1-1-2013

Investigation Of Pressure Fluctuations In The Hyporheic Zone In Response To Flow Around A Hydraulic Structure

Timothy Calappi
Wayne State University,

INVESTIGATION OF PRESSURE FLUCTUATIONS IN THE HYPORHEIC ZONE IN RESPONSE TO FLOW AROUND A HYDRAULIC STRUCTURE

by

TIMOTHY J. CALAPPI

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2013

MAJOR: CIVIL ENGINEERING

Approved by:

_____

Advisor Date

_____

_____

_____

DEDICATION

This work is dedicated to Rebecca, Madeline and Colton, thanks for understanding the many hours, nights and weekends spent on this project. Without your support this manuscript would not have made it this far. A good many thanks are also owed to my mother, Carol. Her weekly visits to help with the kids giving me a few more hours to work on this project are greatly appreciated.

Rebecca's patience while proof reading much of this document are also appreciated. She now knows more about bridge crud (as she calls it) and I know more about punctuation.

ACKNOWLEDGEMENTS

The Wayne State University Department of Civil and Environmental Engineering funded this work. This research was conducted in a Hydraulics Laboratory at the University of Windsor in Ontario, many thanks are owed to the department and staff for their guidance and hospitality.

Contributions from my advisor and committee are appreciated, this is truly a better manuscript because of their thoughts, recommendations and insights. A special thanks is owed to my advisor, Professor Carol Miller, for her patience and letting me explore a research topic well outside the original intent.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

The surface of the earth is constantly evolving. Surface processes driven by water (including ice), wind and gravity shape and reshape everything from the hardest rocks to the softest minerals. The removal of soil, sediment and rock by surface processes is generally termed erosion, but has more specific names depending on context. For example, sediment transport describes the general movement of sediment from source to sink, either with water- or wind-borne currents. When rivers intersect either man-made or natural structures, the flowing water accelerates due to a reduction in cross-sectional area and discharge continuity; this leads increases in the range of pressure fluctuations and erosion around the structure. This process will subsequently be referred to as *scour* in this dissertation. Scour around a circular cylinder is the focus of this work.

Much of the science of sediment transport in current use is empirical. Of these, the most famous is the work of Albert Shields, which relates shear stress on a particle to incipient motion. The *Shields relationship*, like many other sediment transport relationships, requires identification of a critical parameter below which little or no sediment movement takes place. Understanding the intricacies of sediment transport requires knowledge of actual forces applied to individual particles. Over the last 20 years, computers have become more powerful and sensors smaller and more precise. This combination allows more detailed numerical modeling, as well as increases the precision of measurements of the forces at work in sediment transport processes.

Fluid velocity and pressure are jointly responsible for the hydrodynamic forces leading to erosion of particles (Detert, *et al.* 2010b). It is necessary to understand this joint relationship to gain a detailed understanding of the mechanisms contributing to bed instabilities (Detert, *et al.* 2010b). Knowledge of these forces and their origins increases the understanding of erosion on a particle scale.

This dissertation effort focuses on measuring the subsurface (i.e. beneath the water-sand interface) pressure field in response to riverine flow around a vertical cylinder; multi-span bridges are frequently supported with cylindrical piers. Synchronous recordings of fluid velocity and subsurface pressure measurements are used to derive a probability distribution of the subsurface pressure field around the cylinder. Under favorable transport conditions, subsurface pressure fields supply additional apparent lift on individual particles, destabilizing and mobilizing the bed. This occurs when localized zones of high-pressure fluctuation develop in a relatively deep stratum while a localized zone of low-pressure fluctuation appears in a more elevated stratum. In order to accommodate the pressure sensors in this experiment, they are placed inside a cylinder. This placement allows for subsurface pressure measurements with only limited disturbance of the ambient flow field. Since flow around a cylinder is well studied, results from previous experiments aid in analysis of this work. Results from this study are applicable to the fields of sediment transport, hyporheic exchange, bed armoring and, due to the cylindrical obstruction, local pier scour.

Objectives

1. Compile experimental data consisting of near-bed, three-dimensional velocities synchronously recorded with subsurface pressure measurements within a uniform sand bed.
2. Identify dominant processes relating velocity and subsurface pressure.
3. Map the subsurface pressure field around a cylinder and relate it to surface water velocity measurements using a probability distribution function.
4. Generate data with sufficient spatial and temporal resolution to support subsequent numerical modeling.

Hypotheses

1.) Pressure decay follows an exponential decay described by Detert and Parker (2010), Bregum et al.(2006) and Vollmer et al. (2002)

2.) Pressure decay varies with radial position

3.) Pressure is most correlated with vertical velocity fluctuations at the zero degree, and the streamwise velocity component at the 45 and 90 degree positions.

Motivation

Riverbed scour is a continuous process with natural and anthropomorphic causes. Local accelerations in river velocity increase the ability for a river to erode sediment. Bridge support structures at river crossings create local acceleration. Removing enough sediment from the river bottom near bridge piers or abutments can cause the bridge to become unstable, increasing the risk of failure.

According to the Federal Highway Administration, the United States has approximately 600,000 bridges; about 80 percent require some sort of scour mitigation (Nassif, *et al.* 2002). Due to uncertainty in current scour prediction equations, ultimate scour depth is typically overestimated to ensure safety. While the incremental cost for deeper foundations may be reasonable for a small bridge with a single pier, it can be exorbitant for larger bridges with several large-diameter piers. Decreasing uncertainty associated with scour-prediction models can lead to cheaper construction costs without sacrificing safety.

Over the last few decades, statistical and physical modeling dominated scour research with the goal of relating hydrodynamics, geometry and sediment data to scour depth. In practice, scour is estimated with empirical equations with functional forms dependent on surficial processes.

As seen throughout this work, recent research in the field of hyporheic exchange processes has helped shed new light on the force balance present at incipient motion on the individual grain scale. Until now, investigations into hyporheic pressure fluctuations have not included hydraulic structures. Including a vertical cylinder in this work is necessary due to the grain size (but it also serves to extend the ideas and applications developed for nutrient exchange and auxiliary lift created by pressure fluctuations in the hyporheic zone into a new area of study.

Background

Turbulent Velocity Decomposition

Turbulent velocities are often described by horizontal (u) and vertical (v) velocity components. Horizontal velocity is often described by the streamwise component and assigns positive in the downstream direction. Vertical velocities are positive in the upward direction. Additionally, each component is described by a mean ($\bar{u}$ and $\bar{v}$) and a fluctuating velocity (u' and v'). The instantaneous velocity is the sum of the mean and fluctuating velocities. For example, the total instantaneous horizontal velocity u =$\bar{u} + u'$. When u' is positive and v' is negative the turbulent event is termed a *sweep*. A positive value of v' and a negative value of u' defines an *ejection*. Sweeps and ejections along with outward and inward interactions describe a phenomenon called turbulent bursting. Figures 1.1 and 1.2 taken from Keshavarzy and Ball (1997) provide further illustration of turbulent bursting. The discovery of turbulent bursting in 1967 led to new interest in the link between boundary layer turbulence and sediment entrainment (Dey and Papanicolaou 2008).

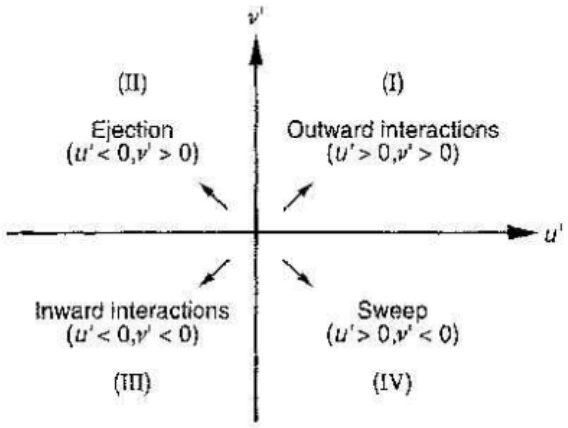


Figure1.1 Four types of turbulent bursting events (Keshavarzy and Ball 1997)

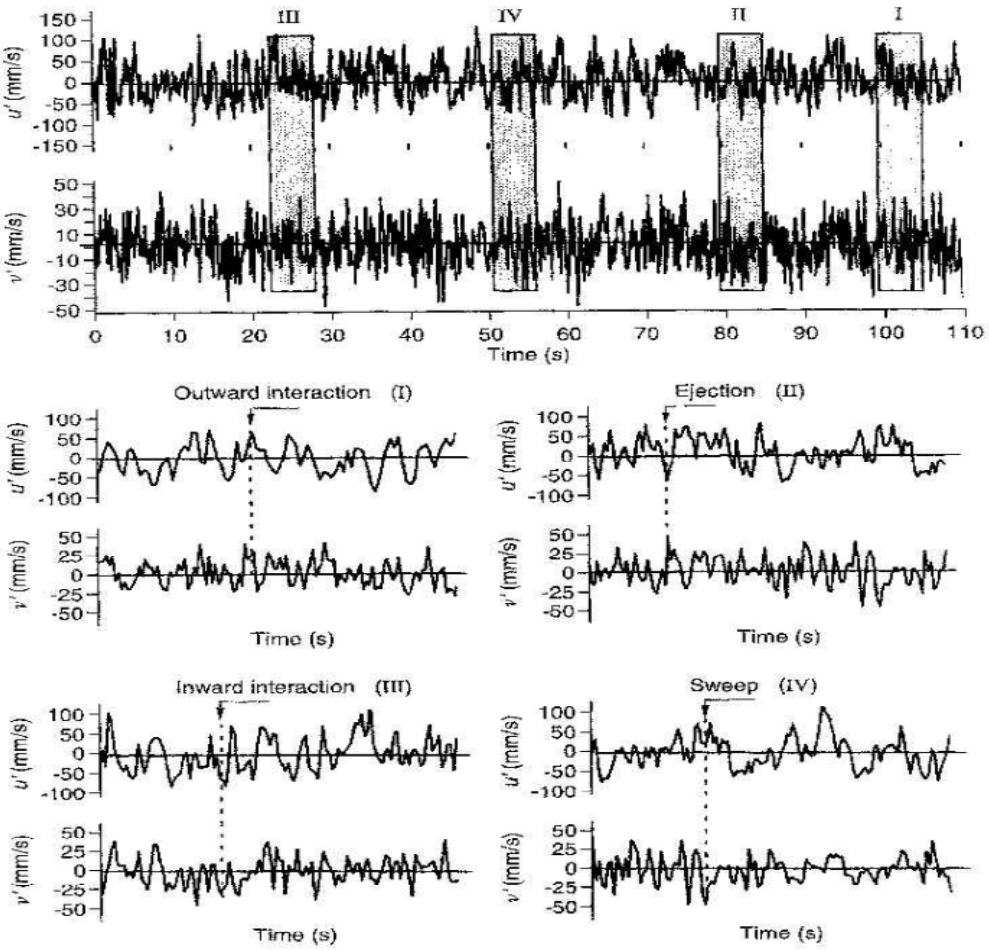


Figure 1.2 Examples of the four types of turbulent bursting events. Taken from (Keshavarzy and Ball 1997)

Flow around a cylinder

Characteristics of the velocity field when a fluid flows past a cylinder is Reynolds number dependent. When the Reynolds number is less than one, the flow has upstream-downstream symmetry. Between Reynolds number of 5 to 40, vortices form in the wake of the cylinder but remain attached to the cylinder. At Reynolds numbers approaching 100, the vortices begin to peel off the cylinder in a regular and periodic manner but top to bottom symmetry is gone. This trail of vortices is termed a Karman street. The Karman

street is considered laminar until a Reynolds number of 200. When the Reynolds number approaches 400, low levels of turbulence appears, but the periodic behavior remains strong. At Reynolds numbers approaching 10^5 a fully turbulent wake has formed; however, coherent vortex shedding is still detectable (Davidson 2009).

Figure 1.3 shows streamlines around a cylinder. These streamlines represent an ideal fluid, at low Reynolds number. Actual streamlines are a function of velocity and specific geometry and do not always remain attached.

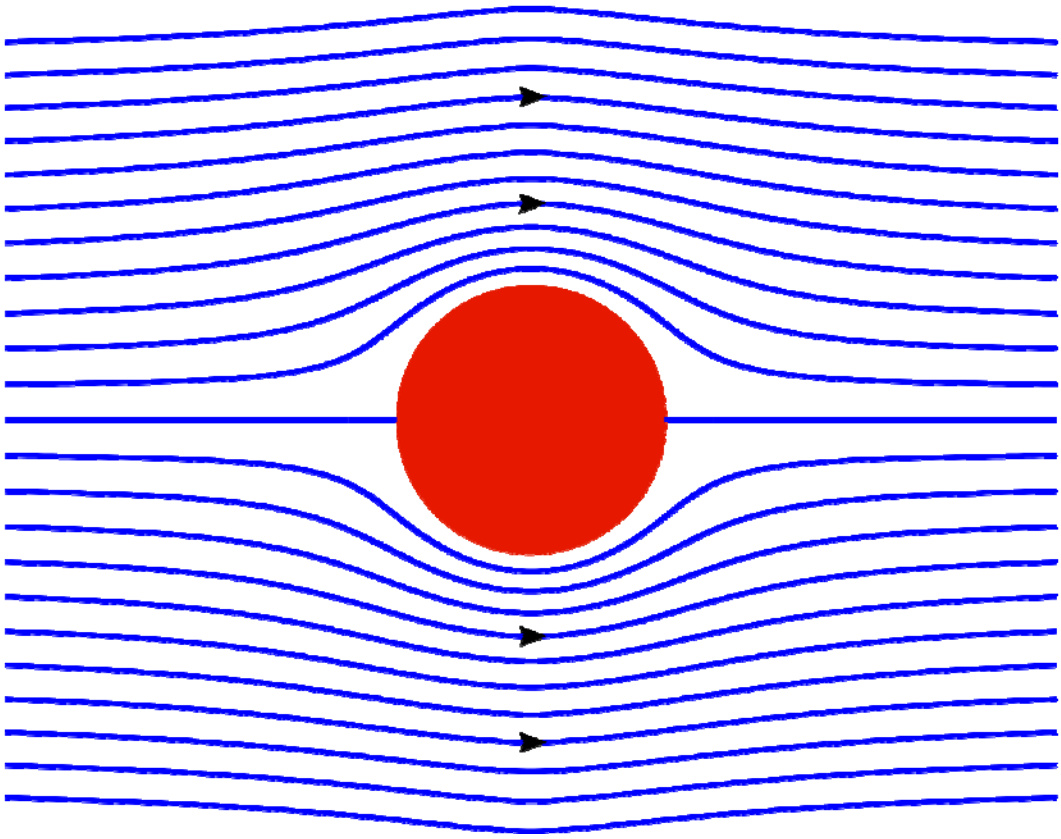


Figure 1.3: Streamlines of flow around a stationary cylinder.

