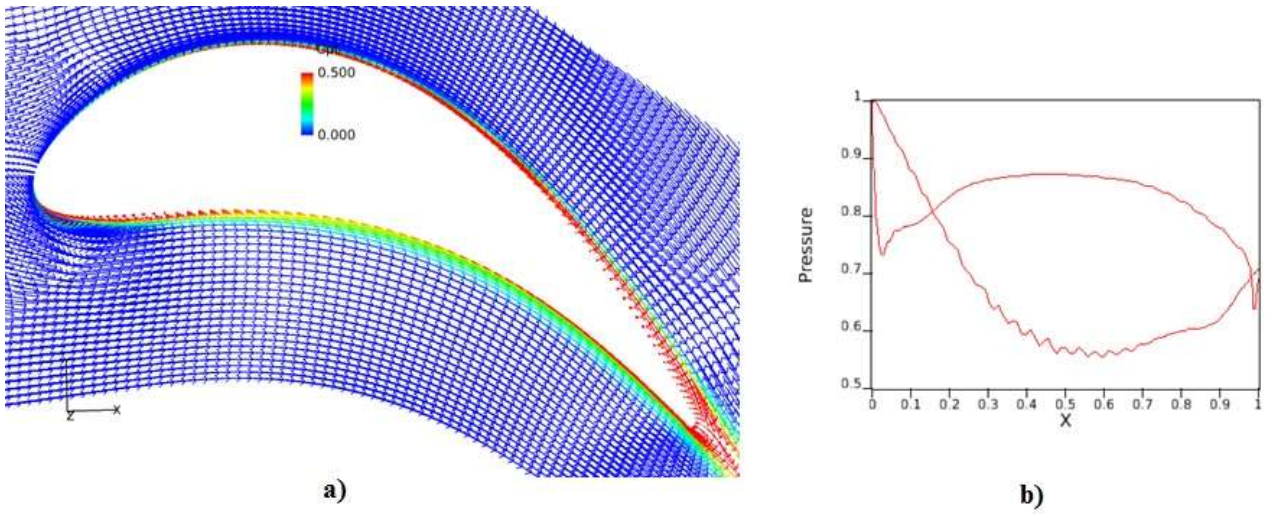


Figure 40. VSPT at -37° incidence angle.

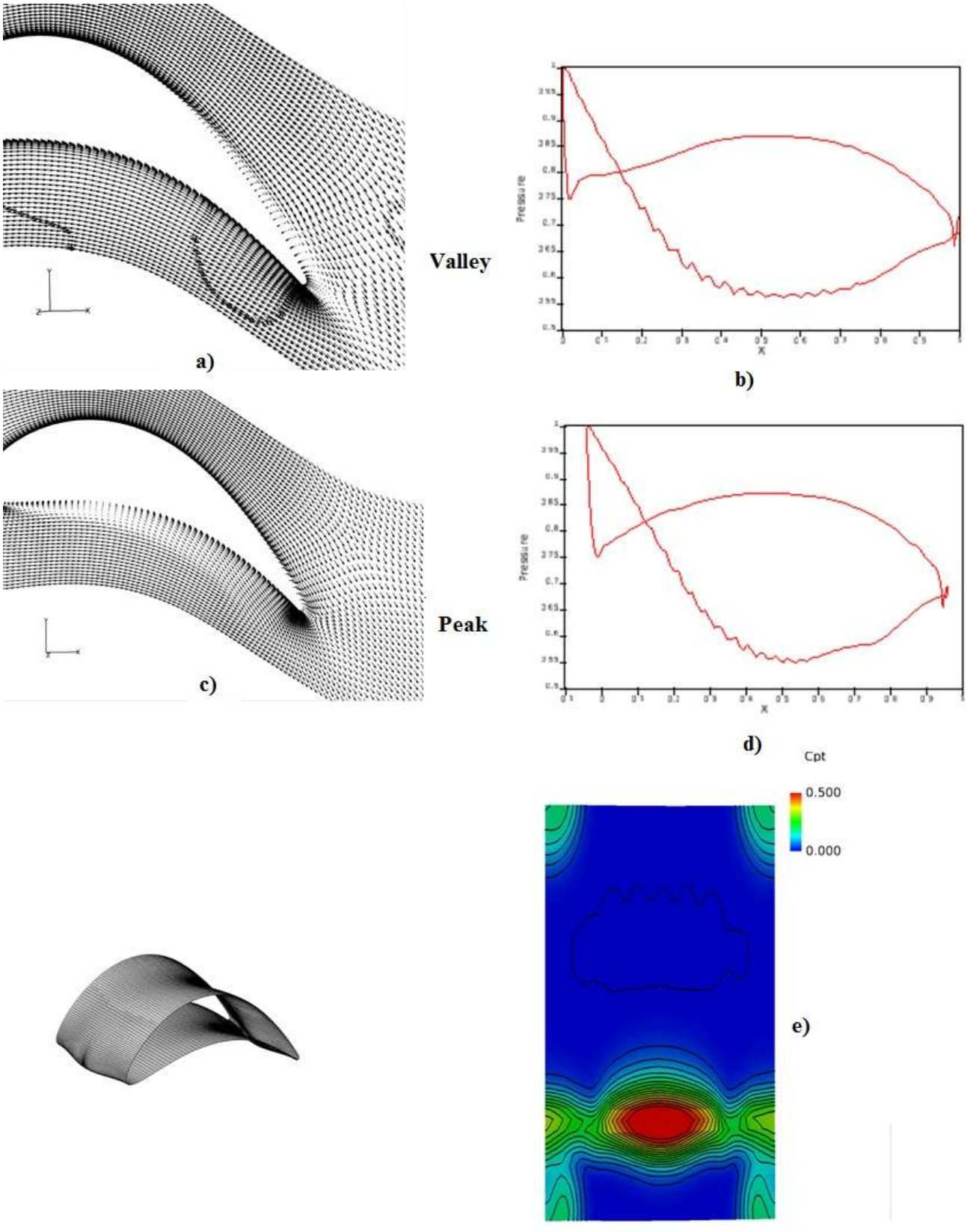


Figure 41. S1R0P5 at -37° incidence angle.

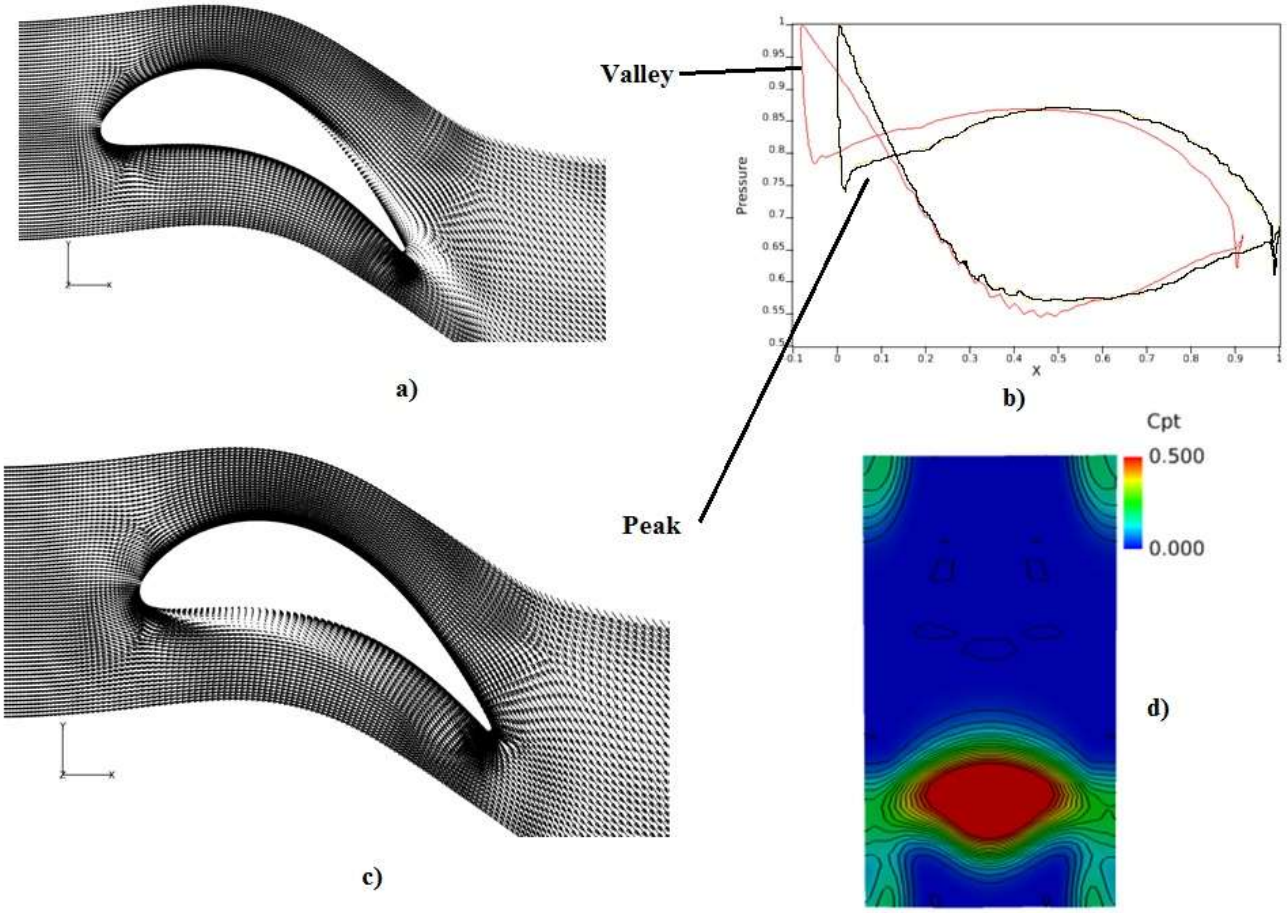


Figure 42. S1R1P0 at -37° incidence angle.

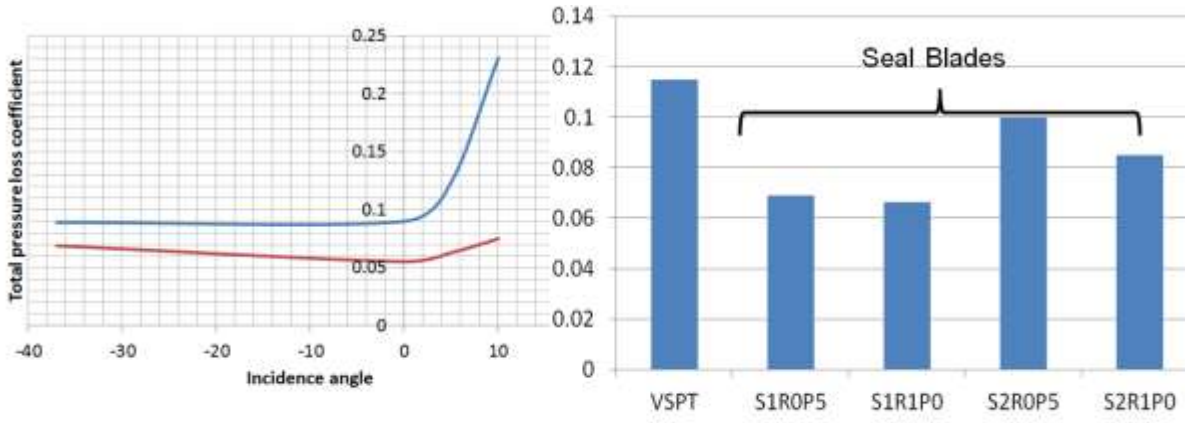


Figure 43. Incidence tolerance (Left) - Comparison between baseline VSPT blade and S1R0P5 over a wide range of incidence angles (-37°, 0°, 5° and 10°). Performance improvements from Seal blades (Right) – average pressure loss coefficient in the wake for seal blades compared to VSPT blade at 5° incidence angle.

Figure 44 shows the surface pressure loading on the VSPT and on the S1R0P5 blade. The solid black lines were added for reference. The VSPT blade experiences separation that the Seal blade S1R0P5 does not experience near the Peak. At the Valley, the S1R0P5 blade does separate but not as dramatically as the VSPT blade. There is loss of lift at the mid-chord location but this lift is more than compensated for in the aft region of the chord.

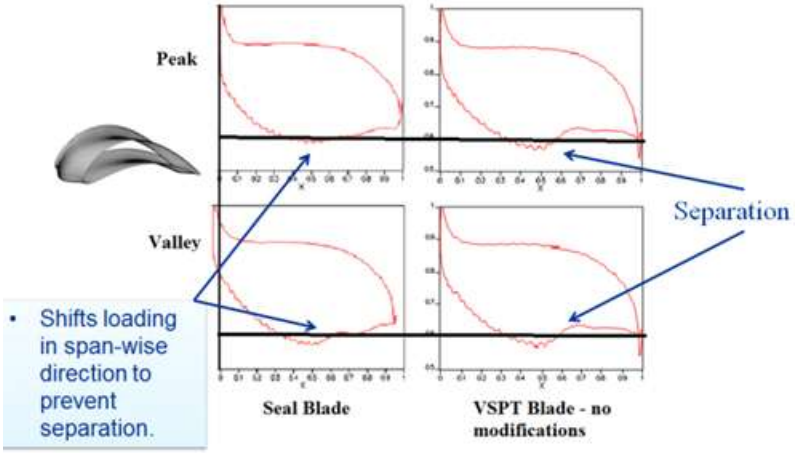


Figure 44. Direct comparison of loading on blade between S1R0P5 and VSPT at 5° incidence angle.