Chapter 2: Literature Review

Experimental work in the field of sediment transport began in the 1930s. One of the earliest and most important was the work of Albert Shields. Shields' research related critical shear stress to incipient motion of a single, uniformly distributed, noncohesive grain. To this day, the Shields diagram is widely used. Since the mid 1960s, many modifications have been proposed to account for both smooth and rough regimes as well as probabilistic approaches (Dey and Papanicolaou 2008). Concurrent with Shields' work, Jeffreys (1929) showed hydrodynamic forces could provide a necessary lift to initiate sediment motion. By 1939, Lane and Kalinske suggested particles near the bed experience lift if the settling velocity is less than the turbulent vertical velocity fluctuations (Dey and Papanicolaou 2008). However, according to Schmeeckle and Nelson (2003) the understanding of lift on a particle in a sediment bed remains poorly understood.

Measurements of hydrodynamic forces on particles at the surface of a rough bed with pressure sensors stretch back to 1949 when Hans Einstein glued three-inch hemispheres to the bottom of a flume. Einstein and El-Samni (1949) observed the behavior of sediment particles in motion and concluded large instantaneous variations of lift force must be present. This variation influences bed stability and measurements of the fluctuations were deemed important. Einstein and El-Samni noted pressure distributions around a particle correlate with roughness height so any convenient particle diameter could be used to statistically describe the pressure variation around particles. Instantaneous lift was calculated based on pressure measurements made at the top and bottom of individual hemispheres. These measurements were made over a rough, impermeable bed. Einstein and El-Samni (1949) used their measured data to determine an

empirical relationship between the average lift pressure (pressure difference between top and bottom of the particle) and velocity.

In the past, turbulence characteristics were examined for rough and smooth surfaces only. However, roughness is not the only thing effecting turbulence characteristics. More recently, researchers examined effects of turbulence characteristics over permeable beds in conjunction with roughness variation. Experiments by Ruff and Glehar (1972) and Zagni and Smith (1976) indicate the effect of wall (for the purposes of this work the bed is also a wall) permeability on turbulence is different from the effect of roughness (Breugem, *et al.* 2006). This implies permeability alters the structure and dynamics of turbulence (Breugem, *et al.* 2006) and opposes Einstein's claim that turbulent pressure fields scale with roughness height only. The relationship between velocity and pressure can be complex and bed permeability only increases the complexity (Smart and Habersack 2007). Numerical solutions can more easily separate the effect of roughness and permeability (Breugem, *et al.* 2006). Breugem accomplished this separation by choosing parameters such that the Reynolds number based on roughness is very small, and the Reynolds number based on permeability is relatively large. Breugem used direct numerical simulations to study the effect of permeability on turbulence characteristics. The mean velocity profile decreased exponentially through the porous subsurface. The exponential decay is governed by the balance between turbulent diffusion of momentum into a permeable wall and the removal of momentum by drag (Breugem, *et al.* 2006). Breugem further suggests displacement and equivalent roughness heights found through mean velocity log-law fits might be outside the range found in literature. This implies a relation between turbulence characteristics and permeability as well as roughness (Breugem, *et al.* 2006). A well-known feature of flow past an impermeable wall is the low-and high-speed streaking associated with the streamwise velocity.

Breugem states a reduction in shear caused by a relaxation of the no-slip condition at the wall explains the absence of these streaks. The absence of the streaks leads to a reduction in the streamwise root mean square velocity. Breugem's work also shows turbulent motion inside the wall is induced by pressure fluctuations and does not contribute to the Reynolds shear stress (Breugem, *et al.* 2006).

While permeability effects on turbulence are important in any porous media work, it is often difficult to separate those effects from roughness. Manes et al. (2011) covered a flume interior with a high-density foam with relatively large pores compared to the filament thickness. This material allowed for high permeability while minimizing the effects from roughness. Manes concluded flow resistance and depth of shear penetration increased with an increasing Reynolds number ($\frac{\sqrt{k}u_*}{\nu}$) where k is the permeability, u_* is the friction velocity and ν is the kinematic viscosity of water. Manes also shows the depth of shear penetration defines the characteristic inner length scale as well as helps define the zero plane position for velocity profiles. He then uses quadrant analysis to describe permeability effects. The study shows turbulent sweeps dominate the near-wall region. Farther away from the wall, ejections dominate. Inward and outward interactions are progressively filtered out with increasing permeability (Manes, *et al.* 2011). Like Breugem, Manes (2011) found permeability affects parameters used in fitting mean velocity profiles. Unlike mean velocity fitting procedures described by Krogstad (1992), Manes suggests determining the von Karman constant rather than choosing a fixed value.

Hyporheic pressure fluctuations are associated with many flow characteristics as well as physical geometry including coherent flow structures, bedforms, surface waves and obstructions to the flow (Khezri and Chanson 2012). Kline (1967) describes the structure of turbulent

boundary layers in the language of sweeps and ejections. Thomas and Bull (1983) show these coherent structures are cyclic and pressure fluctuations associated with these structures are periodic and adverse pressure gradients can be produced between sweeps and low-speed streaks, also described by Kline (1967). Pressure fluctuations generated in conjunction with bedforms and standing waves are approximately 20 times larger than pressure fluctuations associated with near-bed turbulent coherent motions (Higashino, et al. 2009).

Incipient motion in noncohesive material is closely tied to high-frequency pressure fluctuations caused by turbulent sweeps and ejections (Dey and Papanicolaou 2008, Vollmer, *et al.* 2002). Turbulent bursting events impose a rapid and significant pressure fluctuation on the river bed; these events have a major influence on sediment entrainment (Keshavarzy and Ball 1997). The two most important characteristics of the bursting events in terms of sediment entrainment are sweeps and ejections. Sweeps aid in the detachment of bed material, while ejections keep material suspended (Keshavarzy and Ball 1997). Sweeps and ejections each contain one negative velocity component. In these cases, the instantaneous Reynolds stress contributes positively to the total shear stress shown in equation 2.1. Equation 2.1 shows both the steady and turbulent (Reynolds) components of shear stress. τ is the total shear stress, μ is the dynamic viscosity, u and v are the streamwise and vertical velocities, respectively; y is the height above the bed; ρ is the water density; the embellishments on u and v indicate fluctuation from the mean.

$$\tau = \mu \frac{du}{dy} - \rho u' v' \qquad (2.1)$$

Parametric relations of sediment transport by steady uniform flow yield poor results when applied to more complex flows (Nelson, *et al.* 1995). This is because in steady, uniform flow,

turbulence is characterized by the bed shear stress; however, in nonuniform or unsteady flow, turbulent fluctuations do not necessarily scale with the local bed shear stress (Nelson, *et al.* 1995). Nelson uses flow separation over a dune as an example. Points near the bed, on either side of the flow separation over the dune, can have the same local bed shear stress but significantly different turbulence intensities. A better understanding of flow-sediment interaction does not require further modeling or equations, but rather, careful experimentation to develop more realistic models (Schmeeckle, *et al.* 2007). Schmeeckle (2007) and Smart and Habersack (2007), claim no average characteristic of flow (including bed shear stress) is responsible for the entrainment of sediment. Rather, extreme values of fluctuating forces, such as lift and drag, imparted by the flow directly on the particles are responsible for incipient motion of individual particles (Schmeeckle, *et al.* 2007). Quantification of the spatial distribution of turbulence characteristics is critical for determining the forces acting on a particle (Hofland, *et al.* 2005). Models predicting the motion of sediment rely on assumptions about the origin of the forces; however, little data actually exist to validate these assumptions (Schmeeckle, *et al.* 2007).

Many analytical descriptions of unsteady forces at the water sediment interface exist. However, these descriptions are not especially relevant to the present studies because the fluctuations are limited to the surface and neglect induced pressure fluctuations through the upper sediment layers (Vollmer and Kleinhans 2007). Vollmer (2002) shows pressure differences across a particle are not only described by fluctuating velocities and pressures from above the particle, but also from within the upper sediment layer as well. The turbulent flow field induces randomly distributed pressure gradients over the entire riverbed. The pressure fluctuations penetrate into the bed to a depth over an order of magnitude larger than the bed particle size (Vollmer, *et al.* 2002). Exponential decay functions describe changes in pressure

fluctuations with depth into the bed. These exponential functions are independent of global flow parameters but are highly dependent on the roughness length scale (Vollmer, *et al.* 2002). "Pressure fluctuations induced by turbulence give rise to lift forces acting on grains. These are far more predominant than the lift forces generated by the curvature of streamlines of the flow over the grains" (Zanke 2003). Hydraulic effects of surface-subsurface water exchange are well known, but a lack of knowledge exists describing small-scale unsteady exchange processes (Vollmer, et al. 2002). Knowledge regarding pressure fluctuations within the hyporheic zone aids in the understanding of destabilizing lift forces at work during initiation of sediment motion (Vollmer and Kleinhans 2007). Due to randomly fluctuating velocity and pressure, Hofland (2006) suggests a probabilistic framework to describe lift and drag forces on individual particles.

Detert (2010b) studied the statistics of pressure fluctuations above and within porous beds of uniform and natural sediment as well as spheres of a fixed diameter. All diameters fell within the gravel range, ranging between 7.7 mm and 38.8 mm. The setup in this experiment consisted of miniaturized piezoresistive pressure sensors encased in epoxy with a total diameter of 15 mm. The finished size of the pressure sensor matched the substrate, i.e., it mimicked a particle making up the bed. This is an important breakthrough for measurements of this type and is the controlling factor in determining the limiting particle size for research of this type.

Detert (2010b) positioned pressure sensors from 10 mm above a uniform gravel bed (10mm grain size) to 22 mm below the water-sediment interface. As expected, signal damping increased with depth into the gravel. Detert (2010b) also measured maximum normalized positive pressure fluctuation peaks of $p'/\tau_0 \cong +40$ and extreme negative peaks of $p'/\tau_0 \cong -20$ where p' is the pressure fluctuation defined as the difference between an instantaneous pressure measurement and the average of the time series of pressure measurements

and $\tau_0$is the average shear stress. This indicates a skewed signal. Detert's 2010 work has three main findings: 1) pressure fluctuations scale with the shear stress and equivalent grain roughness, 2) open-channel flow turbulence strongly influences the standard deviation of the pressure signal above and in the roughness layer where the pressure fluctuations exponentially decay with increasing depth of cover, and 3) within the subsurface, the standard deviation of the pressure signal reaches a nonzero constant dominated by long-wave pressure fields convected in the outer flow. Detert (2010a) uses particle imaging velocimetry to describe the vertical and streamwise velocity field in his work. These measurements show streaky structures in the near-bed region. Detert's work however, does not try to decouple the effect of roughness and permeability the way Breugum's works does. Further, Detert's work shows a negative correlation between subsurface pressure fluctuations and the "quasi-mean-shear source term", $u'\bar{u}$, where u' is the fluctuating streamwise velocity and the over bar indicates the mean streamwise velocity. This work also shows $v'\bar{u}$ (v is the vertical velocity component, the embellishments are defined the same as the streamwise components) does not correlate well with bed pressure fluctuations and is of minor importance. Under the conditions in Detert's experiment, bed particles were subject to uplift when sweep events (positive streamwise fluctuation, negative vertical fluctuation) dominate. This implies an upward vertical velocity is not necessary to entrain a particle.

Sediment transport is often characterized by lift and drag forces. The horizontal velocity characterizes drag forces while vertical velocities describe lift forces. Both forces are composed of time averaged and turbulent components. The instantaneous force, either lift or drag is written $F_{L,D} = \overline{F}_{L,D} + F'_{L,D}$ where the over bar is the time averaged force and F' is the turbulent component. The L and D subscripts correspond to lift and drag, respectively. Detert also mounted pressure sensors at the sediment-water interface. Sensors aligned horizontally (with the

pressure pick-up pointed upstream) determined instantaneous drag, while vertically aligned sensors measured instantaneous lift. Detert built histograms of lift and drag for various depths through the gravel. The deepest sensors had no skew and he concluded turbulence due to near-bed velocity fluctuations played no further role in lift and drag. Detert's work also examines pressure spectra. Since turbulence is damped through the porous layer, the resulting spectra curves are shifted toward lower values of spectral power and frequency. These measurements, along with probability distribution functions developed by Hofland (2006), show that for coarse-grained material lift and drag fluctuations are related to either the near bed velocity or the turbulent wall pressure. For distances greater than one roughness height (k_s) above the bed, lift and drag fluctuations are related to the near bed velocity. For distances between the bed and one k_s above the bed, the fluctuations are more related to the turbulent wall pressure.

Hofland (2006) mounted small pressure sensors to the faces of a cube (edge length 3cm) surrounded by gravel of similar size to record instantaneous pressures acting on the top, as well as on the upstream and downstream faces. These pressures were used as a proxy for lift and drag on a single particle and to develop a probability distribution function to describe these forces. Probabilistic approaches are nothing new, but they are usually assumed to be Gaussian. However, it is not known how these distributions change in nonuniform flow (Hofland and Battjes 2006). Hofland derives a PDF of bed shear stress that considers near-bed turbulence intensity accounting for nonuniform flow; therefore, the PDF is applicable near hydraulic structures.

Volmer and Kleinhans (2007) examined turbulence-generated pressure fluctuations through the bed to the bottom of particles and suggest an analytical model of incipient motion. The model is a modified Shields-type equation and was verified with existing data. Critical

forces on individual particles consist of lift and drag forces. Lift forces are decomposed into both steady and turbulent lift components. The turbulent lift is short-lived and originates from vertical pressure gradients in the fluid surrounding the particle (Vollmer and Kleinhans 2007). This turbulence-induced lift becomes more important in the overall force balance with decreasing particle exposure (i.e. decreasing exposed area of the particle). This is due to less surface area for time-averaged drag to act. As a consequence, the turbulent lift force which was dominated by steady drag, contributes a larger portion to the total force acting on a particle.

While many of the elements of the present research were captured in previous experiments, none has combined subsurface pressure measurements in the presence of a hydraulic structure and none have evaluated a substrate composed of grains in the sand fraction. Detert's work captured subsurface pressure fluctuations but in a gravel substrate and no hydraulic structure was present. Hofland derived a probability distribution function to describe pressure fluctuation. The PDF is applicable near hydraulic structures, but measurements were made at the sediment-water interface only and the PDF does not apply to the subsurface. The present research helps complete the picture of the forces and processes at work in the initiation of sediment transport for sand particles around a cylinder.

Chapter 3: Equipment and Methods

The experimental portion of the project comprised nine experimental runs to capture sub-surface pressure fluctuations around a vertical cylinder. Experiments gathered data at three radial locations of the cylinder. The pressure ports on the cylinder were aligned at 0-degrees, 45-degrees and 90-degrees, each relative to the long, centerline of the flume. Each location captured a different element of pressure fluctuation around the cylinder: 0-degree alignment provides upstream fluctuations; 90-degree alignment provides fluctuations transverse to the flow direction; and the 45-degree alignment provides midway fluctuations. Three trials were conducted for each alignment.

Facilities

The University of Windsor Hydraulic Engineering Research Facility provided space and equipment for this research. Three flumes were available for this study along with a variety of velocity measurement techniques. Velocity measurement techniques include particle image velocimetry, laser Doppler velocimetry and acoustic Doppler velocimetry. This research used the longest flume and acoustic Doppler velocimetry. This allowed the greatest length and time for flow development, as well as simultaneous collection of three-dimensional velocity components at a single point.

Flume

This research was conducted in a 91 cm wide, 13-meter long flume. The flume has clear acrylic sides and an engineered plywood bottom. A 60Hz variable speed pump and a 20,000 liter reservoir supply water to the flume. The flume has a flow straightener, elevated sediment test section and a v-notch weir to control the flow. See Figure 3.1 for a schematic and approximate dimensions.

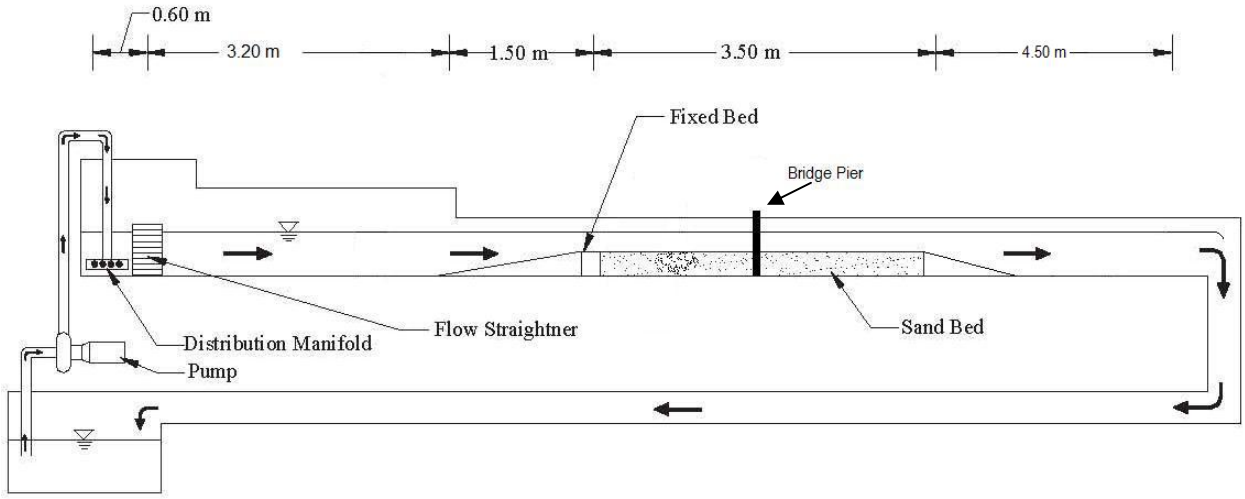


Figure 3.1: Flume schematic modified from Hodi (2009)

The cylinder was placed approximately 7 meters from the head of the flume. At each sampling location around the cylinder, synchronous near-bed velocities and three subsurface pore-water pressure measurements were recorded at three elevations on the cylinder, spaced at 0, 45, and 90-degrees from the streamwise orientation as shown in Figure 3.4. The three sampling elevations were spaced at 3 mm vertical increments over a total depth of 15 mm in the sand bed. Velocity in the flume upstream of the cylinder was maintained at 66-percent of the critical velocity required to entrain the particle. The critical velocity is based on the Hjulstrom curve, Figure 3.2. When the water velocity reaches the line separating erosion from transport, the water is able to entrain material of a given grain size from the bed. The particles can remain entrained until the velocity separating transport and deposition is reached. The Shields diagram, Figure 3.3, was also consulted to determine critical shear stress in the system. Given the parameters in this experiment, the calculated dimensionless boundary shear stress was 93-percent of the critical shear stress.

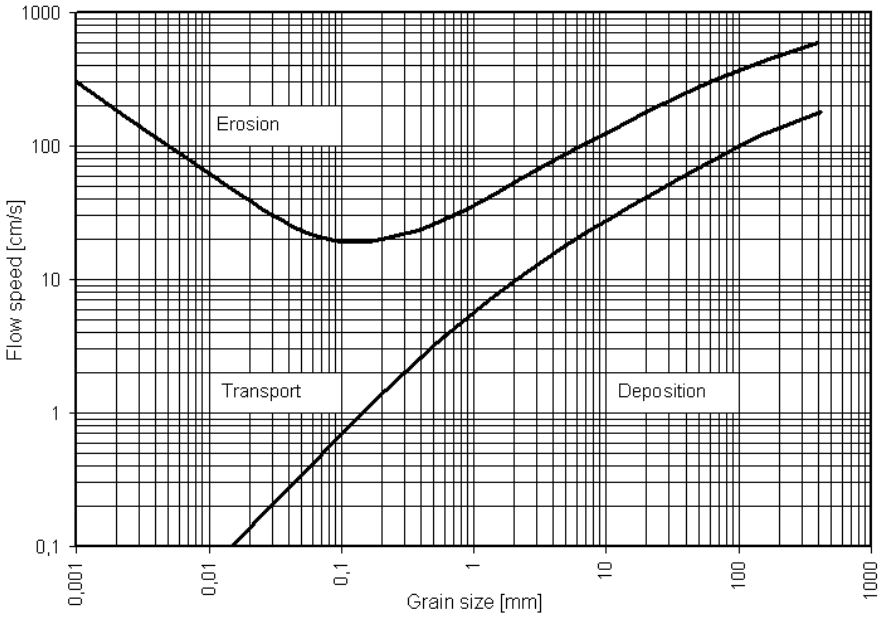


Figure 3.2: The Hjulstrom curve defines critical entrainment and deposition velocities by grain size.

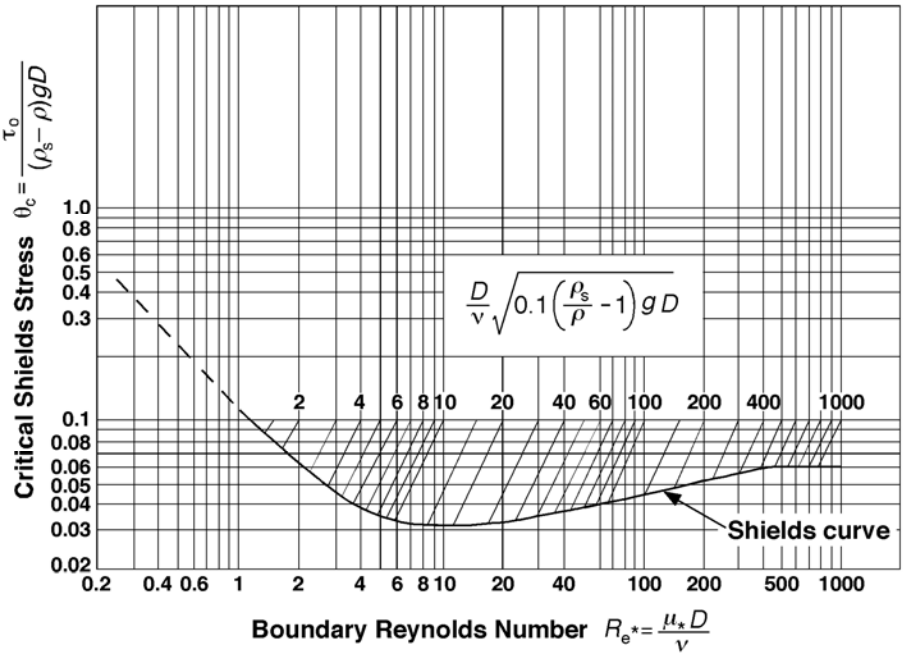


Figure 3.3: The Shield's diagram defines critical entrainment and deposition velocities by grain size.

The low-pressure port for pressure sensor three was fully exposed to the flow (i.e. it was fully above the sand-water interface) at the start of the experiment. Grains moved during the course of the experiment but not enough to expose the high-pressure port on sensor three.

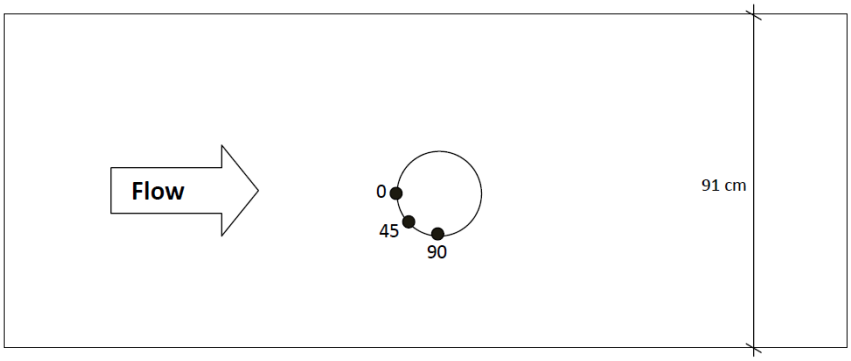


Figure 3.4: Flume plan view schematic showing the locations of pressure measurements relative to the flow. For each location (0, 45 and 90) three pressure sensors were vertically mounted 3 mm on center through the first 15 mm of sand.

V-notch Weir Construction

The v-notch weir is constructed from 18 mm thick, waterproof, engineered plywood. The weir is at the end of the flume 13 meters from the inlet and has the same cross-sectional dimensions as the flume. The notch angle is 68.4 degrees and the distance from the vertex of the notch to the bottom of the flume is 11cm on the upstream side; the downstream side drops more than 130 cm into a return channel, ensuring that submergence is not a problem. The final notch opening was inlayed with aluminum sheet metal 3 mm thick to provide a sharp edge. A bead of silicone around the edge of the weir prevented leakage. Figure 3.5 shows a photo and schematic drawing of the v-notch weir.

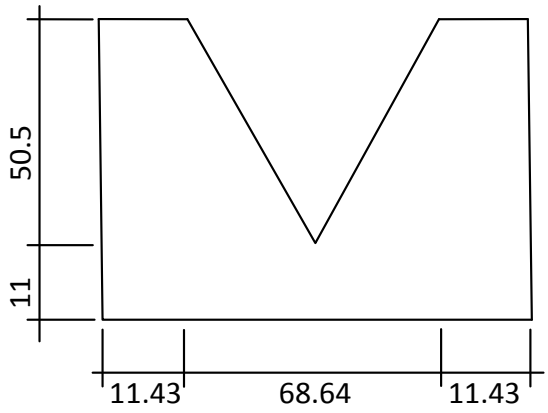


Figure 3.5: Photo of aluminum inlay and schematic drawing of the v-notch weir. All dimensions given in cm.

Weir Calibration

The weir was calibrated through multiple simultaneous measurements of upstream depth and channel discharge. Water depths were measured in the same location every time with the same meter stick approximately 250 cm upstream of the weir. The meter stick was placed in the flow with the width of the meter stick parallel to the streamwise direction, i.e. the most efficient hydrodynamic orientation. The measured upstream depth was used in equation 3.1 to determine total discharge in the flume (Bean 1971). Q is the total discharge (cubic meters per second or cubic feet per second depending on the units of g and h), C is the weir coefficient defined by Bean (1971), g is gravitational acceleration and h is the head on the weir. The weir coefficient was determined from tables in Bean (1971).

$$Q = \frac{8}{15} C * \sqrt{2gh^5} \tan \frac{\theta}{2} \qquad (3.1)$$

Typical weir calibration requires the use of a separate measuring device to determine the weir coefficient. However, the weir coefficient was determined based on tables for v-notch weirs (Bean 1971).

The weir coefficient was also independently determined with an inline flow meter. Flows were measured with an inline Digiflo® electronic flow meter model F2000 by Blue-White industries. The flow meter produces a range of flows for a given pump frequency. The flow meter was observed for a period of two minutes and the high and low flow recorded. The difference between the high and low flow for a single pump frequency ranged from three to 15 percent. No pattern between variance and pump frequency emerged. Calibration curves are provided in Figures 3.6 (a) and 3.6 (b)

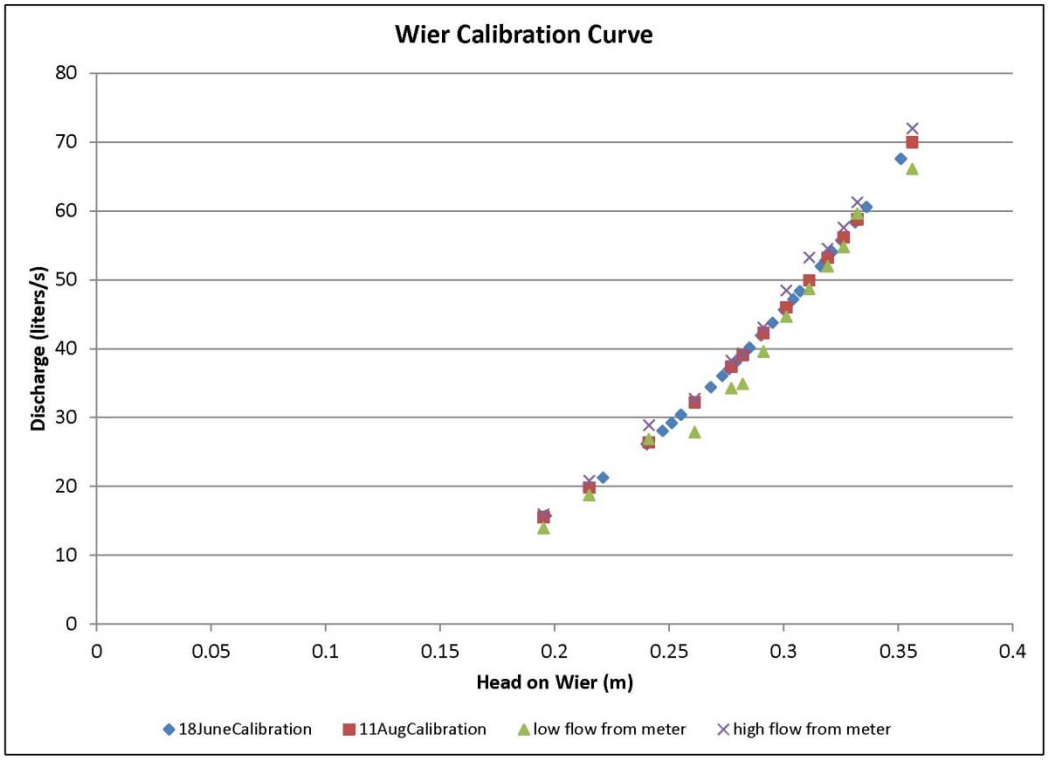


Figure 3.6 (a): V-notch weir calibration curve with 48 cm of water in the return channel