Unsteady snapshots (figure 45 bottom) of the wake revealed an interesting clumping of structures that deserve analysis for potential noise reduction. Figure 45 (top) shows that low frequency broadband noise and tones were reduced although these results are preliminary and are in the process of being verified. Seal Vibrissae are known to cause rapid breakdown of large turbulent structures. This allows noise to be dissipated close to the source. Acoustic measurements should be taken further downstream to verify this result and a wake survey is required.

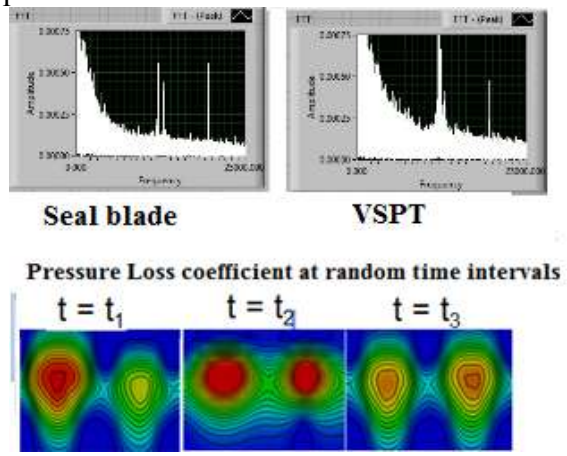


Figure 45. Unsteadiness in the wake. Top - FFT (fast fourier transform) of hotwire data in the wake. Bottom - CFD simulation results at 3 random time steps.

ACFC (Autonomous Closed-Loop Flow Control) Results

Endwall Suction Slot Simulation (Source flow for flow control)

The idea of endwall suction was to remove the endwall boundary layer which gives rise to blockage in the blade passage and to introduce the removed gas in the endwall boundary layer into the blade on the

suction side and through the trailing edge. The following simulations look at the first part of this idea in isolation, namely the suction of the boundary layer. For that purpose, two computations were made. One without suction and another with suction through the endwall, upstream of the blade as shown in figure 18. The suction rate was set based on the pressure ratio that would be established between the upstream flow and the suction side of the blade at mid-span. That value was determined to be 0.71. The Reynolds number was 100,000 based on inlet condition and an exit Mach number of 0.72. The inlet boundary layer thickness was 15% of the full linear cascade height and the inlet angle was 40 degree or at +5 degrees of incidence.

Results are shown in the following three sets of figures. In figure 46 the velocity vectors showing separation over the suction side at mid-span are shown. It may be difficult to judge whether the separation is diminished due to suction at the hub. Figure 47 shows planes normal to the axial direction on which contours of the total pressure loss are shown. The top figure is with suction and the one below is without suction. The view on the suction side is being shown. The contours of the losses just downstream of the L.E. show larger total pressure loss values for the Without Suction case as may be expected. This trend continues into the downstream wake plane where both the total pressure losses due to secondary flow and those due to mid-span boundary layer separation are reduced for the Suction case. Figure 48 shows the same planes viewing the pressure side. It is observed that the level of losses for the With Suction case is reduced compared to the Without Suction case. Without suction, the average loss coefficient, C_{pt}, was calculated to be 0.134 at a plane located x/c = 0.1 downstream of the trailing edge. With suction, the loss coefficient was found to be 0.105. This is a 22% reduction in aerodynamic loss.

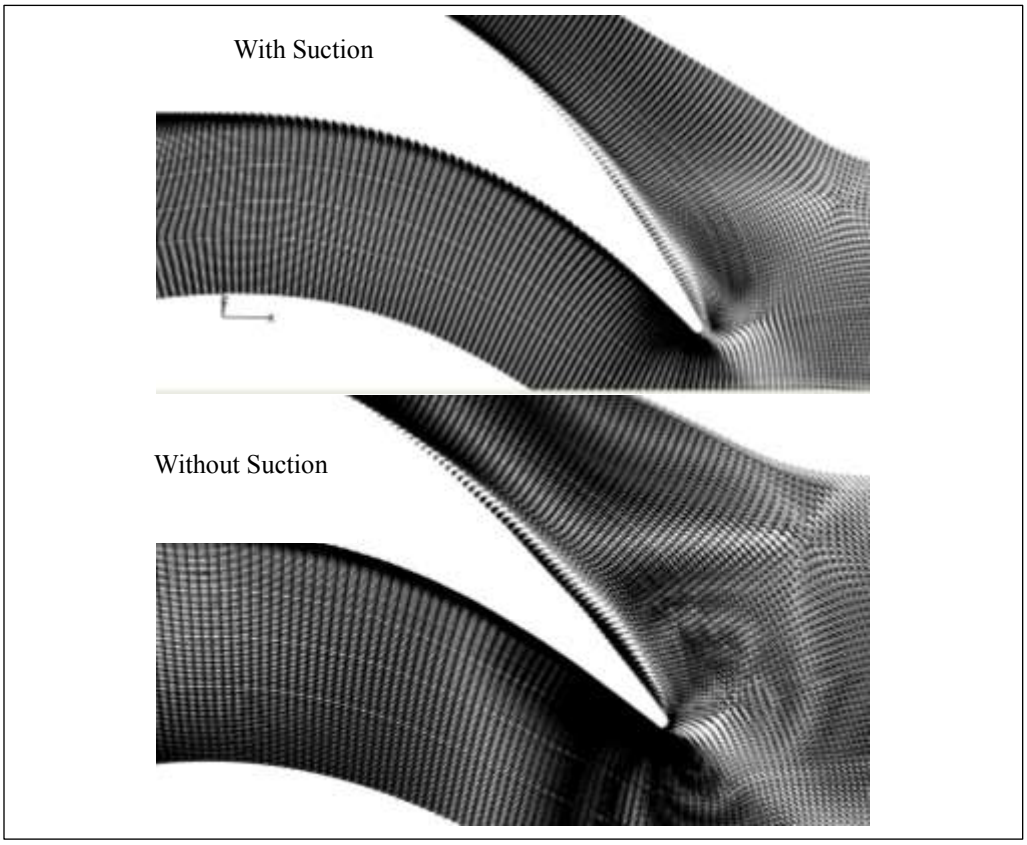


Figure 46. Velocity vectors showing separation over the suction side. Top - With Suction, Bottom - Without Suction

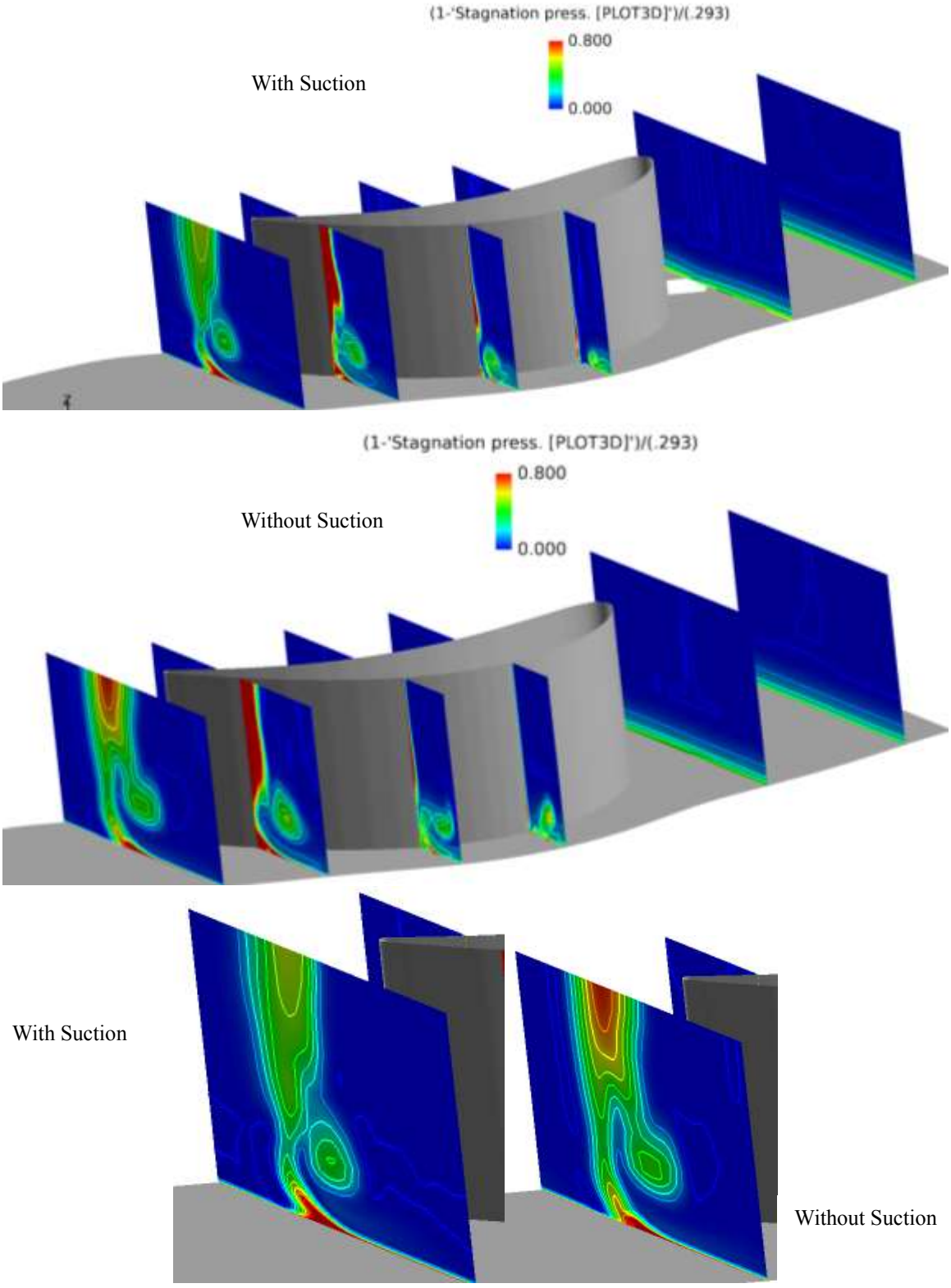


Figure 47. Total pressure losses on axial planes. Top with suction, Middle without suction and bottom comparing wake plane (enlarged).

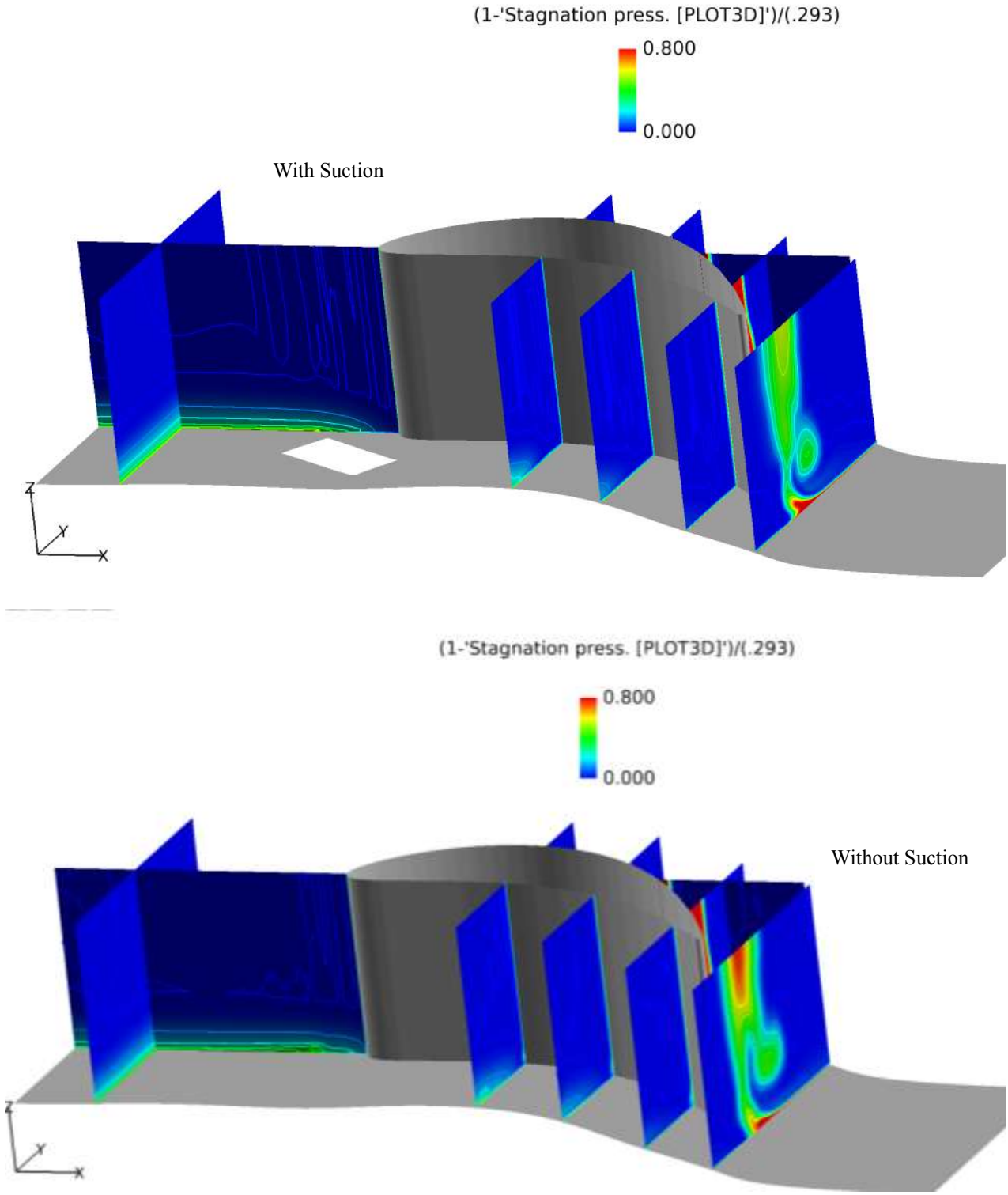


Figure 48. Total pressure losses on axial planes. Top with suction, Middle without suction and bottom comparing wake plane (enlarged).