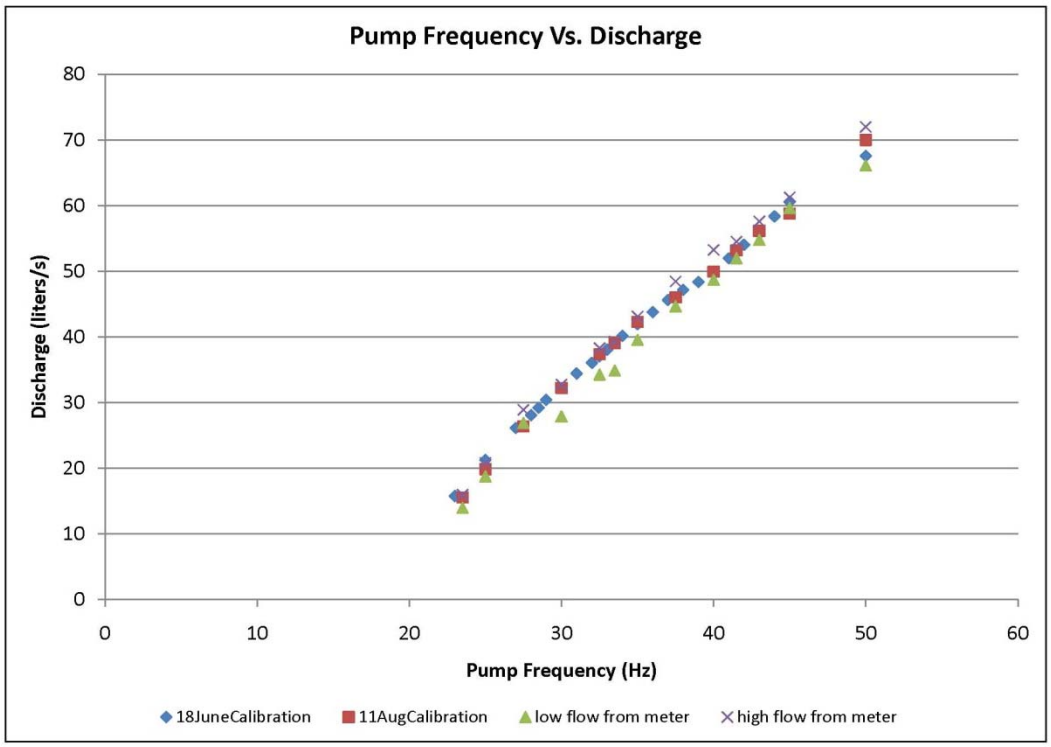


Figure 3.6 (b): pump frequency versus discharge with 48 cm of water in the return channel

Filling the Flume

The weir was covered with a piece of acrylic and the flume was slowly filled with a garden hose. After 10 cm of water covered the sand, the variable speed pump was set to 40 Hz. After the pump primed, (approximately 10 minutes) the water level rapidly increased and the acrylic plate covering the weir was slowly removed. The pump was then set at the operating speed. The operating speed for velocity and pressure measurements was 23.75 Hz and for weir calibration, the operating speed varied from 23.5 Hz to 50 Hz. The head on the weir was measured to check for an equilibrium water level. Once the pump was at the operating speed, at least 15 minutes passed before measuring the head for the first time. Another head measurement was made a minimum of 15 minutes later to check for equilibrium. An additional 30 minutes passed after changing the pump speed before any measurements were made (either velocity/pressure or height on the weir for calibration). Unless a constant water level is

maintained in the reservoir, flow depth in the flume varied as much as 8 mm for the same operating speed. After starting the pump, the water level in the return channel was measured. Water was either added or removed from the system until the depth of water in the return channel measured 48 cm. This ensured the pump added the same head to the system between runs and produced consistent depths in the flume.

Velocimetry

Three velocimetry methods were investigated for this study: particle imaging velocimetry, laser Doppler velocimetry and acoustic Doppler velocimetry. Particle imaging velocimetry uses a laser to illuminate a single plane in the flume. The plane is oriented either vertically or horizontally depending on the velocity components of interest. A camera captures two images with high temporal resolution. The image captures only the particles on the laser-illuminated plane. Successive images are analyzed with image processing software to determine the distance and direction each particle travels. Time between successive images is known and velocities determined. The flume walls are made of Lexan® and are too cloudy and scratched in the test section to use particle imaging velocimetry (PIV).

Laser Doppler velocimetry (LDV) was also considered. Laser Doppler velocimetry uses two (or three) lasers, each with a different color (wavelength/ frequency). The lasers intersect in a small sample volume inside the flume. Light is scattered from particles traveling in the sample volume. The intensity of the reflected light fluctuates and is related to the Doppler shift which is used to determine particle velocity. Near the bed, two component laser Doppler velocimetry provides only the streamwise velocity component. LDV provides point velocities in a small sample volume. The sample volume associated with LDV is significantly smaller than those used in acoustic technologies. Smaller sample volumes allow for measurements closer to the bed and

resolve smaller turbulent features. However, only one-dimensional velocity is possible in the near-bed region. Water clarity is also a concern with LDV. Too many particles in the water creates a diffuse scattering of light. With these limitations, LDV was eliminated as a possible technique for velocity determination in this study.

An acoustic device determined velocities in this study. An acoustic instrument was chosen because there are no practical limits on suspended sediment concentration and it is capable of three-dimensional velocity determination. The first measurement is theoretically possible 5 mm above the bed but, in practice, 7 to 9 mm is a more realistic limit (Nortek help forum, information supplied by Nortek engineers P.J. Rusello and Alte Lorhman, 4 December 2009). Acoustic instruments emit a sound wave with a constant frequency. The sound is reflected off particles in the sample volume and the receivers determine the frequency of the return signal. Some particles are moving toward the receiver and some away from the receivers each with a shifted frequency. The difference in the frequency is used to determine velocity in each of three directions.

Two Nortek® acoustic instruments were used in this research; a side-looking point velocimeter (Vectrino Plus®) and a down-looking profiling velocimeter (Vectrino II®). Both measure velocity in a remote sample volume ~5 cm from the transmitter. This reduces flow disturbances from the probe in the water and is referred to as a blanking distance. The distance to the sample volume varies slightly based on water temperature and other physical characteristics of the system such as bottom type but it is the distance between the sample volume and the bed that is important in this study. The location of the sample volume is determined based on data quality parameters collected with the velocity. Both instruments have a cylindrical sample volume with a fixed 6 mm diameter and a user adjustable height. Heights range from 3 mm to 15

mm for the Vectrino Plus® while the Vectrino II® sample height extends to 30 mm with velocity determined at 1 mm to 4 mm resolution. This project used the full resolution of the instrument and recorded samples every millimeter.

Each instrument has advantages and disadvantages. The most obvious difference (other than the orientation of the probe head) is single point measurement versus a velocity profile. The Vectrino II® measures several sample volumes simultaneously. The entire height of the sample volume in the Vectrino II® is user selectable up to 30 mm with velocities determined every 1 to 4 mm. This significantly decreases the total time required to measure a velocity profile when compared to a single-point velocimeter. The blanking distance on a down-looking probe prevents measurement of the upper 4 cm of flow. This can represent a significant portion of the flow in shallow depths. Side-looking probes can measure a larger portion of the total depth but cannot measure as close to the bed as a down-looker. The Vectrino II® has a significant advantage compared to the Vectrino Plus®; it can capture a time-series of the distance between the transmitter and a solid boundary. A down-looking probe can capture temporal variation in erosional features. The Vectrino Plus ® measures this distance once, prior to a velocity measurement.

The velocimeters were moved into position manually. Once coarsely adjusted, the final position was attained with three, single-axis linear stages and micromanipulators. The micromanipulators had micron resolution but were read to the nearest 0.01 mm. Fine adjustment was used to move the sample volume as close to the bed or cylinder as possible. They were also used for velocity profiling and adjusting the probe up or down a known distance. The linear stages are shown in Figure 3.7 with the side looking probe attached.

Figure 3.7 Three-axis micromanipulator with side-looking velocimeter

Velocity measurement is sensitive to probe alignment. Slight rotation of the stage on the cross flume member resulted in poor data quality indicators, especially for measurements made near the wall. An alignment bracket mounted to the stage resolved the problem. The misaligned stage (exaggerated for the photo) and alignment bracket are shown in Figures 3.8 and 3.9.

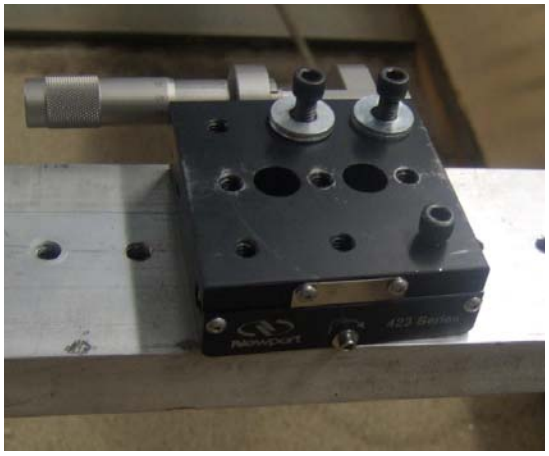


Figure 3.8 misaligned stage (exaggerated for the photo)

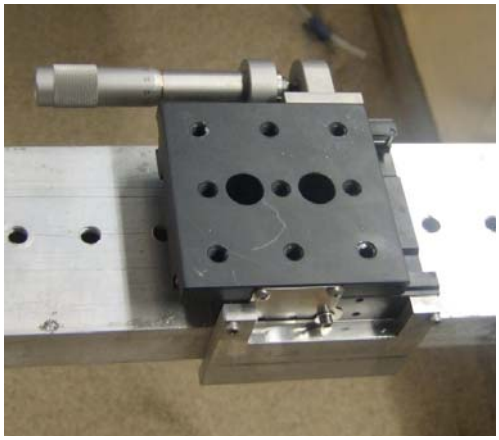


Figure 3.9
alignment bracket

Following alignment of the stages, procedures were taken to ensure alignment of the side-looking probe. The probe was free to rotate within a bracket (Figure 3.10) used to attach the probe head to the vertical stage (not shown here). The probe head was clamped to a level, smoothly machined granite block (Figure 3.11). The probe could rotate within the bracket and was rotated until level. The probe face was checked with a level indicator, Figure 3.12 When the face is level, the needle on the level indicator does not move. The level indicator is accurate to ±0.05 mm.

Figure 3.10 side-looker probe bracket

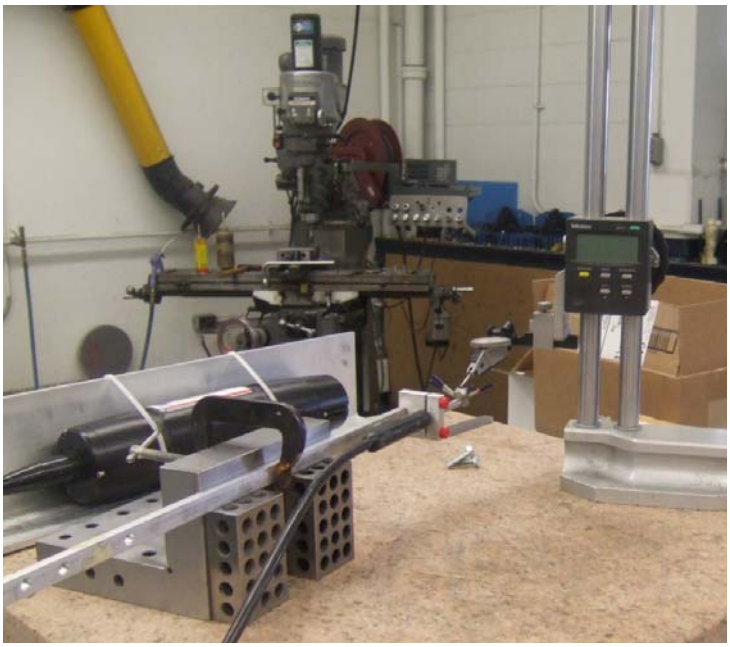


Figure 3.11 probe face alignment, broad view

Prior to the cylinder being placed in the flume, the friction velocity was cortumscently determined using the method described by Krogstad *et al.* (1992). The friction velocity (U_τ, Table 4.1) for the system was determined with a Krogstad (1992) fit. Velocities were collected from $0.2 < y/\delta < 1$ where δ is the thickness of the boundary layer and y is the distance above the bed.

Figure 3.12: probe face alignment. If the needle on the gauge remains stationary as the probe face is moved, the entire face is level.

Seeding the Flow

Both velocimeters required seeding the flow. Seeding increases the amount of material available to scatter the acoustic signal and improves the signal-to-noise ratio. Three materials were considered for seeding: natural river silt and clay, highly processed powdered bentonite and two varieties of hollow glass beads. The natural materials (the silt, clay, and bentonite) have a specific gravity of 2.65 while the hollow glass beads have specific gravities of 0.6 and 1.07.

The natural river clay is from the Grand River upstream of Lansing, Michigan, and the full geotechnical characterization is available in McClerren (2009). It consists of 12-percent gravel, 58-percent sand and 30-percent fines. The bentonite is comprised of uniformly distributed particle sizes and is retained on a 200 mesh. Both of these clays were added to a tub of water, continuously stirred with a propeller mixer and slowly released into the flume through a small-

diameter Tygon® tube. The tube broke the free surface of the water so dripping seeding material did not disturb the surface. A fraction of the natural river clay settled out upstream of the velocimeter creating small bedforms. This altered the local velocity field. Use of this material was discontinued. Figure 3.13 shows the natural river sediment. Bentonite drilling mud was used due to the small, uniform grain size. However, a light sticky film started to coat parts of the flume. Use of the bentonite was discontinued to prevent buildup on internal pump parts. Hollow glass beads, shown in Figure 3.14, were the seeding material of choice in this study. Potters Industries® supplied the glass beads and Table 3.1 shows the physical characteristics. Velocity measurements were attempted with two different density beads. Due to the density, product 60P18 was not a good candidate for velocity measurements. The beads floated to the top and the measured vertical velocity was biased. Particle densities closest to 1 are ideal because they more accurately track the flow.

Table 3.1 Physical properties of SPHERICEL® hollow glass beads

Properties	SPHERICEL® Products	
	110P8	60P18
Density, g/cc	1.10±0.05	0.60±0.05
Bulk density, g/cc	0.49	0.32
Size Distribution µm		
10%	5	9
50%	10	19
90%	21	33
97%	25	36

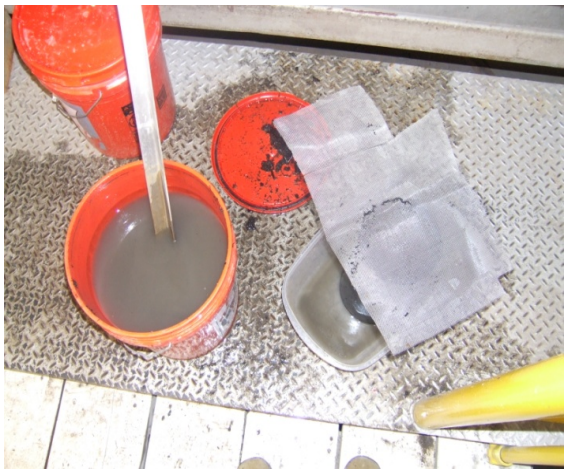


Figure 3.13 Natural river sediment seeding

Figure 3.14 Hollow glass beads

Pressure Sensors

This research used differential, dry-gas pressure sensors from SensorTechnics®. The pressure sensors output a zero-to-five volt analog signal corresponding to a differential pressure head of zero-to-five inches of water. The pressure sensors have an associated accuracy of 0.6% full scale or 0.03 inches of water. For comparison, the most accurate wet pressure sensors found after an exhaustive search were on the order of 1 to 1.5% full scale. The dry gas pressure sensors were mounted in a box above the flume and connected to the cylinder with flexible Tygon® tubing with an inside diameter of 3 mm (6 mm outside diameter). The pressure sensors were connected to the cylinder in two columns: a high-pressure side and a low pressure side. Vertical spacing between the tubes was 6 mm. Figure 3.15 shows the pressure sensors (3.15a), the mounting (3.15b), and the tube placement on the cylinder (3.15c).