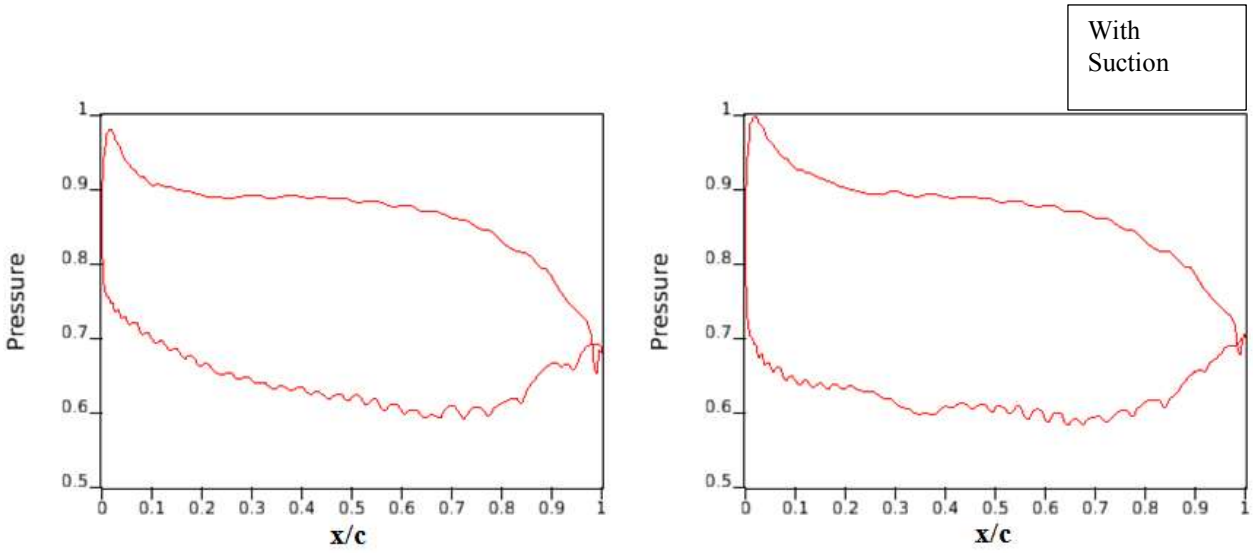


Figure 49. Comparison of blade loading at 5% span. Left - Without Suction, Right - With Suction.

Figure 49 shows a comparison of pressure loading at 5% span for the With Suction and Without Suction cases. There is a clear improvement in pressure distribution between the leading edge (x/c = 0) and x/c = 0.7 on the suction side of the blade. The pressure recovery at the leading edge is also greatly improved. This would lead to a reduced blade height to obtain the same work or in the case of a compressor blade to enable better compression.

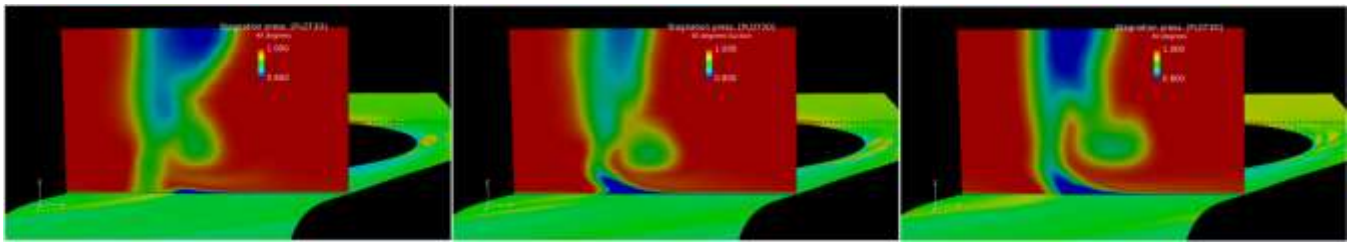


Figure 50. Effect of incidence on wake for With Suction case. Left - +10° incidence angle, Center - +5° incidence angle, Right – No Suction, +5° incidence angle.

Figure 50 shows the effect of incidence change on the normalized stagnation pressure at a plane x/c = 0.1 downstream of the trailing edge. There is an interesting shift in the pattern of the wake. For both cases with suction, the wake deficit is greatly reduced. However, it is clear that as conceptualized at the beginning of this Phase 1 effort, a distributed suction slot would be more amenable to boundary layer control.

Fluidic Network Results

Figure 51 shows the functioning of a diverter-type fluidic device that was modelled using Solidworks and printed using a FORTUS 250mc 3D printer. Several orifice diameters, channel widths and exit and inlet geometry were studied to determine the sensitivity of the device. The devices were supplied with shop air and allowed to exit onto a level water surface in a 20 gallon aquarium. Videos of the operation of the devices were made for various conditions of the actuator (port 1 closed, port 1 open, port 2 closed, port 2 open.) All possible sequences were attempted and a state map was created to determine the possible modes of operation of the device and to ensure repeatable, consistent control. Figure 51 lists a summary of the outcome for F1.

Figure 52 shows a fluidic pulsing device, F2 that sweeps the exit flow using a Helmholtz resonator. This requires some frequency content entering the device. It is assumed that if the source flow is in turbomachinery, unsteady content is always present in the flow. The resonator is able to then create small amplitude fluctuations at frequencies close to the resonator frequency.

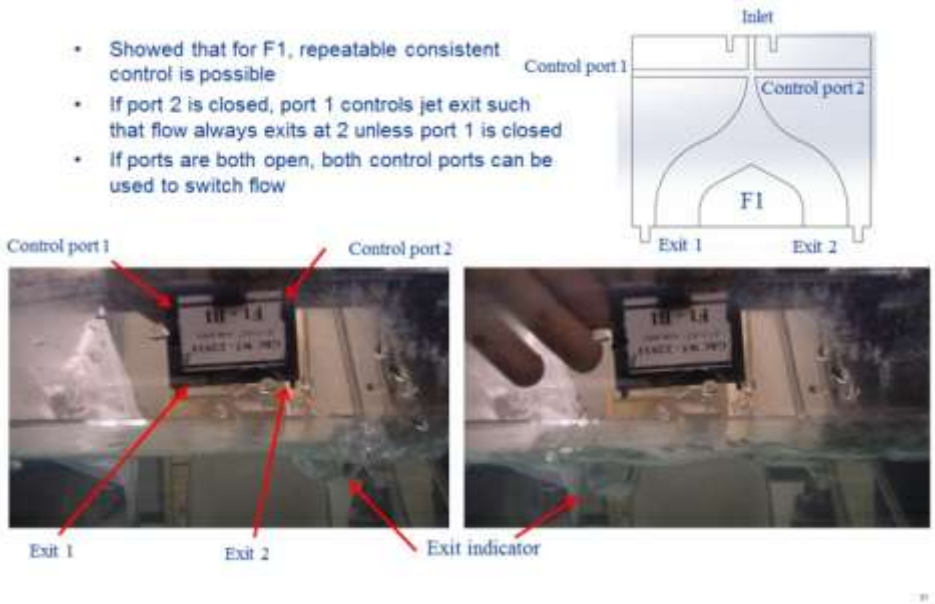


Figure 51. Operation of Fluidic diverter, F1.

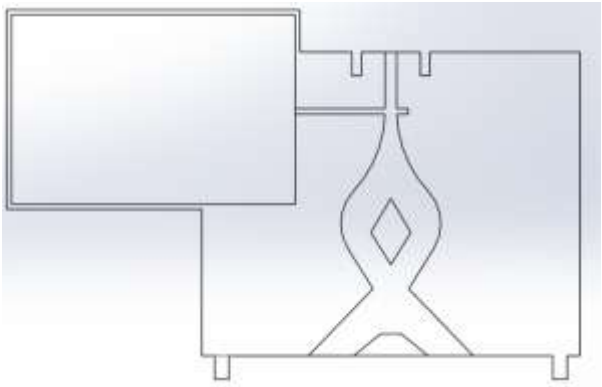


Figure 52. Helmholtz resonator driven fluidic pulser or sweeper.

The actuator shown in figure 52 was tested for frequency content using shop air at 35psig. There being no imposed frequency in the inlet, the outlet frequency content only showed minor tonal content.

Figure 53 shows part of the fluidic network system assembled for a bench top test. The objective of the test was to test the functionality of the system once all components were assembled. The actuator F3 was found to produce oscillations at a frequency of 1700Hz that is in close agreement with its design frequency of 2000Hz. The discrepancy can be attributed to the pressure ratio across the device and the lack of airflow over the exit plane of the oscillator.

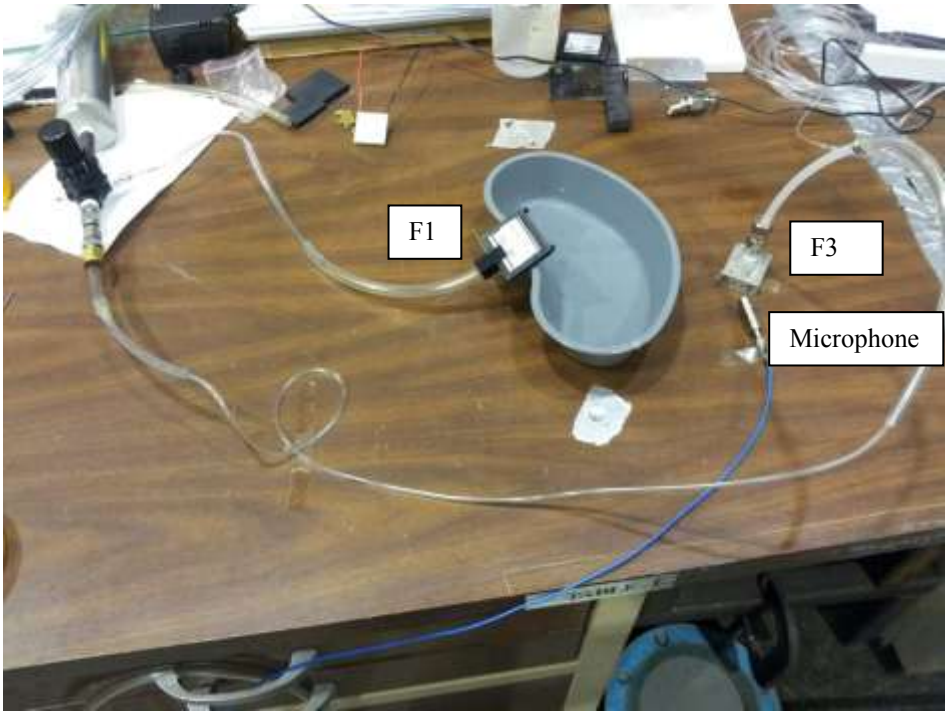


Figure 53. Testing of fluidic network.

Figure 54 shows pressure contours within a fluidic sweeping device that exits onto a flat plate. The contours were obtained using the Glenn-HT simulation tool.

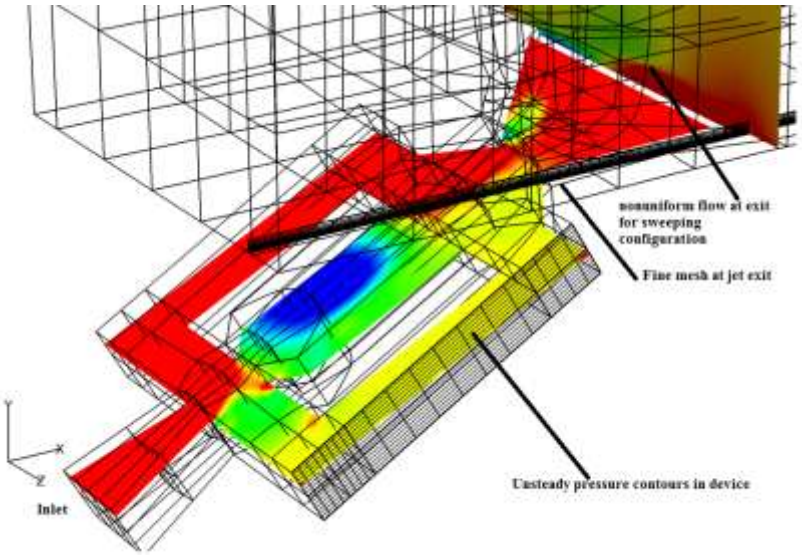


Figure 54. Computational analysis of a fluidic sweeping device.

At the trailing edge it was proposed to create pulsed jets using the actuator shown in figure 55 (patented by Advanced Fluidics LLC.) The actuator exits (figure 55) are spaced 0.5" apart and supplied with air at 35psig. The actuator was tested in NASA's SW-2 cascade facility) at inlet Mach number of 0.1 and using a single-wire hotwire probe at 10% and 20% chord downstream of the trailing edge. The low Mach number testing is due to the high cost of power and low resource availability in Phase 1.

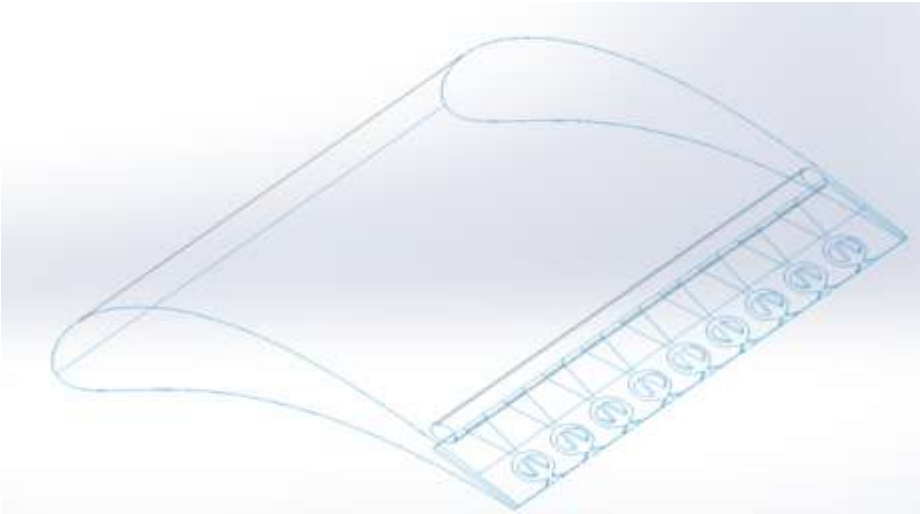


Figure 55. Advanced fluidics trailing edge fluidic sweeper.

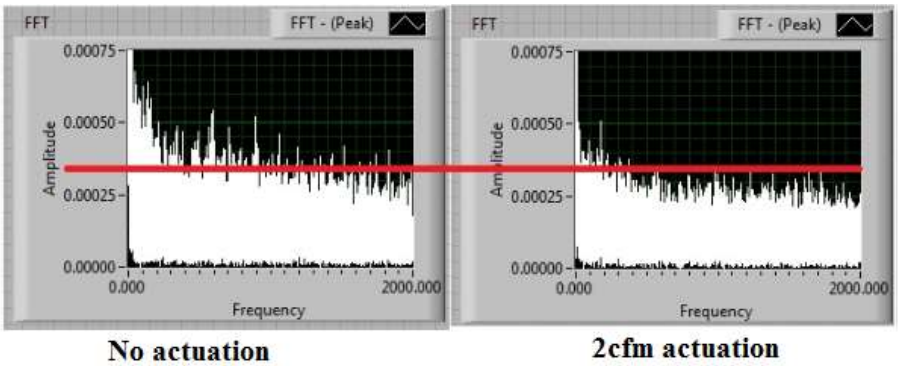


Figure 56. Frequency content in the shear layer on the suction side wake at 80fps tunnel velocity

The inlet pressure to the plenum that feeds the oscillators was varied using a pressure regulator and metered using a flow meter. Flow rates of 0cfm, 1cfm, 2cfm and 3cfm were attempted at tunnel velocities of 55fps and 80fps. At a flow rate of 4cfm, the actuator blowing ratio exceeded 2 and led to additional tonal content in the wake. Figure 56 shows the frequency content in the sheer layer on the suction side of the wake with and without actuation for a flow rate of 2cfm and a tunnel inlet velocity of 80fps. Wake surveys are currently in progress to obtain frequency content over a larger area of the wake. The preliminary results do however show that the amplitudes of the tones observed in the no actuation case dropped significantly and the broadband levels also dropped. Figure 57 shows frequency content in the wake at a tunnel velocity of 55fps. The overall amplitude levels at each frequency are seen to drop. Again, a 1cfm actuation is seen to decrease the amplitudes across the spectrum.

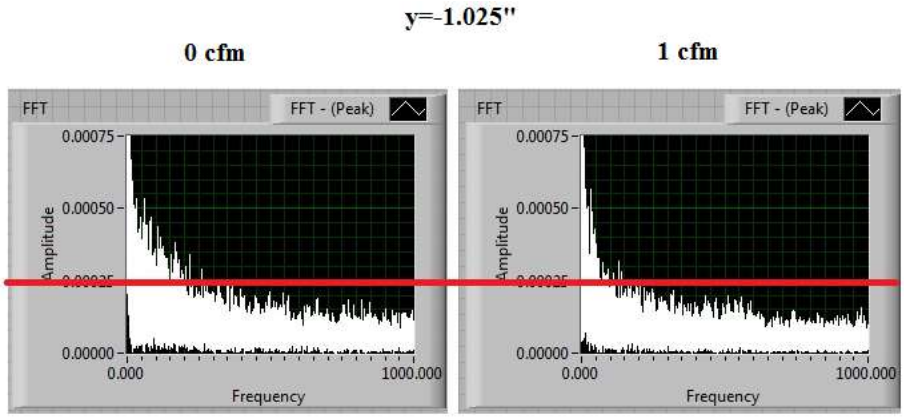


Figure 57. Frequency content in the shear layer on the suction side wake at 80fps tunnel velocity.

Figure 58 shows the filling of the wake deficit upon actuating with 2cfm flow rate through the oscillators.

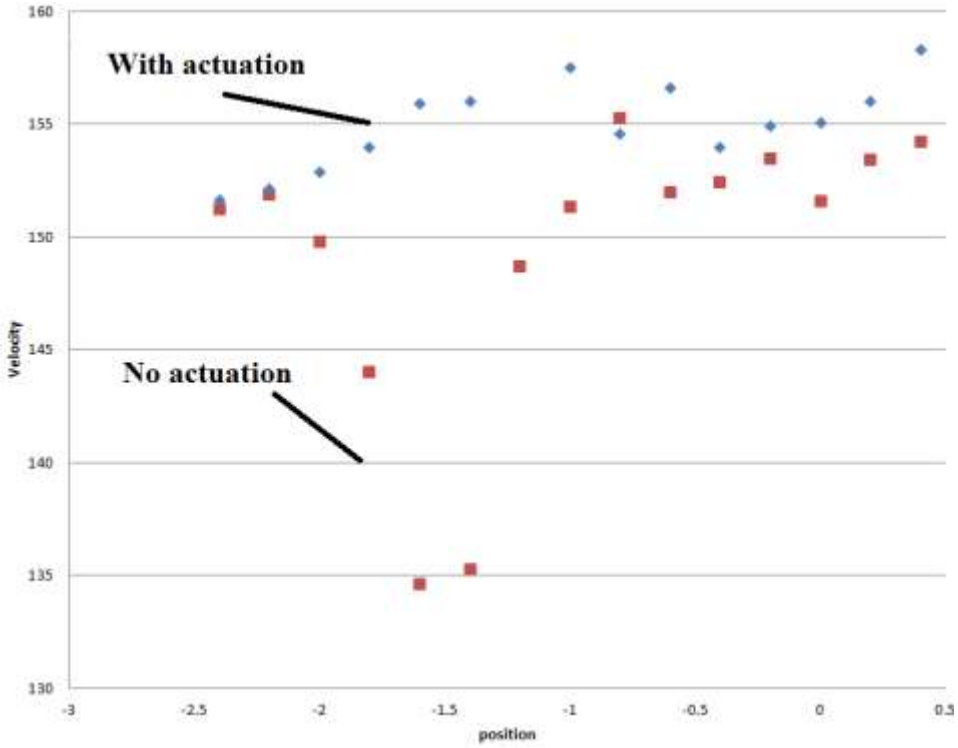


Figure 58. Velocity profile in the wake.

Systems benefits

Although both LPT performance and weight are important aircraft design variables in determining aircraft fuel burn, percentage improvements in LPT efficiency result in greater fuel burn reduction than a similar LPT weight decrease (which yields a more-modest engine weight decrease). This is shown in Figure 59, developed from NASA system studies for large, long-range FW aircraft (where fuel weight is more than an order of magnitude larger than engine weight) as well as a notional, regional-class RW vehicle (where fuel weight is only 3-4x of engine weight). A 5% reduction in engine weight enables a

1% reduction in overall fuel burn for the FW aircraft, and about ½ that for the RW vehicle. However, each 1% improvement engine specific fuel consumption (similar to a 1% improvement in LPT efficiency) yields a 1.67% reduction in overall fuel burn for the FW aircraft and a 1.2% reduction for the RW vehicle. Also on the figure is the initial estimate for reduced engine weight and improved efficiency, predicted at the start of the Phase 1 effort. Two regions of expected improvement in fuel burn are also included, based on the results of this Phase 1 effort. Although they are modest improvements, these reductions are a significant amount of fuel and potential savings in operational costs.

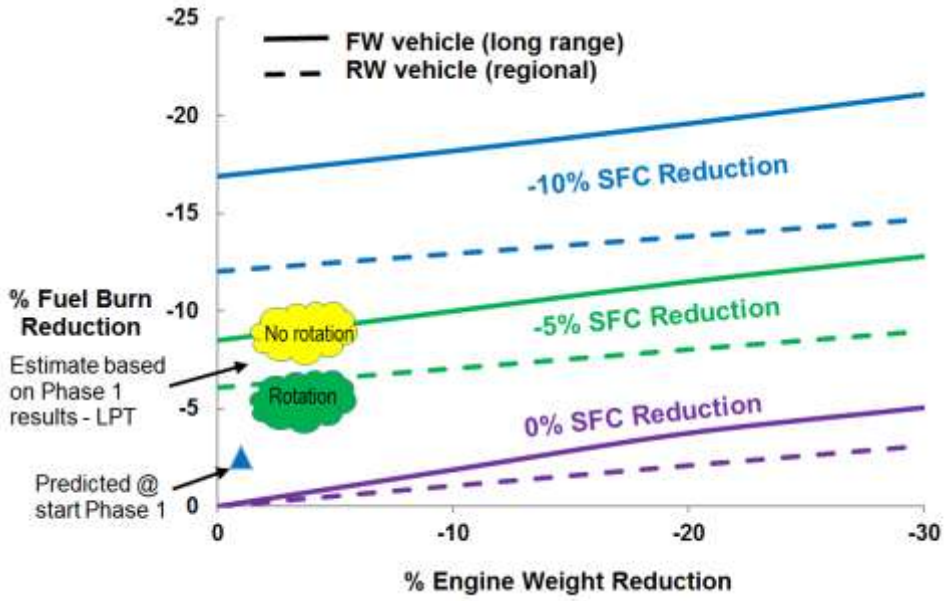


Figure 59. Fuel burn reduction predicted as a result of Phase 1 effort.

Acoustic benefits were not quantified in this report due to the lack of complete surveys of the wake. However, given the reduction in amplitudes in the shear layer shown from both the Seal Vibrissae inspired blades and from the trailing edge pulsing, it is estimated that noise reduction of at least 3-5db can be achieved through careful arrangement of the pulsing actuators and from the seal blade undulations. This figure of 3-5db is significant because this is the sensitivity of human hearing to changes in noise levels. Without further testing it is not possible to determine how much more than this base noise reduction could be achieved.

Conclusions:

- Feasibility of Biomimetic geometry shown for Fuel burn reduction
  - Smaller undulation pitch has larger benefit on performance.
  - Amplitude of undulation does not have a significant effect.
  - Blades treated with the Seal Vibrissa–like undulations were shown to be insensitive to incidence tolerance.
  - Blades treated with the Seal Vibrissa–like undulations were shown to decrease aerodynamic losses compared to an untreated blade.
  - 5% fuel burn reduction may be achieved.
  - Acoustic benefits possible.
- Feasibility of Autonomous Closed-Loop Flow Control concept

    - Source flow through suction slot shown to improve loading for lower 10% of span
    - Source flow mechanism shown to reduce loss by 20%.
    - Fluidic network system was assembled and found to operate as expected using benchtop testing.
    - Trailing edge pulsing reduces the broadband amplitude in the wake that indicates acoustic benefits.
- SW-2 linear cascade facility was modified with a new test section, inlet guide boards and tail boards.
- PIV system was set up in NASA's SW-6 facility in collaboration with Fundamental Aeronautics Fixed Wing Program.
- Water table facility set up in NASA's SE-1 to obtain quantitative flow information for biological and bio-inspired features that are difficult and costly to analyze in wind tunnels.
- Real-time quantitative flow visualization prototype demonstrated.
- More can be achieved by applying to fan noise reduction, compressor stall control, airframe drag and noise reduction, landing gear drag reduction and structural integrity. Similar gains may be achieved for struts, probes, road signs, electrical cables, marine transport.