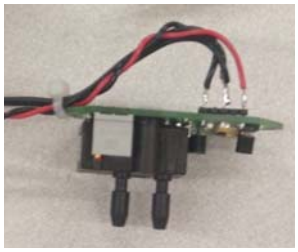


Figure 3.15 a) Pressure sensor

Figure 3.15 b) Pressure sensor assembly

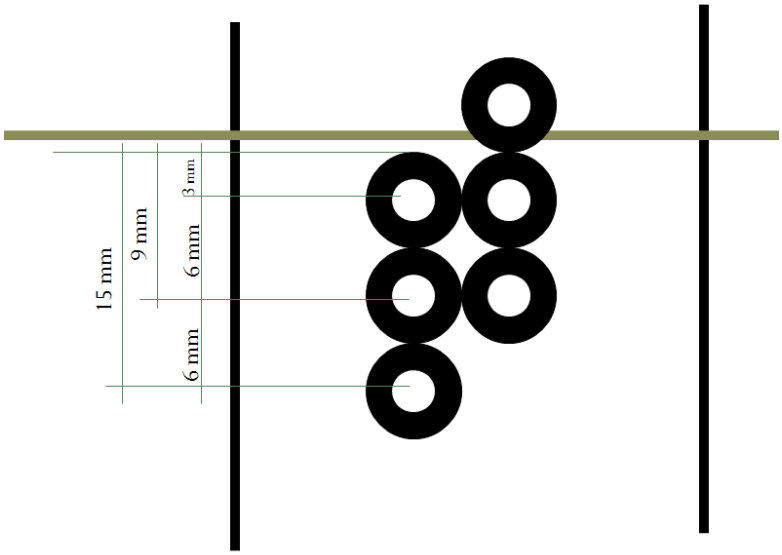


Figure 3.15 c) Differential pressure sensor configuration on the cylinder. Ports on the right are the low-pressure ports, ports on the left are the high-pressure side. The bottom pair of ports represent pressure sensor 1, the top pair of ports are pressure sensor 3.

Smaller connections were attempted using 1.5 mm diameter stainless steel Pitot tubes. The tubes required at least a small length of Tygon® tubing to connect to the pressure sensors. This configuration, shown in Figure 3.16, led to losses in the tubing - as discovered during test runs conducted in a static fluid with known head differential. The pressure differences were not accurately captured until the coupling between the Pitot and Tygon® tubing was removed.

Figure 3.16 Pitot and Tygon® tube connection.

Pressures were sampled at 1000 Hz. They were over-sampled compared to the velocity (100Hz) for maximum versatility. Sampling at 1000 Hz provides high temporal resolution (better for spectral analysis) but also allows for resampling down to 100 Hz to reduce noise and align with velocity measurements.

Bed Condition

The initial sand bed was smoothed such that it was level with the highest, low-pressure tube. Figure 3.17 shows the initial bed condition with the highest, low-pressure port exposed to the water. The bed experienced slight erosion over the course of a five minute measurement cycle. Figure 3.18 shows the distance-to-bed as a function of time (sampled 10 Hz). The first two minutes of the measurement shows a static bed condition. A few millimeters of the bed did erode over the course of the measurement this amounted to about three to four grains of sand. The delay is due to the distance from the cylinder where the velocity is measured. As seen in Figure 3.18, it took about two minutes for any evidence of initiation of erosion to intersect the velocity sampling volume.

Figure 3.17: Initial bed condition and initial pressure port exposure

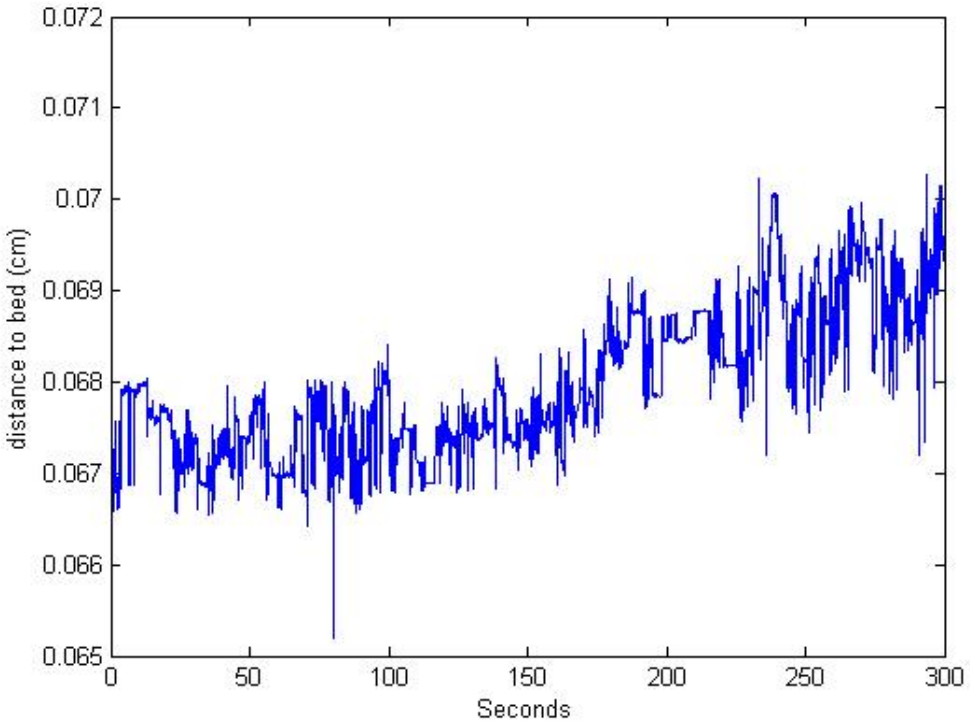


Figure 3.18: Distance to bed as measured from the face of the down-looking velocity profiler

Sand

A uniform, frosted and angular sand was used in this experiment. Hodi (2009) performed an ASTM sieve analysis and describes the sand as follows:

> the mean sediment size (d_{50}) is 0.85mm and corresponds with the manufacturer specifications. The uniformity coefficient C_u (d_{60}/d_{10}) is 1.4. A uniformity coefficient less than 4 in the sand fraction is classified as uniform. The gradation coefficient C_c defined as $d_{30}^2/d_{60}d_{10}$ is 1.03. $C_c \sim 1$ indicates the sediment is well graded. The sieve analysis curve is presented in Figure 3.19

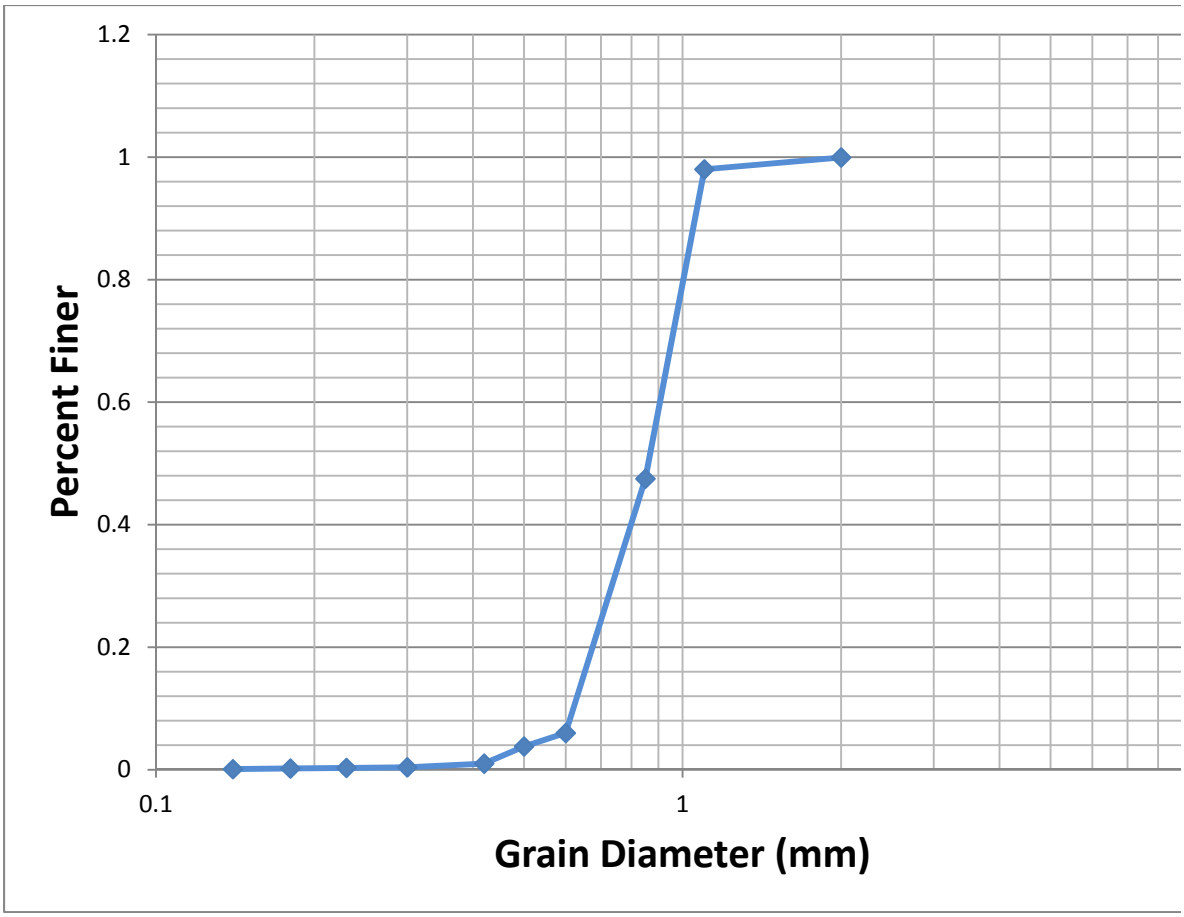


Figure 3.19: Grain size distribution

Analog to Digital Converter

Data acquisition was performed with a LabJack® U3-High Voltage device. This device has 12-bit resolution, and can record 50,000 samples per second on a single channel. The timer has an accuracy of 1.5%. This research used three pressure sensors; maximum throughput for the converter is approximately 16,500 samples per second per pressure sensor. The LabJack® was used to synchronize the velocimeter and the pressure sensors through transistor-transistor logic (TTL). TTL is a standardized protocol included with LabJack® and the velocimeter. The LabJack® was programmed to start sampling the pressure sensors and simultaneously send a small voltage spike to the velocimeter. The velocimeter was configured to start sampling upon receiving a voltage spike of 3.5 volts.

Analog to digital converters are used to digitize either voltage or amperage output from peripheral devices such as pressure sensors. The pressure sensors operate on a piezoresistive principal; that is, a constant voltage is supplied to the sensors and the returned voltage is proportional to the differential pressure head and recorded on a laptop.

Chapter 4: Results

Pressures and velocities were measured with sufficient spatial and temporal resolution to describe subsurface pressure decay at three radial positions around a cylinder. Pressures and velocities were measured and analyzed to determine probability distributions; distributions were developed for each location and depth around the cylinder.

Three trials were conducted for each pressure port location (0, 45 and 90 degrees). The same flow conditions were used for all nine experiments. Velocity and pressure measurements were recorded at 100 Hertz for five minute trials. Spectral densities of each signal were examined with the Welch method using 50-percent overlap. A peak occurred in both series (velocity and pressure) at approximately 0.4 Hertz as seen in Figure 4.1, Properties of the flume, sand and velocity for all runs are shown in Table 4.1.

Table 4.1: Flow and flume characteristics

Pier diameter (m)	Flume width (m)	Depth (m)	U (m/s)	U/U_c	U_τ (m/s)	D_{50} (mm)	Reynolds Number	Particle Reynolds Number
0.11	0.90	0.08	0.20	0.66	0.0198	0.85	22,000	15

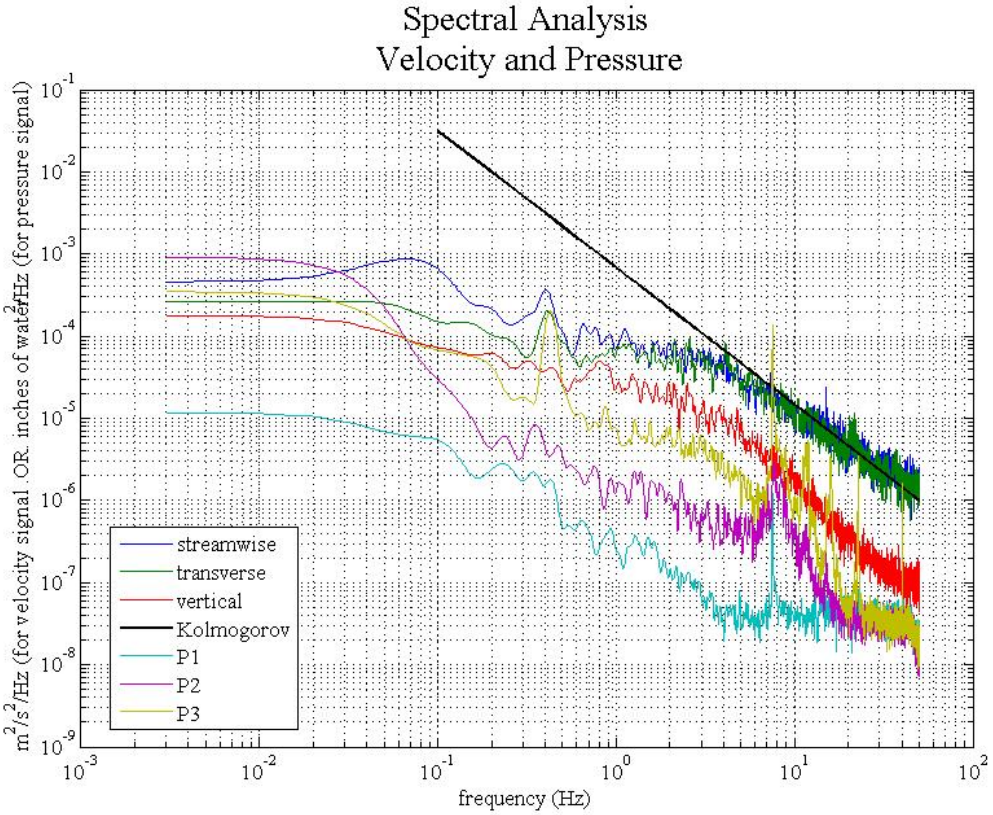


Figure 4.1: Spectral analysis shows the connectedness of the subsurface pressure and velocity data. Both velocity and pressure spectra have spikes at about 0.4 Hz which corresponds to the theoretical value of 0.38 Hz

Figures 4.2 and 4.3 provide a general overview of the synoptic velocity and subsurface pressure signals. Figure 4.2 shows a long time sequence to compare longer-term trends, while 4.3 focuses on the smaller time scale.

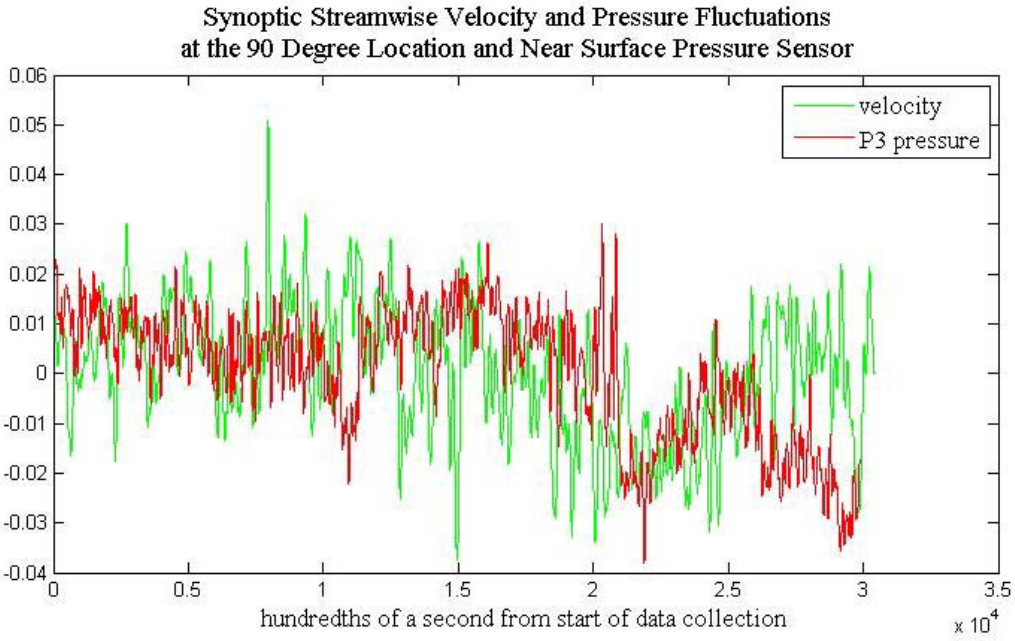


Figure4.2: Long-term time-window to evaluate pressure and velocity correlation

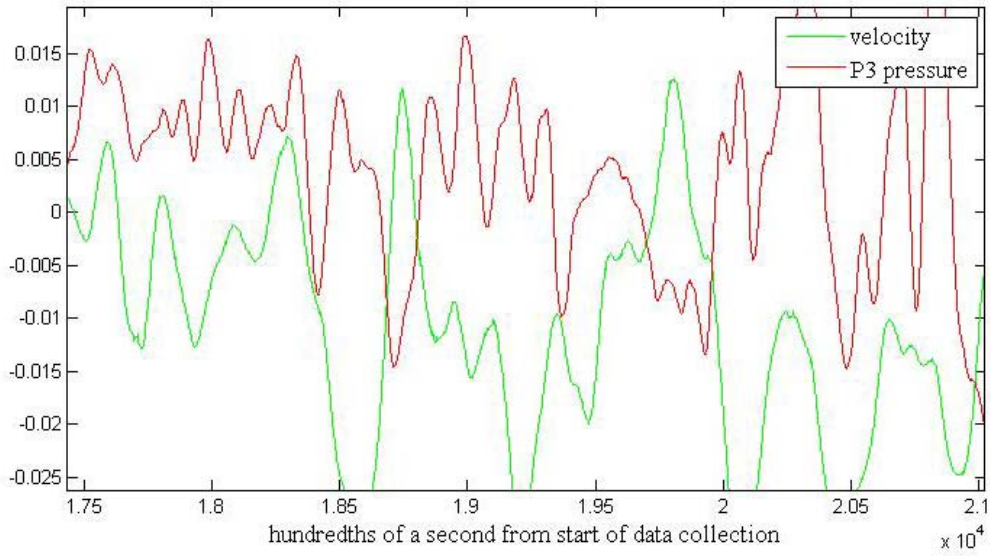


Figure4.3: short time window to evaluate trend in pressure and velocity correlation

Looking beyond pressure sensor three, the subsurface pressure distributions change with radial position and depth. At all radial locations, the intensity of the pressure

fluctuations decreases with depth as seen in Figure 4.4. Figure 4.4 shows the standard deviation of the pressure fluctuations as a function of depth of cover. Both parameters were non-dimensionalized with shear stress and median grain size, respectively. Each of the points on the graph represents the standard deviation of approximately 30,000 measurements.

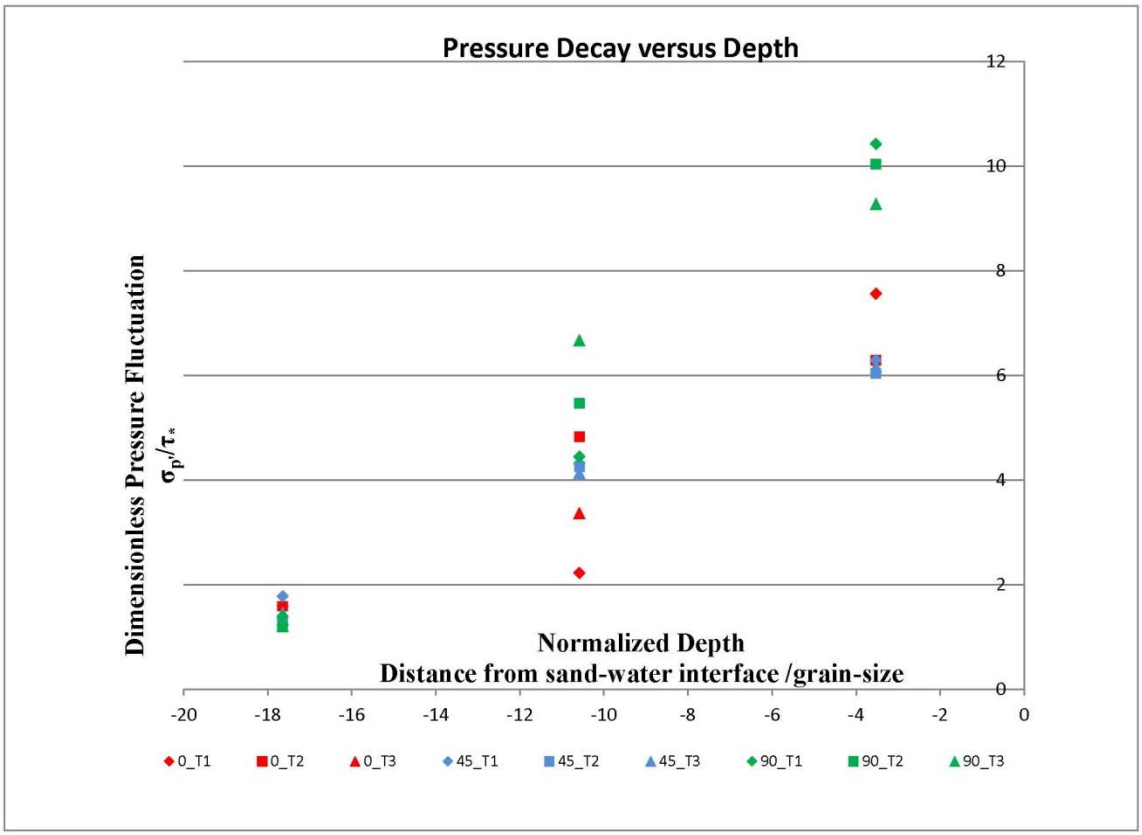


Figure 4.4: Non-dimensional pressure fluctuations versus non-dimensional depth for all three trials at the zero, 45- and 90-degree locations

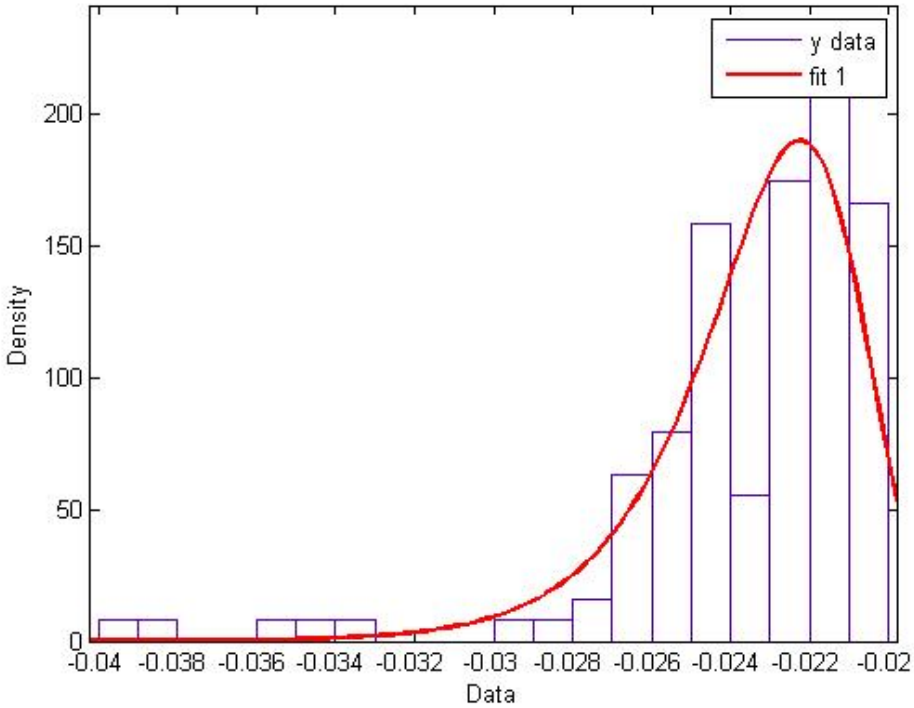
In general, the pressure fluctuations recorded at the deepest probes are normally distributed, but the shape of the distribution changes with depth and radial position as discussed below. Pressure distributions are analyzed with a combination of normal probability plots and histograms. When data plots on a straight line in a normal

probability plot they are considered normally distributed. As the deviation from the straight line becomes significant, the distribution of data is no longer considered normal.

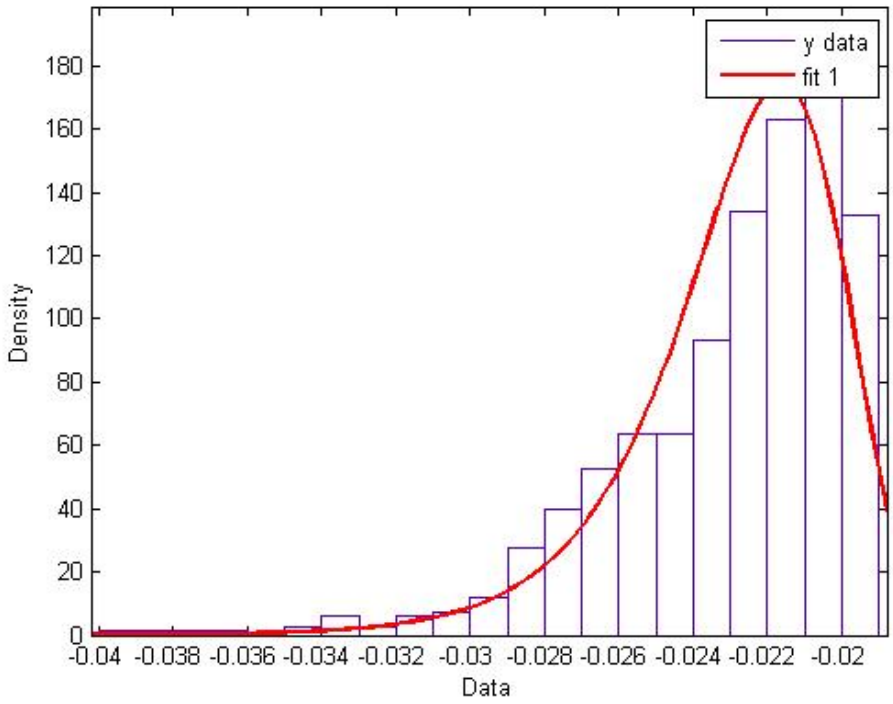
Several authors including Blake (1970), Hinze (1975) and Detert (2004) determined that the root mean square of the pressure fluctuations at the bed is approximately three times the boundary shear stress for the smooth or rough flat bed case. Pressure sensor three is 3mm beneath the surface of the sand/water interface and is useful for comparisons between near bed pressures and shear stress similar to those by Vollmer and Kleinhans (2007). Based on data from this study for the 0 and 45-degree alignments, the root mean square of the pressure fluctuations is approximately seven times larger than the boundary shear stress.$P_{rms} \approx 6\tau_*$. For the 90-degree location $P_{rms} \approx 10\tau_*$.

When the instantaneous pressure fluctuation above a particle is reduced relative to the average differential pressure at the same point, the particle requires a reduced force to initiate motion. This distribution of reduced pressure fluctuations is well described with a Gumbel distribution. The Gumbel distribution is an extreme value distribution and is frequently used to model natural phenomena. The largest near surface events (pressure fluctuations greater than 1.5 standard deviations) in every trial at every position is modeled well with a Gumbel distribution. Figure 4.5(a)-(c) shows examples from the 0-degree position (a) to the 90-degree position(c). Table 4.3 presents the p-values from a Kolmogorov-Smirnov test, showing the Gumbel distribution is a good fit for all pressure fluctuation measurements in all the locations (except in Trial 1 at the 90-degree location).