Due to separated nature of the flows considered, the computations tend to be unsteady. Three-equation models using phenomenological transition models were helpful and should be attempted for computation of the cases involving Seal Vibrissae in combination with experimental measurement as possible. Direct numerical simulation is also a possible approach that can be attempted albeit at lower Reynolds numbers.

Acknowledgements

The authors would like to thank the National Aeronautics Research Institute for funding the work reported herein.

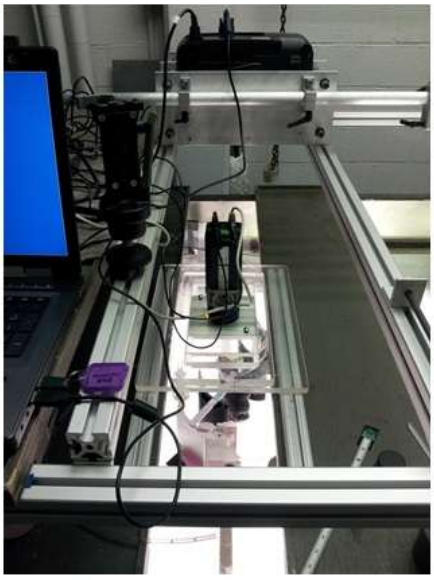
References

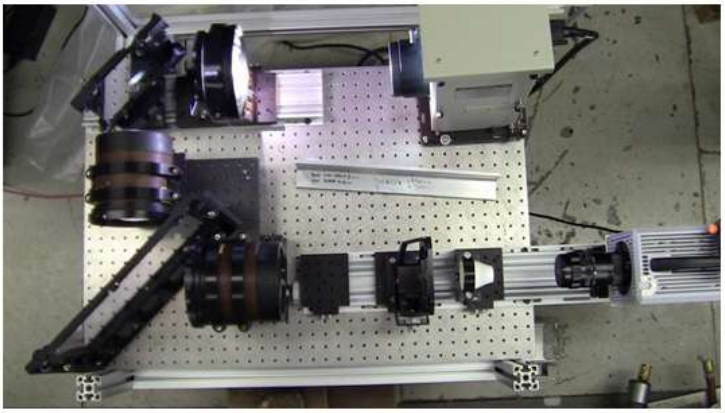
1. Welch, Gerard E., McVetta, Ashlie B., Stevens, Mark A., Howard, Samuel A., Giel, Paul W., Ameri, Ali A., To, W. M, Skoch, G. J., and Thurman, D. R., "Variable-Speed Power Turbine Research at Glenn Research Center," Presented at the American Helicopter Society 68th Annual Forum, Fort Worth, Texas. May 1-3, 2012.
2. Johnson, W., "NDARC, NASA Design and Analysis of Rotorcraft," NASA TP 2009-215402, December 2009.
3. Hodson, H.P. and Howell, R. J.; Bladerow Interactions, Transition, And High-Lift Aerofoils In Low-Pressure Turbines. Annu. Rev. Fluid Mech. 2005. 37:71–98. D. S. Miklosovic, M. M. Murray, L. E. Howle, and F. E. Fish. "Leading-edge tubercles delay stall on humpback whale (Megaptera novaeangliae) flippers." Physics of Fluids, Vol.16, March 2004.
4. Bloxham M.M., et al. (2009): Synchronizing Separation Flow Control With Unsteady Wakes in a Low-Pressure Turbine Cascade. J. Turbomach., vol. 131, no. 2, pp. 021019.
5. Bohl, D.G.; and Volino, R.J. (2006): Experiments With Three-Dimensional Passive Flow Control Devices on Low-Pressure Turbine Airfoils. J. Turbomach., vol. 128, pp. 251–260.
6. Bons, J.; Sondergaard, R.; and Rivir, R.B. (2000): Turbine Separation Control Using Pulsed Vortex Generator Jets. J. Turbomach., vol. 123, no. 2, pp. 198-206.
7. Bons, J.; Sondergaard, R.; and Rivir, R.B. (2002): The Fluid Dynamics of LPT Blade Separation Control Using Pulsed Jets. J. Turbomach., vol. 124, pp. 77–85.

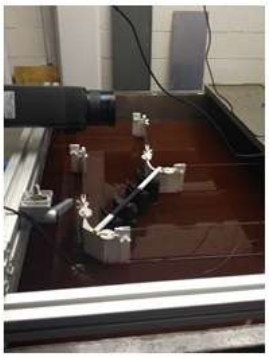
8. Thomas C. Corke, C. Lon Enloe, and Stephen P.Wilkinson, "Dielectric Barrier Discharge Plasma Actuators for Flow Control", Annual Review of Fluid Mechanics, Vol. 42: 505-529 (Volume publication date 2010), First published online as a Review in Advance on September 21, 2009, DOI: 10.1146/annurev-fluid-121108-145550.
9. Miller, C., Shyam, V., Rigby D., "Multifunctional Low-Pressure Turbine for Core Noise Reduction, Improved Efficiency, and Nitrogen Oxide (NOx) Reduction", NASA/TM—2013-218095.
10. Halasz, C., "Advanced Trailing Edge Blowing Concepts for Fan Noise Control: Experimental Validation," MS Thesis, Virginia Polytechnic Institute, 20 June 2005, Blacksburg, VA.
11. Frank E. Fish, Paul W. Weber, Mark M. Murray and Laurens E. Howle, "The Tubercles on Humpback Whales' Flippers: Application of Bio-Inspired Technology", Integrative and Comparative Biology, volume 51, number 1, pp. 203–213, doi:10.1093/icb/icr016.
12. Ernst A. van Nierop, Silas Alben, and Michael P. Brenner. "How bumps on whale flippers delay stall: An aerodynamic model." Physical Review Letters, 100(5):054502, February 2008.
13. Paul W.Weber, Laurens E. Howle, Mark M. Murray, and David S. Miklosovic. "Computational evaluation of the performance of lifting surfaces with leading-edge protuberances." Journal of Aircraft, 48(2):591{600, 2011.
14. Kristy L. Hansen, Richard M. Kelso, and Bassam B. Dally. Performance variations of leading-edge tubercles for distinct airfoil pro_les. Journal of Aircraft, 49(1):185{194, 2011.
15. H. Johari, C. Henoch, D. Custodio, and A. Levshin. E_ects of leading-edge protuberances on airfoil performance. AIAA Journal, 45(11):2634{2642, 2007.
16. D.S. Miklosovic, M.M. Murray, and L.E. Howle. Experimental evaluation of sinusoidal leading edges. Journal of aircraft, 44(4):1404{1408, 2007.
17. Okamoto, M., Yasuda, K., and Azuma, A., "Aerodynamic Characteristics of the Wings and Body of a Dragonfly." Journal of Experimental Biology. Vol. 199, 1996, pp. 281- 294.
18. Antonia, B. Kessel, "Aerodynamic Characteristics Of Dragonfly Wing Sections Compared With Technical Aerofoils," The Journal of Experimental Biology 203, 3125–3135 (2000) 3125, Printed in Great Britain © The Company of Biologists Limited 2000, JEB2811.
19. Masatoshi Tamai, Zhijian Wang, Ganesh Rajagopalan, Hui Hu, "Aerodynamic Performance of a Corrugated Dragonfly Airfoil Compared with Smooth Airfoils at Low Reynolds Numbers", 45th AIAA Aerospace Sciences Meeting and Exhibit, Jan 8 – 11, 2007, Reno, Nevada AIAA-2007-0483.
20. Babu, P., and Mahesh, K., " Aerodynamic Loads on Cactus-shaped Cylinders at Low Reynolds Numbers", Phys. Fluids 20, 035112 (2008); doi: 10.1063/1.2887982.
21. S. Talley, G. Iaccarino, G. Mungal, and N. Mansour, "An experimental and computational investigation of flow past cacti," Annual Research Briefs, Center for Turbulence Research, NASA Ames/Stanford University, 2001, pp. 51–63.
22. Mayer, H., Schindler, D., "Wind Effects on Trees," Proceedings of the 2nd International Conference Albert-Ludwigs-University of Freiburg, Germany, 13-16 October 2009.
23. Halle, J., "Aerodynamic Features of the Tree", Journal of Arboriculture 16(9); September 1990.
24. Chen, Y., Chew, y.t., Khoo B.C., "Turbulent Flow Manipulation by Passive Devices", Proceedings of the 13th Asian Congress of Fluid Mechanics, 17-21 December 2010, Dhaka, Bangladesh.
25. Woong Sagong, Chulkyu Kim, Sangho Choi, Woo-Pyung Jeon, and Haecheon Choi, "Does the Sailfish Skin Reduce the Skin Friction Like the Shark Skin?" Phys. Fluids 20, 101510 (2008); doi: 10.1063/1.3005861.
26. T. Geyer, E. Sarradj, C. Fritzsche, "Silent Owl Flight: Experiments in the Aeroacoustic Wind Tunnel," NAG/DAGA 2009, Rotterdam.

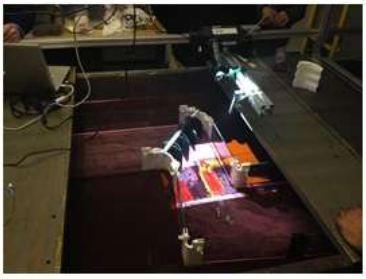
27. Dehnhardt, G., Mauck, B., Hanke, W. and Bleckmann, H., "Hydrodynamic trail following in harbor seals (Phoca vitulina)," Science 293, 102-104, 2001.
28. Wolf Hanke, Matthias Witte, Lars Miersch, Martin Brede, Johannes Oeffner, Mark Michael, Frederike Hanke, Alfred Leder, and Guido Dehnhardt, "Harbor seal vibrissa morphology suppresses vortex-induced vibrations," The Journal of Experimental Biology 213, 2665-2672 © 2010. Published by The Company of Biologists Ltd., doi:10.1242/jeb.043216.
29. Bloxham, M., "A Global Approach to Turbomachinery Flow Control: Loss Reduction using Endwall Suction and Midspan Vortex Generator Jet Blowing," PhD thesis, The Ohio State University, Columbus Ohio, 2010.
30. Raghu, S. & Raman, G. (1999) Miniature fluidic devices for flow control. ASME FEDSM 99-7256.
31. Raman, G., Raghu S. and Bencic T.J. (1999) Cavity Resonance Suppression Using Miniature Fluidic Oscillators, AIAA-99-1900, 5th AIAA/CEAS Aeroacoustics Conference, Seattle, WA, May 10-12, 1999.
32. Woszidlo R., Nawroth H., Raghu S. and Wygnanski I.J. (2010) Parametric Study of Sweeping Jet Actuators for Separation Control AIAA 2010-4247, AIAA Flow Control Conference, Chicago, June 2010.
33. Culley, D.E. and Bright, M.M., Prahst, P.S., and Strazisar, A. J., "Active Flow Separation Control of a Stator Vane Using Surface Injection in a Multistage Compressor Experiment", NASA/TM-2003-212356, GT2003–38863ASME, IGTI Turbo Expo, Atlanta, Georgia, 16-19 June, 2003.
34. Steinthorsson, E.; Liou, M.S.; and Povinelli, L.A.: Development of an Explicit Multiblock/Multigrid Flow Solver for Viscous Flows in Complex Geometries. AIAA–93–2380 (NASA TM–106356), 1993.
35. Langtry, R.B.; and Sjolander, S.A.: Prediction of Transition for Attached and Separated Shear Layers in Turbomachinery. AIAA 2002–3641, 2002.
36. Walters, D.K. and Leylek, J.H., 2004, "A New Model for Boundary-Layer Transition Using a Single-Point RANS Approach, " ASME Journal of Turbomachinery, Vol. 126, pp. 193-202, ASME Paper No. IMECE2002-HT-32740.
37. Thurman, D., Poinsatte, P., Heidmann, J., "Heat Transfer Measurements for a Film Cooled Turbine Vane Cascade," Proceedings of ASME Turbo Expo 2008: Power for Land, Sea and Air GT2008, June 9-13, 2008, Berlin, Germany, GT2008-50651

Appendix A – Pictures from phase 1.

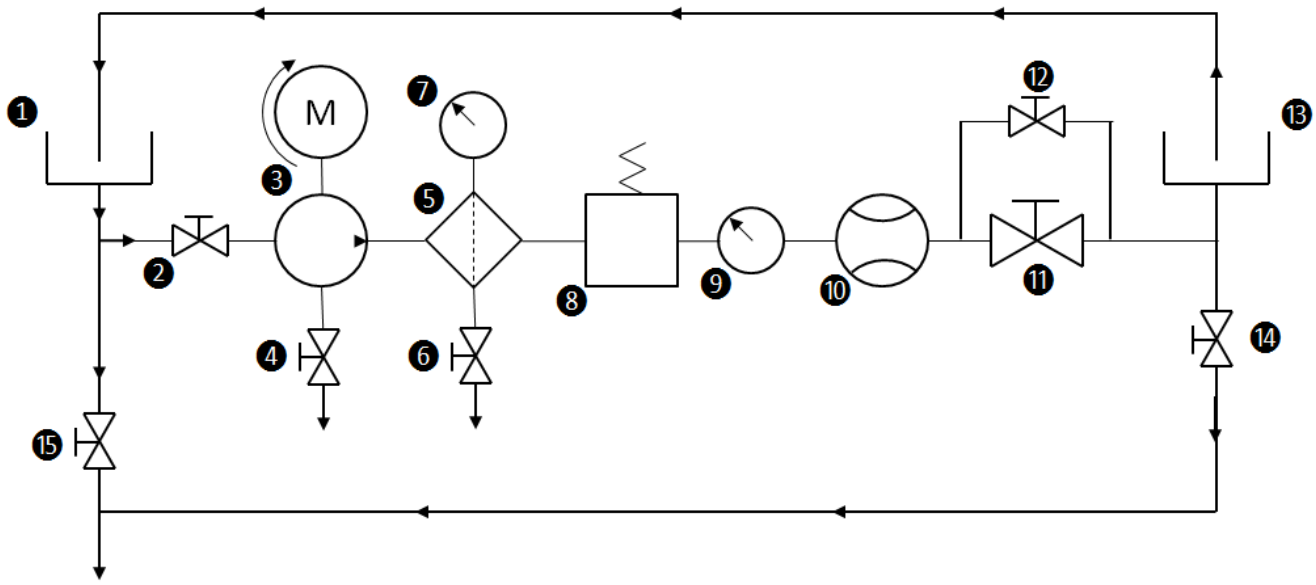






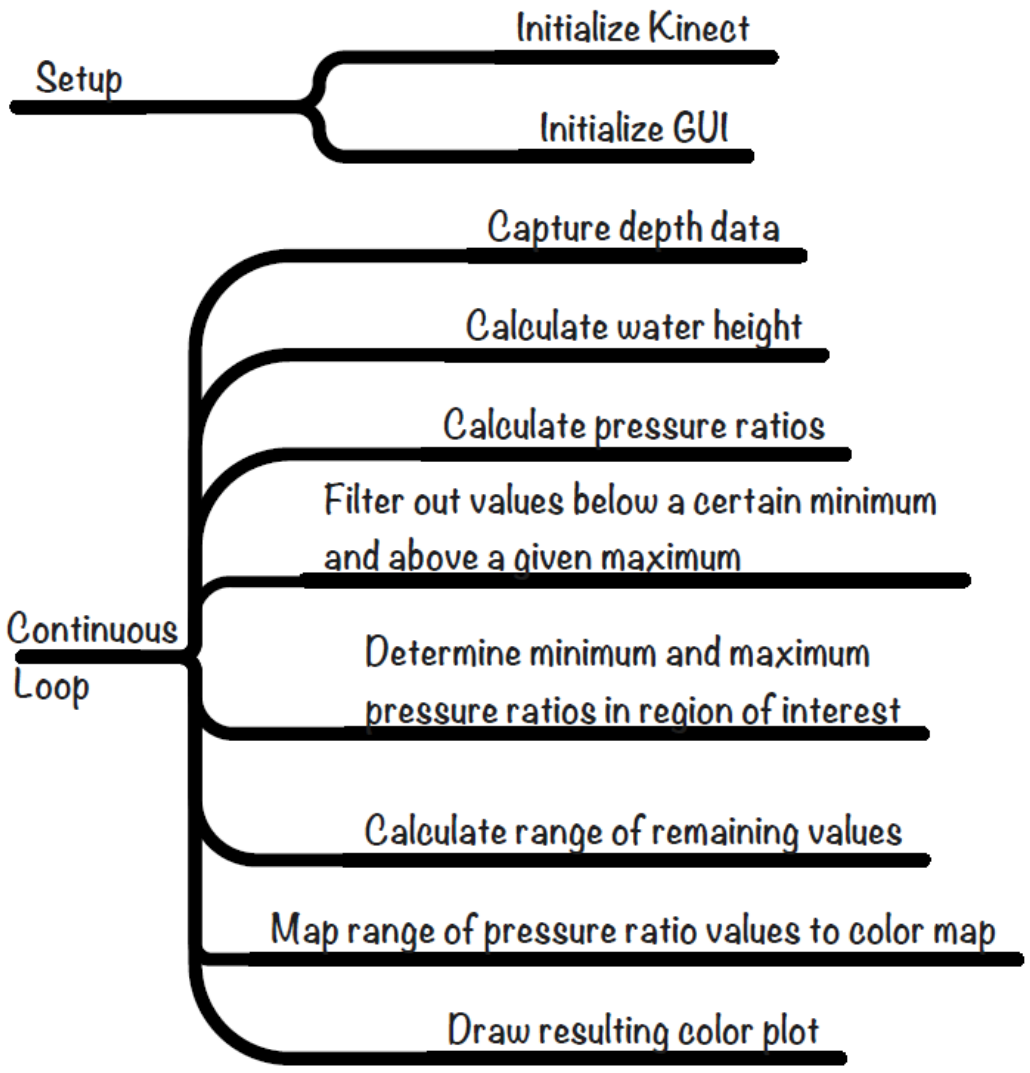


Appendix A2 – Water table circuit diagram



1. Exit reservoir
2. Pump throttle valve
3. Electric motor and hydraulic pump
4. Pump primer valve
5. Filter
6. Filter drain valve
7. Filter pressure gauge
8. Regulator
9. Flow meter pressure gauge
10. Flow meter
11. Main flow control valve
12. Precision flow control valve
13. Inlet reservoir
14. Inlet reservoir drain valve
15. Exit reservoir drain valve