Table 4.2: P-values for the KS-test for Type I Gumbel distribution fits			
P values for 0-degree location for a Gumbel fit of large pressure fluctuation			
	P1	P2	P3
Trial 1	0.0039	0.0055	0.0049
Trial 2	0.0174	2.35E-02	8.34E-04
Trial 3	0.0042	0.0017	0.0125
P values for 45-degree location for a Gumbel fit of large pressure fluctuation			
	P1	P2	P3
Trial 1	0.0016	0.0723	0.0424
Trial 2	0.0024	1.90E-05	2.58E-04
Trial 3	0.00045	0.004	0.0033
P values for 90-degree location for a Gumbel fit of large pressure fluctuation			
	P1	P2	P3
Trial 1	0.0228	0.7695	0.3599
Trial 2	0.004	1.06E-04	2.74E-01
Trial 3	0.0178	7.80E-04	7.90E-04



(a) Pressure sensor three: Typical Gumbel Distribution fit at the zero degree position



(b) Pressure sensor three: Typical Gumbel Distribution fit at the 45-degree position

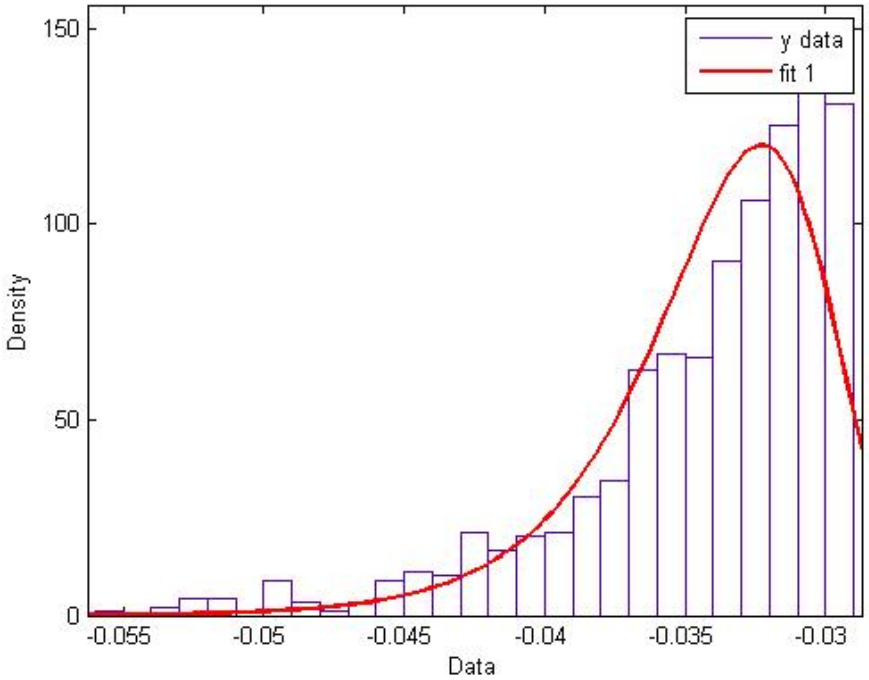


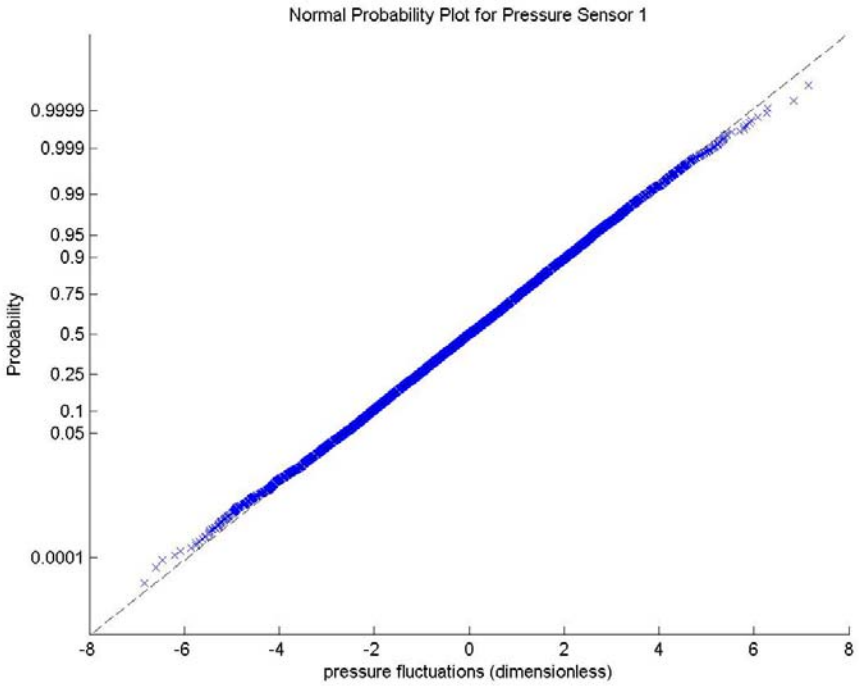
Figure 4.5 (c): Pressure sensor three: Typical Gumbel distribution fit at the 90-degree position

Zero Degree Location

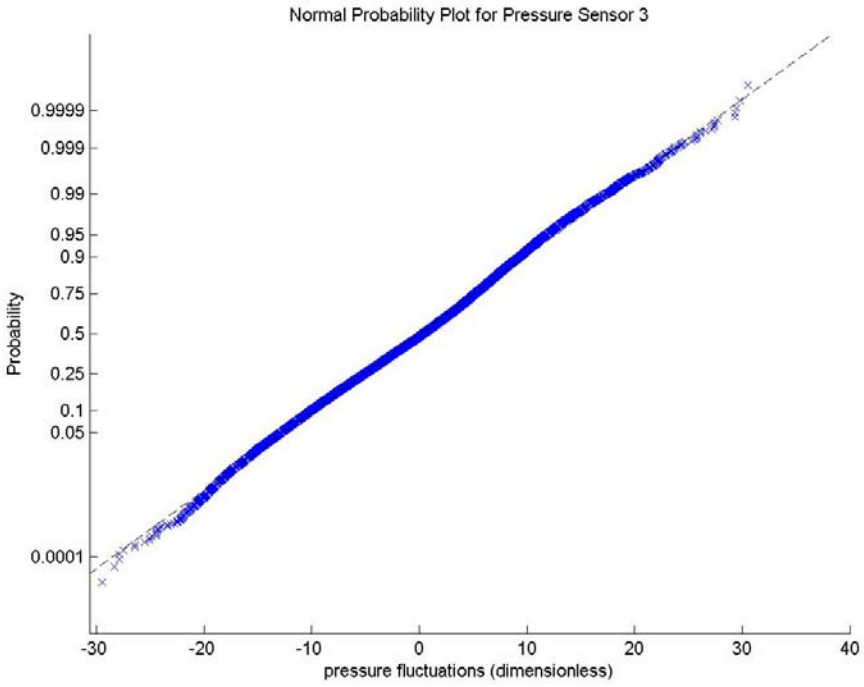
Pressure sensors located at a 0-degree alignment had an approximate normal distribution at all depths as shown in Figures 4.6a and 4.6b, as well as the histograms in Figures 4.7a and 4.7b. Pressure sensor one is 15 mm below the sand surface. The measured pressures at this sensor most closely match a normal distribution. Pressure sensors two and three show small deviations from normal in the tails. This can also be seen in Table 4.4 with the skewness values of both the pressure data as well as each velocity component.

Table 4.3: skewness values for pressures and velocity components in the 0-degree location

Skew	Trial1	Trial2	Trial3
P1	-0.0092	0.2122	0.0848
P2	-0.1433	0.1257	-0.0631
P3	-0.0964	0.0384	-0.0106
streamwise	0.009	0.0399	-0.0094
transverse	0.0145	0.0096	-0.0113
vertical	0.095	0.144	0.1426

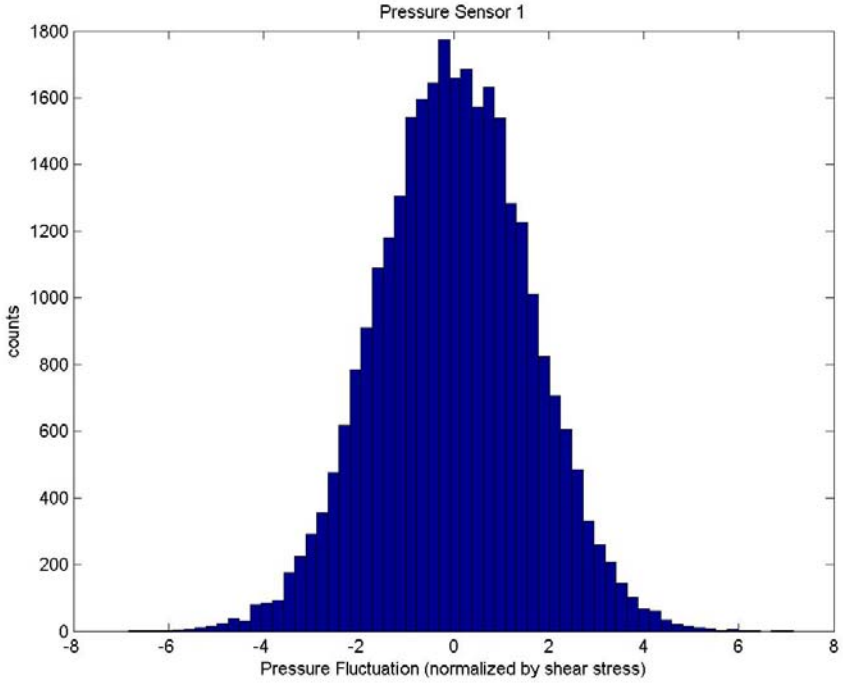


(a)

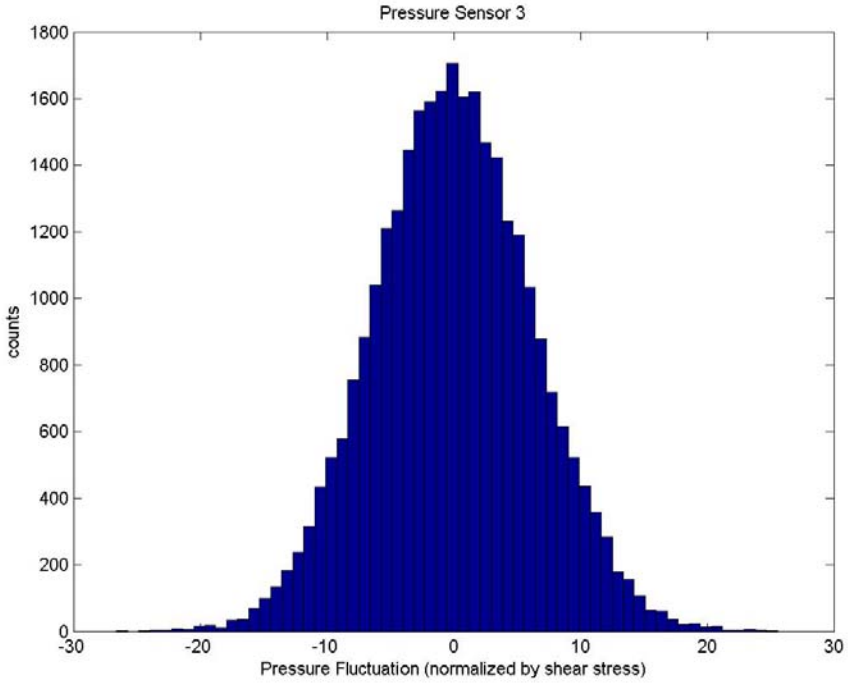


(b)

Figure 4.6: Normal probability plots for subsurface pressures (a) 15 mm, and (b) 3 mm deep. Pressure sensor two has a similar distribution to sensor three.



(a)



(b)

Figure 4.7:. Pressure histograms for subsurface pressures (a) 3 mm, and (b) 15 mm deep. Pressure sensor two has a similar distribution to sensor three.

Raw data displayed in the box plot of Figure 4.8 shows the spread in pressure fluctuations. Pressure sensor one (the deepest sensor) has the smallest inter-quartile range (IQR) while pressure sensor three (closest to the sand-water interface) has the largest IQR.

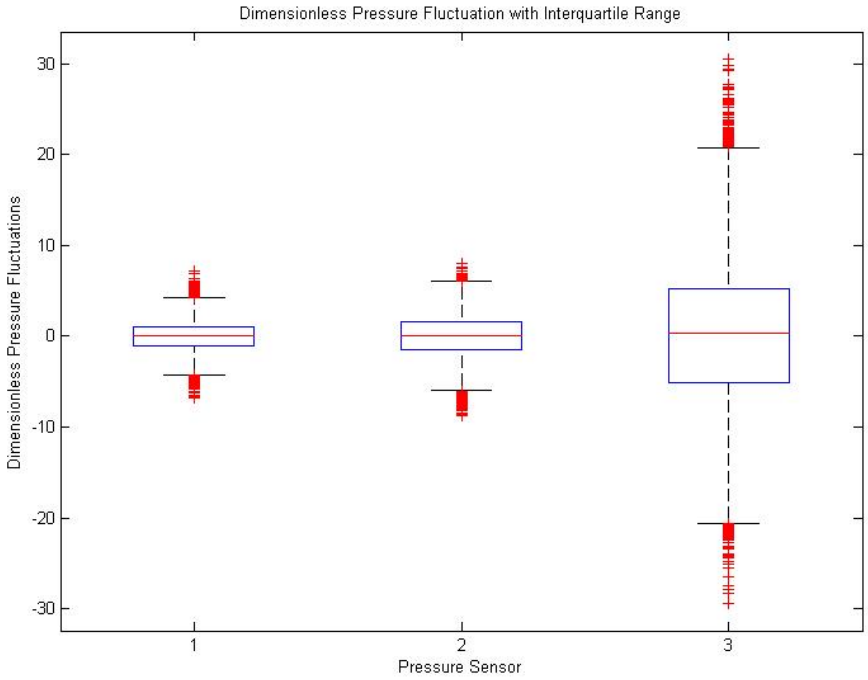


Figure 4.8: Boxplot showing inter-quartile range of all three pressure sensors from the 0-degree location. Each whisker is 1.5IQR long.

Measurements to the bed were made while simultaneously measuring velocity. Distances are referenced to the face of the velocity profiler. For the 0-degree case, point measurements do not indicate a discernible trend, figure 4.9.

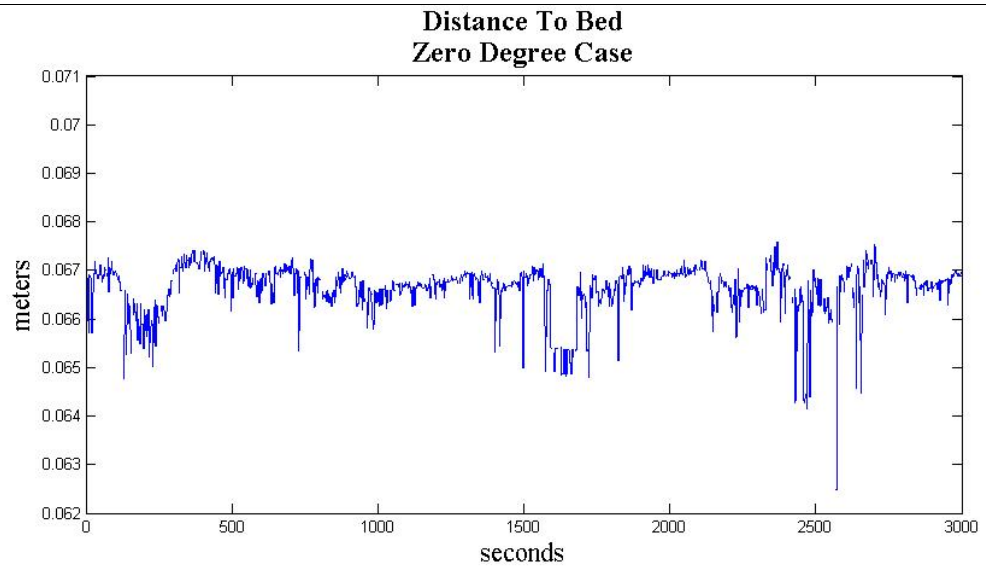
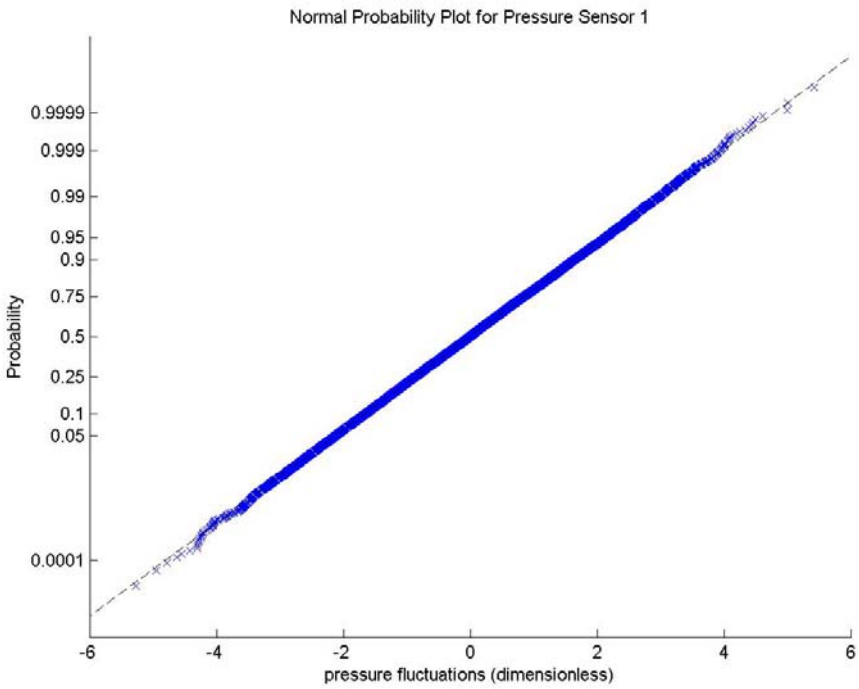