Appendix A3 – Flow chart and source code for Water Table Microsoft Kinect Flow visualization




```
/*

 */

import g4p_controls.*;

import java.awt.Color;
import java.awt.Font;
import java.awt.TextArea;

import processing.event.MouseEvent;

/* Kinect */
import SimpleOpenNI.*;
```

```
SimpleOpenNI  context;
int     step   = 3;  // to speed up the drawing of the depth points, draw every third point
int minDepth = 10000 ;
int maxDepth = 0 ;

/* the x,y coordinates of the upper left and lower right
 corners of the selection region
 */
int regionMinX;
int regionMaxX;
int regionMinY;
int regionMaxY;

float minValue = 10000.0, maxValue = 0.0 ;

int max_depths = 4000;
int[] depth_histogram = new int[max_depths];

boolean depth_plot = true ;

float frame_rate ;
int last_update_time = -1 ;

float reference_water_height, kinect_height ;

// File I/O
PrintWriter depthsFile;

// GUI
GWindow cameraWindow, depthWindow, histogramWindow;
GButton btnMirrorOn, btnMirrorOff, btnCalcMinMaxDepth ;

GOption[] optMmessType;
GToggleGroup opgMmessType;
int md_mtype;

GOption optStep1, optStep2, optStep3, optStep4 ;
GOption optColorScale, optGreyScale ;
GOption optDepthPlot, optPressureRatioPlot ;

//GLabel lblValueMinMax, txtValueMinMax;
GLabel lblValueMinMax;
GTextField txtValueMinMax;
GLabel lblDepthMinMax;
GTextField txtDepthMinMax;

// Height variables
GLabel lblHeightKinect, lblReferenceWaterHeight;
GTextField txtHeightKinect, txtReferenceWaterHeight;
```

```
GButton btnUpdateHeight ;

// Which Min Max to use
GOption optComputedMinMax, optUserSetMinMax ;
GButton btnUserSetMinMax ;
GTextField txtUserSetMin, txtUserSetMax ;

// Upper Height Threshold
GLabel lblUpperHeightThreshold ;
GTextField txtUpperHeightThreshold ;
GButton btnUpperHeightThreshold ;
int upperHeightThreshold = 10000;

// Upper Height Threshold
GLabel lblRegionMinX,lblRegionMaxX,lblRegionMinY,lblRegionMaxY  ;
GTextField txtRegionMinX,txtRegionMaxX,txtRegionMinY,txtRegionMaxY ;
GButton btnRegionSet ;

// Frame rate
GLabel lblFrameRate;
GTextField txtFrameRate;

// Write to file
GButton btnWriteDepthsToFile ;
GTextField txtDepthsFileName;

// Interactive values
GTextField txtInteractiveX, txtInteractiveY, txtInteractiveDepth, txtInteractiveHeight,
txtInteractiveValue ;

/* array of colors for the color map */
color[] colors=new color[256];

void setup() {
  size(500, 1000); // main GUI window

  setup_openni() ;
  setup_gui() ;
  setup_windows() ;
  load_color_scale() ;
}

void setup_openni() {
  int depthHeight ;
  int depthWidth ;
  context = new SimpleOpenNI(this);

  // mirror is by default enabled
```

```
  context.setMirror(false);

  // enable depthMap generation
  if (context.enableDepth() == false)
  {
    println("Can't open the depthMap, maybe the camera is not connected!");
    exit();
    return;
  }
  if (context.enableRGB() == false)
  {
    println("Can't open the rgbMap, maybe the camera is not connected or there is no
rgbSensor!");
    exit();
    return;
  }

    // try and keep it at 30fps
  frameRate(30);

  calcMinMaxDepth() ;

  depthHeight = context.depthHeight() ;
  depthWidth = context.depthWidth() ;
  regionMinX = 0 ;
  regionMaxX = depthWidth - 1 ;
  regionMinY = 0 ;
  regionMaxY = depthHeight - 1 ;

}

void setup_windows() {
  // last param can be any of JAVA2D / P2D / P3D / OPENGL
  // JAVA2D better than P2D and P3D in terms of frame rate. OPENGL is good too
  cameraWindow = new GWindow(this, "Camera", 50, 100, context.rgbWidth() ,
context.rgbHeight() , false, OPENGL);
  cameraWindow.addDrawHandler(this, "cameraWindowDraw");
  cameraWindow.addMouseHandler(this, "cameraWindowMouseEventHandler");

  depthWindow = new GWindow(this, "Depth", 1200, 100, context.depthWidth(),
context.depthHeight(), false, JAVA2D);
  depthWindow.addDrawHandler(this, "depthWindowDraw");
  depthWindow.addMouseHandler(this, "depthWindowMouseEventHandler");
  //depthWindow.setBackground(Color.RED);

  histogramWindow = new GWindow(this, "Histogram", 1200, 900, 500, 300, false, JAVA2D);
  histogramWindow.addDrawHandler(this, "histogramWindowDraw");
  histogramWindow.addMouseHandler(this, "histogramWindowMouseEventHandler");
  histogramWindow.addPreHandler(this, "histogramWindowPreHandler");
```

```
}



void writeDepthsToFile() {
  int[]   depthMap = context.depthMap();
  int depthHeight = context.depthHeight() ;
  int depthWidth = context.depthWidth() ;
  int index ;
  int depth;

  depthsFile = createWriter(txtDepthsFileName.getText());
  depthsFile.println("X,Y,depth_in_mm");
  for (int y=0;y < depthHeight;y+=step) {
    for (int x=0;x < depthWidth;x+=step) {
      index = x + y * depthWidth;
      depth = depthMap[index] ;
      depthsFile.println( str(x) + "," + str(y) + "," + str(depth)) ;
    }
  }
  depthsFile.flush();  // Writes the remaining data to the file
  depthsFile.close();  // Finishes the file
}



void userSetMinMaxDepth() {
 minValue =  float( txtUserSetMin.getText() ) ;
 maxValue =  float( txtUserSetMax.getText() ) ;
}

void calcMinMaxDepth() {
  int[]   depthMap = context.depthMap();
  int depthHeight = context.depthHeight() ;
  int depthWidth = context.depthWidth() ;
  int depth, h ;
  int index ;
  float value ;

  for (int i=0; i < max_depths; i++) {
    depth_histogram[i] = 0 ;
  }
```

```
  minDepth = 10000 ;
  maxDepth = 0 ;

  println( "minDepthThreshold = " + str(minDepthThreshold ) );
  println( "maxDepthThreshold = " + str(maxDepthThreshold ) );

  minValue = 10000.0 ;
  maxValue = 0.0 ;
  for (int y=regionMinY;y <= regionMaxY;y+=step) {
    for (int x=regionMinX;x <= regionMaxX;x+=step) {
      index = x + y * depthWidth;
      depth = depthMap[index] ;
      if ( depth_plot ) {
        value = kinect_height - float( depth ) ;
      } else {
        value = pow( ( kinect_height - float( depth ) ) / reference_water_height, 2 );
      }
      //if ( depth > 0 && ( ( kinect_height - depth ) < upperHeightThreshold ) ) {
        if ( x == regionMinX + 100 * step && y == regionMinY + 100 * step ) println( "depth = " +
depth ) ;
      if ( depth > 0 && ( depth > minDepthThreshold ) && ( depth < maxDepthThreshold ) ) {
      //if ( depth > 0 ) {
        if ( value < minValue ) minValue = value ;
        if ( value > maxValue ) maxValue = value ;
      }
      if ( depth > 0 ) {
        h = int( kinect_height ) - depth ;
        depth_histogram[ h ] += 1 ;
        //println( str(h) + ":" + str( depth_histogram[ h ] ) ) ;
        if ( depth < minDepth ) minDepth = depth ;
        if ( depth > maxDepth ) maxDepth = depth ;
    }
  }
  }
  println( "min max value  " + str( minValue ) + "   " + str( maxValue ) ) ;
}



/**
 * Draw for the main window
 */
void draw() {
  //background(240);


  //context.update();
```

```
  //println("update");
  //cameraWindow.setBackground(context.rgbImage());
  //image(context.rgbImage(), 0, 0);

  // write minDepth and maxDepth
  //println( minValue ) ;
  //println( maxValue ) ;

  //txtValueMinMax.appendText( str( minValue ) + " to " + str( maxValue ) ) ;
  //txtValueMinMax.appendText( str( minDepth ) + " to " + str( maxDepth ) ) ;
  txtDepthMinMax.setText( str( minDepth ) + " to " + str( maxDepth ) ) ;
  txtValueMinMax.setText( str( minValue ) + " to " + str( maxValue ) ) ;
  //txtValueMinMax.setText( str( minDepth )  ) ;
  //txtValueMinMax.setText( "gleep" ) ;
  //txtValueMinMax.setText( str( minDepth ) ) ;

  //draw_depth_histogram() ;

}


///**
//Create the three windows so that they share mouse handling
//and drawing code.
//*/
//void createWindows() {
//  int col;
//  window = new GWindow[3];
//  for (int i = 0; i < 3; i++) {
//    col = (128 << (i * 8)) | 0xff000000;
//    window[i] = new GWindow(this, "Window "+i, 70+i*220, 160+i*50, 200, 200, false, JAVA2D);
//    window[i].setBackground(col);
//    window[i].addData(new MyWinData());
//    window[i].addDrawHandler(this, "windowDraw");
//    window[i].addMouseHandler(this, "windowMouse");
//  }
//}
//
///**
// * Click the button to create the windows.
// * @param button
// */
//void handleButtonEvents(GButton button, GEvent event) {
//  if (window == null && event == GEvent.CLICKED) {
//    createWindows();
//    lblInstr.setVisible(true);
//    button.setEnabled(false);
//  }
```

```
//}
//
///**
// * Handles mouse events for ALL GWindow objects
// *
// * @param appc the PApplet object embeded into the frame
// * @param data the data for the GWindow being used
// * @param event the mouse event
// */
//void windowMouse(GWinApplet appc, GWinData data, MouseEvent event) {
//  MyWinData data2 = (MyWinData)data;
//  switch(event.getAction()) {
//  case MouseEvent.PRESS:
//    data2.sx = data2.ex = appc.mouseX;
//    data2.sy = data2.ey = appc.mouseY;
//    data2.done = false;
//    break;
//  case MouseEvent.RELEASE:
//    data2.ex = appc.mouseX;
//    data2.ey = appc.mouseY;
//    data2.done = true;
//    break;
//  case MouseEvent.DRAG:
//    data2.ex = appc.mouseX;
//    data2.ey = appc.mouseY;
//    break;
//  }
//}
//
///**
// * Handles drawing to the windows PApplet area
// *
// * @param appc the PApplet object embeded into the frame
// * @param data the data for the GWindow being used
// */
//void windowDraw(GWinApplet appc, GWinData data) {
//  MyWinData data2 = (MyWinData)data;
//  if (!(data2.sx == data2.ex && data2.ey == data2.ey)) {
//    appc.stroke(255);
//    appc.strokeWeight(2);
//    appc.noFill();
//    if (data2.done) {
//      appc.fill(128);
//    }
//    appc.rectMode(CORNERS);
//    appc.rect(data2.sx, data2.sy, data2.ex, data2.ey);
//  }
//}
//
```

```
///**
// * Simple class that extends GWinData and holds the data
// * that is specific to a particular window.
// *
// * @author Peter Lager
// */
//class MyWinData extends GWinData {
//  int sx, sy, ex, ey;
//  boolean done;
//}


synchronized public void cameraWindowDraw(GWinApplet appc, GWinData data) {
  //cameraWindow.setBackground(context.rgbImage());
  appc.background(context.rgbImage());

  // draw_selection_region
  appc.stroke(color(255, 0, 0));
  appc.noFill();
  appc.rect( regionMinX, regionMinY, regionMaxX - regionMinX, regionMaxY - regionMinY ) ;
}


void cameraWindowMouseEventHandler(GWinApplet appc, GWinData data, MouseEvent
event){
  int x, y ;

    switch(event.getAction()){
      case MouseEvent.PRESS:
        x = event.getX() ;
        y = event.getY() ;
        regionMinX = x ;
        regionMinY = y ;
        break;
      case MouseEvent.RELEASE:
        x = event.getX() ;
        y = event.getY() ;
        regionMaxX = x ;
        regionMaxY = y ;
        break;
      case MouseEvent.CLICK:
        break;
      case MouseEvent.DRAG:
        x = event.getX() ;
        y = event.getY() ;
        regionMaxX = x ;
        regionMaxY = y ;
        break;
```

```
      case MouseEvent.MOVE:
        break;
    }
}



void load_grey_scale() {
    for (int i = 0; i < 256; i++) {
      colors[i] = color( float(i), float(i), float(i)) ;
    }
}

void load_color_scale() {
  String[] pieces ;
  String[] lines;

  //lines = loadStrings("ncl_default.rgb");
  lines = loadStrings("GMT_seis.rgb");
  for (int i = 0; i < lines.length; i++) {
    if ( i > 2 ) {
      pieces = splitTokens(lines[i], " "); // Load data into array
      colors[255-(i-3)] = color( float(pieces[0])*255, float(pieces[1])*255, float(pieces[2]) *255 ) ;
      //colors[i-3] = color( float(pieces[0])*255, float(pieces[1])*255, float(pieces[2]) *255 ) ;
//      if ( i == lines.length - 1 ) { // this color map only has 254 entries
//        colors[255-(i-2)] = color( float(pieces[0])*255, float(pieces[1])*255, float(pieces[2]) *255 ) ;
//        colors[255-(i-1)] = color( float(pieces[0])*255, float(pieces[1])*255, float(pieces[2]) *255 ) ;
//      }
    }
  }
}


void draw_color_scale() {
  int x, y;
  int colorScaleWidth = 30 ;

  x = 370 ;
    for (int i=0;i < 256;i++) {
        y = 10 + 256 - i;
        stroke(colors[i]);
        fill(colors[i] );
        line(x,y,x+colorScaleWidth,y);
    }
}



synchronized public void depthWindowDraw(GWinApplet appc, GWinData data) {
```

```
 int     index;

 int delta_in_milliseconds ;
 int time_in_milliseconds ;

 //println( "start of depthWindowDraw" ) ;

 context.update();

 if ( last_update_time < 0.0 ) { // First time
  last_update_time = millis();
 } else {
  time_in_milliseconds = millis() ;
  delta_in_milliseconds = time_in_milliseconds - last_update_time ;
  last_update_time = time_in_milliseconds ;
  //println("delta_in_milliseconds =" + str(delta_in_milliseconds));

  if ( delta_in_milliseconds > 0.0 ) {
   frame_rate = 1000.0 / float( delta_in_milliseconds ) ;

    try {
       txtFrameRate.setText( str( frame_rate ) ) ;
     } catch (Exception e){
       println("Error in setText with frame_rate = " + str(frame_rate ));
     }
  } else {
   txtFrameRate.setText( "infinity" ) ;
  }
  //println("after settext" ) ;
 }



 //An array containing all of the distances in millimetres
 int[]   depthMap = context.depthMap();
 int depth ;
 int levelColor ;

 float value ;

 int depthHeight = context.depthHeight() ;
 int depthWidth = context.depthWidth() ;
//  appc.background(100,100,200);
//  appc.fill(0,0,160);
//  appc.noStroke();
//  appc.ellipse(appc.width/2, appc.height/2, appc.width/1.2, appc.height/1.2);
//  appc.fill(255);
//  appc.text("Secondary window", 20, 20);
```

```
//So, we will have to record the height of the kinect, hk, with respect
// to the water table glass floor as well. Then we find the reference
// water height, hw. Measurement of surface distance from Kinect is hs and obtained from the image.
//Therefore, the pressure ratio is [(hk-hs)/(hw)]^2.

//depthHeight = 200 ;
//depthWidth = 200 ;

  for (int y=regionMinY;y <= regionMaxY;y+=step) {
  for (int x=regionMinX;x <= regionMaxX;x+=step) {
    index = x + y * depthWidth;
    depth = depthMap[index] ;
    if ( depth > 0 && ( ( kinect_height - depth ) < upperHeightThreshold ) ) {
      //if ( depth < minDepth ) minDepth = depth ;
      //if ( depth > maxDepth ) maxDepth = depth ;

      depth = depthMap[index] ;
      if ( depth_plot ) {
        value = kinect_height - float( depth ) ;
      } else {
        value = pow( ( kinect_height - float( depth ) ) / reference_water_height, 2 );
      }

      /* Draw the depth value as a color */
      //levelColor   = (int) map( depth, minDepth, maxDepth, 0, 255 );
      levelColor   = (int) map( value, minValue, maxValue, 0, 255 );
      //levelColor   = (int) map( value, minValue, maxValue, 255, 0 );
      if ( levelColor < 0 ) {
        levelColor = 0 ;
      }
      if (levelColor > 255 ) {
        levelColor = 255 ;
      }
      appc.stroke(colors[levelColor]);
      appc.fill(colors[levelColor] );
      //point(x,y);
      appc.ellipse(x, y, 4, 4) ; // use a fat point to make up for the fact that only drawing every third point
    }
    else {
    }
  }
  }
  //println( "end of depthWindowDraw" ) ;

}
void depthWindowMouseEventHandler(GWinApplet appc, GWinData data, MouseEvent event){
```

```
  int x, y ;
  int index ;
  int depth ;
  float value ;
  int depthWidth = context.depthWidth() ;
  int depthHeight = context.depthHeight() ;
  int[]   depthMap = context.depthMap();

  x = event.getX() ;
  if ( x < 0 ) x = 0 ;
  if ( x >= depthWidth ) x = depthWidth - 1 ;
  y = event.getY() ;
  if ( y < 0 ) y = 0 ;
  if ( y >= depthHeight ) y = depthHeight - 1 ;

  index = x + y * depthWidth;
  depth = depthMap[index] ;
  if ( depth_plot ) {
    value = kinect_height - float( depth ) ;
  } else {
    value = pow( ( kinect_height - float( depth ) ) / reference_water_height, 2 );
  }

  txtInteractiveX.setText( str( x ) ) ;
  txtInteractiveY.setText( str( y ) ) ;
  txtInteractiveDepth.setText( str( depth ) ) ;
  txtInteractiveHeight.setText( str( kinect_height - depth ) ) ;
  txtInteractiveValue.setText( str( value ) ) ;
  //println( str(x) + "," + str(y) + "," + str(depth)) ;

}




void setup_gui() {

  // Mirroring
  btnMirrorOn = new GButton(this, 10, 10, 100, 20, "Mirror On");
  btnMirrorOff = new GButton(this, 120, 10, 100, 20, "Mirror Off");

  int x,y ;
  int xData = 200;
  x = 0 ;
  y = 30 ;

  // Step size
```

```
  GToggleGroup tgStep = new GToggleGroup();
  GLabel lblTestStyle ;

  lblTestStyle = new GLabel(this, x, y, 80, 18, "Step Size");
  lblTestStyle.setTextBold();
  optStep1 = new GOption(this, x, y + 20, 80, 18, "1");
  optStep2 = new GOption(this, x, y + 40, 80, 18, "2");
  optStep3 = new GOption(this, x, y + 60, 80, 18, "3");
  optStep4 = new GOption(this, x, y + 80, 80, 18, "4");
  tgStep.addControls(optStep1, optStep2, optStep3, optStep4);
  optStep3.setSelected(true);

  // Color map
  GToggleGroup tgColorMap = new GToggleGroup();
  GLabel lblColorMap ;
  lblColorMap = new GLabel(this, x, y + 120, 80, 18, "Color Map");
  lblColorMap.setTextBold();
  optGreyScale = new GOption(this, x, y + 140, 80, 18, "grey");
  optColorScale = new GOption(this, x, y + 160, 80, 18, "color");
  tgColorMap.addControls(optGreyScale, optColorScale);
  optColorScale.setSelected(true);

  // Calc min max button
  btnCalcMinMaxDepth = new GButton(this, x, y + 200 , 160, 20, "Calc Min Max Depth");

  // Height reference variables
  lblHeightKinect = new GLabel(this, x,  y + 240, 140, 18, "Kinect Height (mm)" );
  lblHeightKinect.setTextBold();
  txtHeightKinect = new GTextField(this, x,  y + 260, 160, 18 );
  txtHeightKinect.setText( "4000.0" ) ;

  lblReferenceWaterHeight = new GLabel(this, x,  y + 300, 200, 18, "Reference Water Height
(mm)" );
  lblReferenceWaterHeight.setTextBold();
  txtReferenceWaterHeight = new GTextField(this, x,  y + 320, 160, 18 );
  txtReferenceWaterHeight.setText( "100.0" ) ;
  btnUpdateHeight = new GButton(this, x,  y + 360, 100, 20, "Update Heights");
  kinect_height = float( txtHeightKinect.getText() ) ;
  reference_water_height = float( txtReferenceWaterHeight.getText() ) ;

  // Plot depth or pressure ratio?
  GToggleGroup tgPlotValue = new GToggleGroup();
  GLabel lblPlotValue ;
  lblPlotValue = new GLabel(this, x, y+400, 80, 18, "Plot Value");
  lblPlotValue.setTextBold();
  optDepthPlot = new GOption(this, x, y + 420, 80, 18, "Height");
  optPressureRatioPlot = new GOption(this, x, y + 440, 140, 18, "Pressure Ratio");
  tgPlotValue.addControls(optDepthPlot, optPressureRatioPlot);
  optDepthPlot.setSelected(true);
```

```
  // Write depths to file
  btnWriteDepthsToFile = new GButton(this, x, y + 480, 140, 18, "Write Depths to File");
  txtDepthsFileName = new GTextField(this, x,  y + 510, 160, 18 );
  txtDepthsFileName.setText( "depths.csv" ) ;

  // Upper Height Threshold
  btnUpperHeightThreshold = new GButton(this, x,      y + 680, 150, 20, "Upper Height
Threshold");
  lblUpperHeightThreshold = new GLabel(this, x, y+710, 40, 18, "ht th");
  lblUpperHeightThreshold.setTextBold();
  txtUpperHeightThreshold = new GTextField(this, x + 40 ,  y + 710, 160, 18 );
  txtUpperHeightThreshold.setText( str( upperHeightThreshold ) ) ;

 // Region Set
  btnRegionSet = new GButton(this, x,   y + 750, 150, 20, "Set Region");
  lblRegionMinX = new GLabel(this, x, y+770, 40, 18, "MinX");
  lblRegionMinX.setTextBold();
  txtRegionMinX = new GTextField(this, x + 40 ,  y + 770, 160, 18 );
  txtRegionMinX.setText( str( regionMinX ) ) ;

  lblRegionMaxX = new GLabel(this, x, y+790, 40, 18, "MaxX");
  lblRegionMaxX.setTextBold();
  txtRegionMaxX = new GTextField(this, x + 40 ,  y + 790, 160, 18 );
  txtRegionMaxX.setText( str( regionMaxX ) ) ;

  lblRegionMinY = new GLabel(this, x, y+810, 40, 18, "MinY");
  lblRegionMinY.setTextBold();
  txtRegionMinY = new GTextField(this, x + 40 ,  y + 810, 160, 18 );
  txtRegionMinY.setText( str( regionMinY ) ) ;

  lblRegionMaxY = new GLabel(this, x, y+830, 40, 18, "MaxY");
  lblRegionMaxY.setTextBold();
  txtRegionMaxY = new GTextField(this, x + 40 ,  y + 830, 160, 18 );
  txtRegionMaxY.setText( str( regionMaxY ) ) ;


   // Plot depth or pressure ratio?
//   GToggleGroup tgMinMaxChoice = new GToggleGroup();
//   GLabel lblValueMinMaxChoice ;
//   lblValueMinMaxChoice = new GLabel(this, x, y+550,120, 18, "Which Min Max?");
//   lblValueMinMaxChoice.setTextBold();
//   optComputedMinMax = new GOption(this, x, y + 570, 80, 18, "Computed");
//   optUserSetMinMax = new GOption(this, x, y + 590, 80, 18, "User Set");
//   tgMinMaxChoice.addControls(optComputedMinMax, optUserSetMinMax);
//   optComputedMinMax.setSelected(true);

   btnUserSetMinMax = new GButton(this, x, y + 570, 140, 18, "Use User Set Min Max");
```

```
  GLabel lblUserSetMin, lblUserSetMax;
  lblUserSetMin = new GLabel(this, x, y+610,30, 18, "Min");
  lblUserSetMin.setTextBold();
  lblUserSetMax = new GLabel(this, x, y+630,30, 18, "Max");
  lblUserSetMax.setTextBold();
  txtUserSetMin = new GTextField(this, x + 40,  y + 610, 160, 18 );
  txtUserSetMax = new GTextField(this, x + 40,  y + 630, 160, 18 );

  ///////// Outputs /////////
  // updated min and max values
  lblDepthMinMax = new GLabel(this, xData,  y + 320, 140, 18, "Depth min max" );
  lblDepthMinMax.setTextBold();
  txtDepthMinMax = new GTextField(this, xData,  y + 350, 160, 18 );

  lblValueMinMax = new GLabel(this, xData,  y + 420, 140, 18, "Value min max" );
  lblValueMinMax.setTextBold();
  txtValueMinMax = new GTextField(this, xData,  y + 450, 160, 18 );
  //txtValueMinMax = new GTextArea(this, xData, y + 50, 160, 100,
TextArea.SCROLLBARS_VERTICAL_ONLY );
  //txtValueMinMax.setFont(new Font("Dialog", Font.PLAIN, 14));

  // frame rate
  lblFrameRate = new GLabel(this, xData,  y + 220, 140, 18, "Frame Rate" );
  lblFrameRate.setTextBold();
  txtFrameRate = new GTextField(this, xData,  y + 240, 140, 18);

  // interactive values of X,Y, Depth, Height, and Value
  GLabel lblInteractiveX = new GLabel(this, xData, y+610,50, 18, "X");
  lblInteractiveX.setTextBold();
  txtInteractiveX = new GTextField(this, xData + 50,  y + 610, 160, 18 );

  GLabel lblInteractiveY = new GLabel(this, xData, y+630,50, 18, "Y");
  lblInteractiveY.setTextBold();
  txtInteractiveY = new GTextField(this, xData + 50,  y + 630, 160, 18 );

  GLabel lblInteractiveDepth = new GLabel(this, xData, y+650,50, 18, "Depth");
  lblInteractiveDepth.setTextBold();
  txtInteractiveDepth = new GTextField(this, xData + 50,  y + 650, 160, 18 );

  GLabel lblInteractiveHeight = new GLabel(this, xData, y+670,50, 18, "Height");
  lblInteractiveHeight.setTextBold();
  txtInteractiveHeight = new GTextField(this, xData + 50,  y + 670, 160, 18 );

  GLabel lblInteractiveValue = new GLabel(this, xData, y+690,50, 18, "Value");
  lblInteractiveValue.setTextBold();
  txtInteractiveValue = new GTextField(this, xData + 50,  y + 690, 160, 18 );

  draw_color_scale() ;
```

```
}
public void handleTextEvents(GEditableTextControl textarea, GEvent event) {
  /* nothing to do but this prevents warning messages */
}


// This method is called when a button is clicked
void handleButtonEvents(GButton button, GEvent event){
  if(button == btnMirrorOn && event == GEvent.CLICKED)
     context.setMirror(true);
  if(button == btnMirrorOff && event == GEvent.CLICKED)
     context.setMirror(false);
  if(button == btnCalcMinMaxDepth && event == GEvent.CLICKED)
     calcMinMaxDepth();
  if(button == btnUpdateHeight && event == GEvent.CLICKED)
    kinect_height = float( txtHeightKinect.getText() ) ;
    reference_water_height = float( txtReferenceWaterHeight.getText() ) ;
    //println( reference_water_height ) ;
    //println( kinect_height ) ;

  if(button == btnWriteDepthsToFile && event == GEvent.CLICKED)
     writeDepthsToFile();

  if(button == btnUserSetMinMax && event == GEvent.CLICKED)
     userSetMinMaxDepth();

  if(button == btnUpperHeightThreshold && event == GEvent.CLICKED)
     upperHeightThreshold = int( txtUpperHeightThreshold.getText() ) ;

  if(button == btnRegionSet && event == GEvent.CLICKED){
     regionMinX = int( txtRegionMinX.getText() ) ;
     regionMaxX = int( txtRegionMaxX.getText() ) ;
     regionMinY = int( txtRegionMinY.getText() ) ;
     regionMaxY = int( txtRegionMaxY.getText() ) ;
  }


}

// Called when a toggle button event happens
public void handleToggleControlEvents(GToggleControl option, GEvent event) {
  if (option == optStep1)
    step = 1 ;
  else if (option == optStep2)
    step = 2 ;
  else if (option == optStep3)
    step = 3 ;
  else if (option == optStep4)
```

```
    step = 4 ;

  if (option == optColorScale) {
    load_color_scale() ;
    draw_color_scale() ;
  } else if (option == optGreyScale) {
    load_grey_scale() ;
    draw_color_scale() ;
  }

  if (option == optDepthPlot )
     depth_plot = true ;
  else if (option == optPressureRatioPlot )
     depth_plot = false ;

  if (option == optComputedMinMax )
     depth_plot = true ;
  else if (option == optUserSetMinMax )
     depth_plot = false ;


}




GTextField txtMinDepth, txtMaxDepth, txtMinRangeDepth, txtMaxRangeDepth ;

GSlider minDepthSlider, maxDepthSlider ;

int minDepthThreshold, maxDepthThreshold ;


int y_bottom = 250 ;
int y_max_height = 50 ;
int x_begin = 40;
int x_end =  480 ;
int txt_width = 40 ;
boolean gui_elements_drawn = false;

float minDepthSliderValue, maxDepthSliderValue ;

synchronized public void histogramWindowPreHandler(GWinApplet appc, GWinData data) {
  if ( ! gui_elements_drawn ) {
    txtMinDepth = new GTextField(appc, x_begin - ( txt_width /2 ),  y_bottom + 30, txt_width, 18 );
    txtMaxDepth = new GTextField(appc, x_end - ( txt_width/2 ),  y_bottom + 30, txt_width, 18 );
    txtMinRangeDepth = new GTextField(appc, x_begin + ( x_end - x_begin ) / 4,  y_bottom + 30,
txt_width, 18 );
```

```
   txtMaxRangeDepth = new GTextField(appc, x_begin + 3 * ( x_end - x_begin ) / 4,  y_bottom +
30, txt_width, 18 );
   // last parameter is the slider track thickness
   minDepthSlider = new GSlider(appc,x_begin,y_bottom +0 ,x_end - x_begin,10,10);
   minDepthSlider.setValue( 0.0 );
   maxDepthSlider = new GSlider(appc,x_begin,y_bottom +10 ,x_end - x_begin,10,10);
   maxDepthSlider.setValue( float( max_depths )  );
   gui_elements_drawn = true ;
  }
}

synchronized public void histogramWindowDraw(GWinApplet appc, GWinData data) {
  int y_height ;
  int x ;
  int max_depth_histogram = 0 ;

  appc.stroke( color( 0,0,0 ) );
  appc.fill( color(0,0,0) );

  // get the height of the highest bar in the histogram so we can scale the plot
   for (int i=0;i < max_depths;i++) {
   if ( depth_histogram[i] > max_depth_histogram ) max_depth_histogram = depth_histogram[i] ;
  }

  for (int i=0;i < max_depths;i++) {
    x = int( float(i) / float(max_depths) * float( x_end - x_begin )  ) ;
    y_height = int( float( depth_histogram[i] ) / float( max_depth_histogram ) * float(y_max_height)
) ;
    appc.line( x + x_begin, y_bottom, x + x_begin, y_bottom - y_height ) ;
  }
  txtMinDepth.setText( str(0) ) ;
  txtMaxDepth.setText( str(max_depths) ) ;

   // draw min max lines
   appc.stroke( color( 255,0,0 ) );
   appc.fill( color(255,0,0) );
   x = int( x_begin + ( x_end - x_begin ) * minDepthSliderValue );
   appc.line( x, y_bottom, x, y_max_height ) ;

   appc.stroke( color( 0,255,0 ) );
   appc.fill( color(0,255,0) );
   x = int( x_begin + ( x_end - x_begin ) * maxDepthSliderValue );
   appc.line( x, y_bottom, x, y_max_height ) ;

}
```

```
void histogramWindowMouseEventHandler(GWinApplet appc, GWinData data, MouseEvent event){

  int x, y ;
  int index ;
  int depth ;

  x = event.getX() ;
  y = event.getY() ;

  depth = int( float(x - x_begin) / (x_end - x_begin ) * max_depths ) ;
  //txtDepth.setText( str(depth) ) ;

}

public void handleSliderEvents(GValueControl slider, GEvent event) {

  if (slider == minDepthSlider) {
    minDepthSliderValue = minDepthSlider.getValueF();
    minDepthThreshold = int( max_depths * minDepthSliderValue );
    txtMinRangeDepth.setText( str(minDepthThreshold) ) ;
  }
  else if (slider == maxDepthSlider) {
    maxDepthSliderValue = maxDepthSlider.getValueF();
    maxDepthThreshold = int( max_depths * maxDepthSliderValue );
    txtMaxRangeDepth.setText( str(maxDepthThreshold) ) ;
  }
}
```