Figure 4.9: Typical distance to bed measurements for the 0-degree case

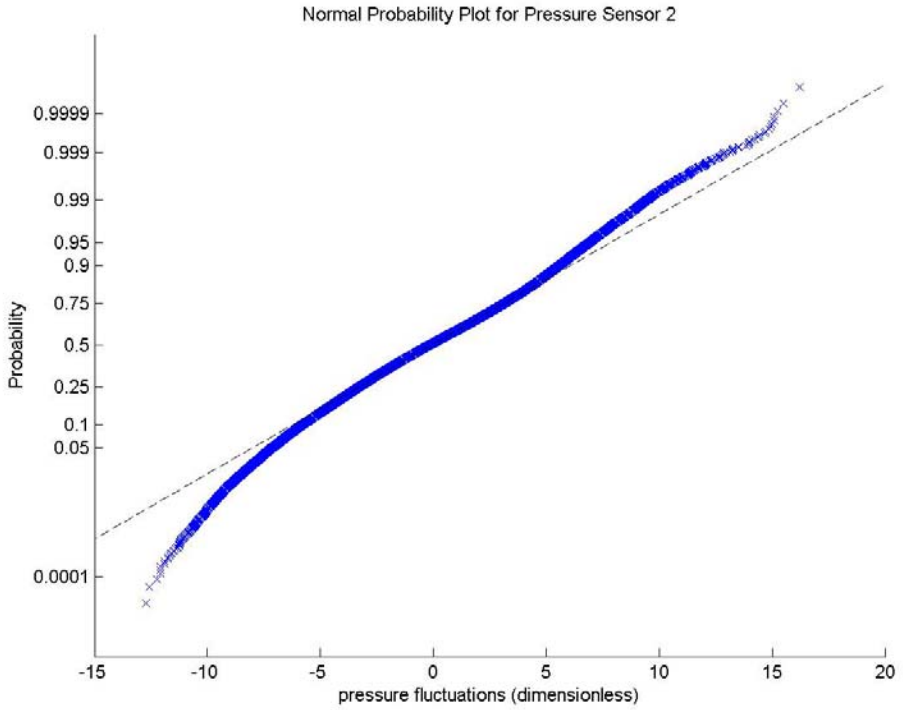
45-degree Location

At this location, pressure sensor one, approximately 15 mm below the surface, has an approximate normal distribution for all trials; a sample is shown in Figure 4.10a. However, some anomalies exist at the 45-degree location for pressure sensor two and pressure sensor three. In trials two and three, pressure sensors two and three show a deviation from normal in the tails and have negative skew (see Figures 4.10b and 4.10c, as well as Table 4.5). Trial 1 is similarly skewed in pressure sensor two and pressure sensor three, however, in the other (positive) direction. The vertical velocity in trial one also has a positive skew; trials two and three have a negatively skewed vertical velocity component.

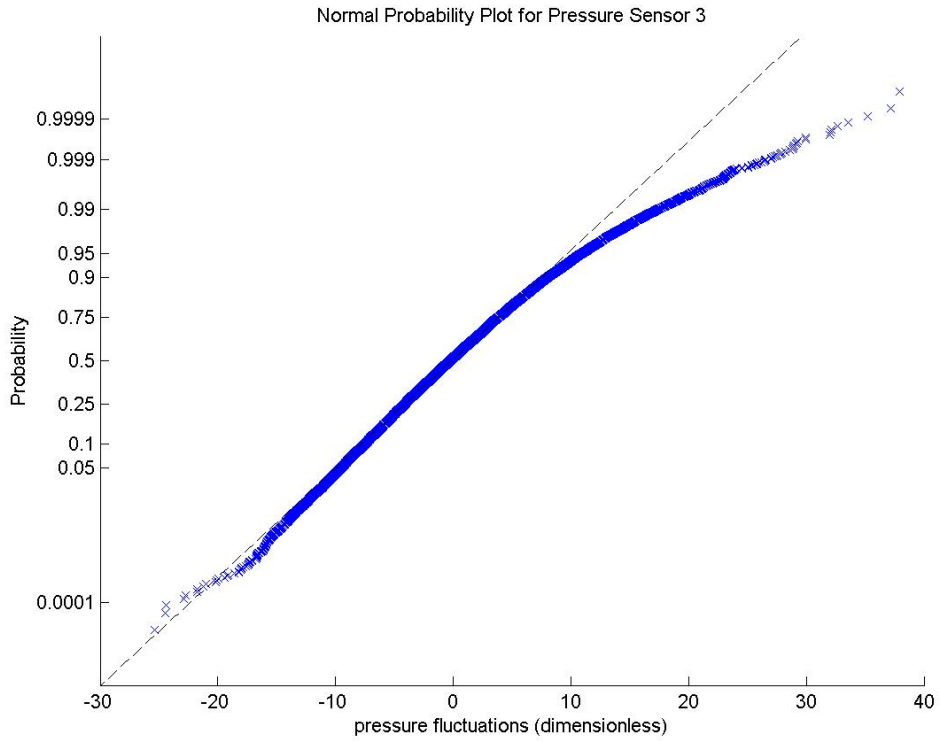
Table 4.4: skewness values for pressures and velocity components in the 45-degree location			
Skew	Trial1	Trial2	Trial3
P1	0.0179	0.0058	-0.0378
P2	0.0603	-0.7787	-0.3627
P3	0.4548	-0.3516	-0.1305
streamwise	-0.3897	0.6197	0.2134
transverse	-0.0973	-0.5608	-0.1592
vertical	0.7083	-0.4427	-0.2318



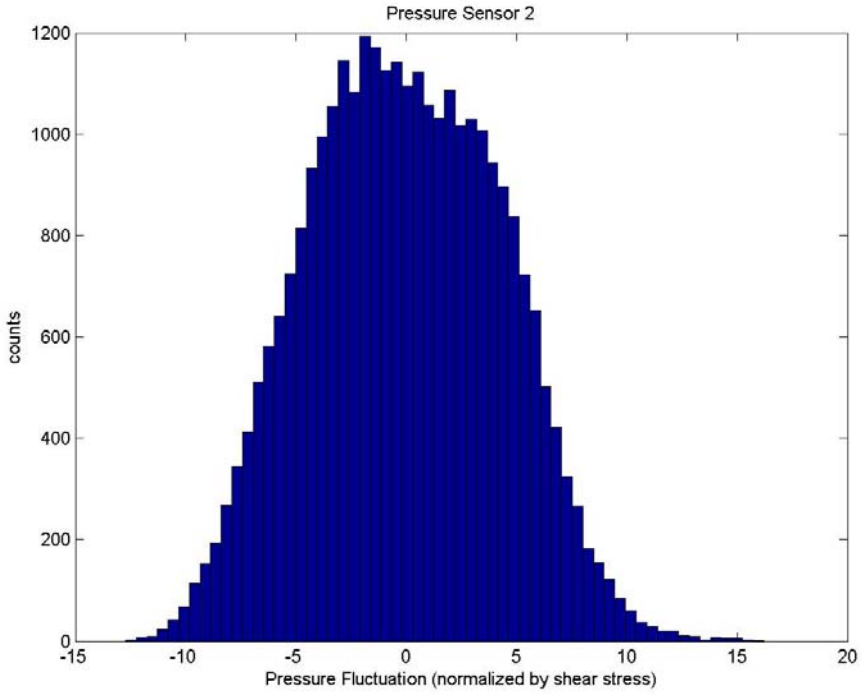
(a)



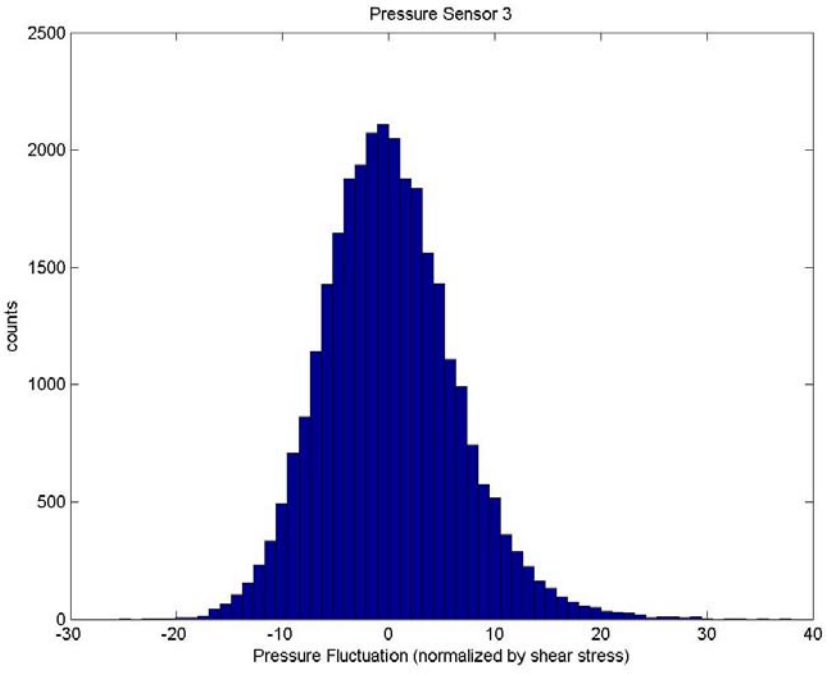
(b)



(c)



(d)



(e)

Figure 4.10: Normal probability plots show a near normal distribution of the deepest pressure sensor (a) and a slight negative skew for pressure sensor two and three in (b) through (e)

The box plots in Figure 4.11 show the spread in the pressures from all three depths. These plots, like the ones from the 0-degree location all show an inverse relationship between depth and inter-quartile range.

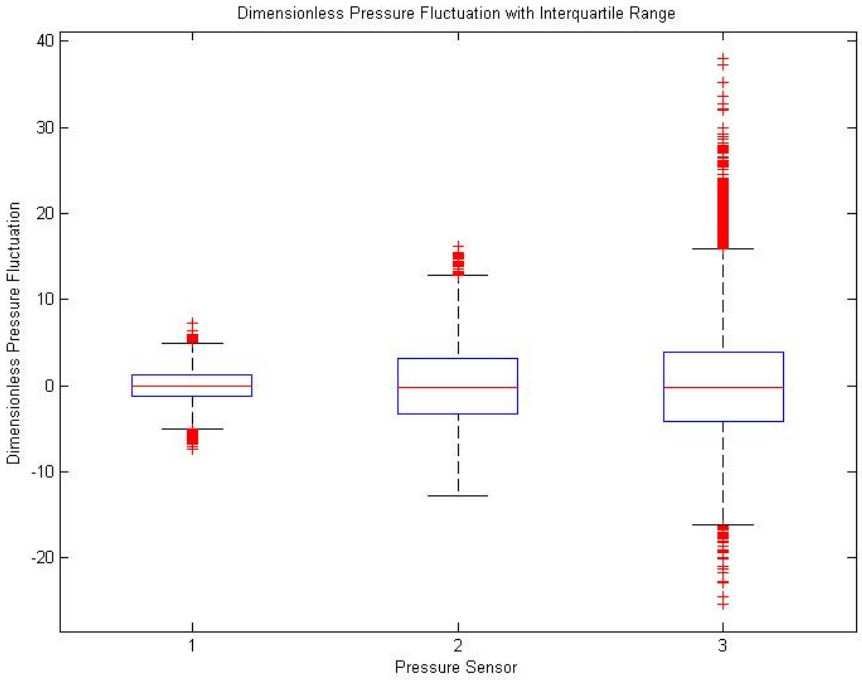


Figure 4.11: Box plots showing the inter-quartile range of the pressure sensors for the 45-degree location

Distance to bed measurements were made while simultaneously measuring velocity. Distances are referenced to the face of the velocity profiler. For the 45-degree case, point measurements indicate slight erosion. Over the course of a five-minute, measurement the bed elevation lost three millimeters or slightly more than three, median grain diameters, Figure 4.12.

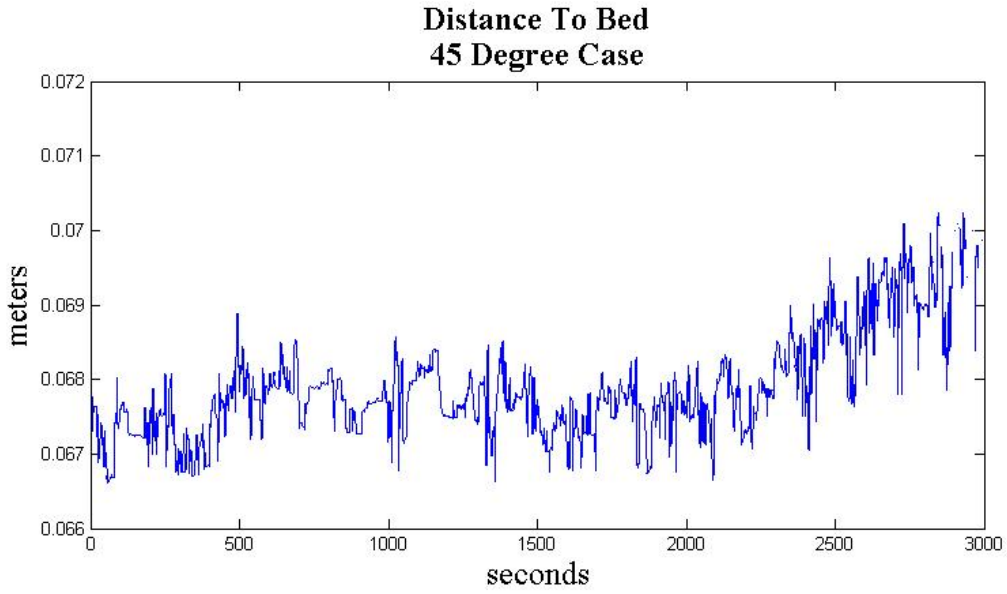


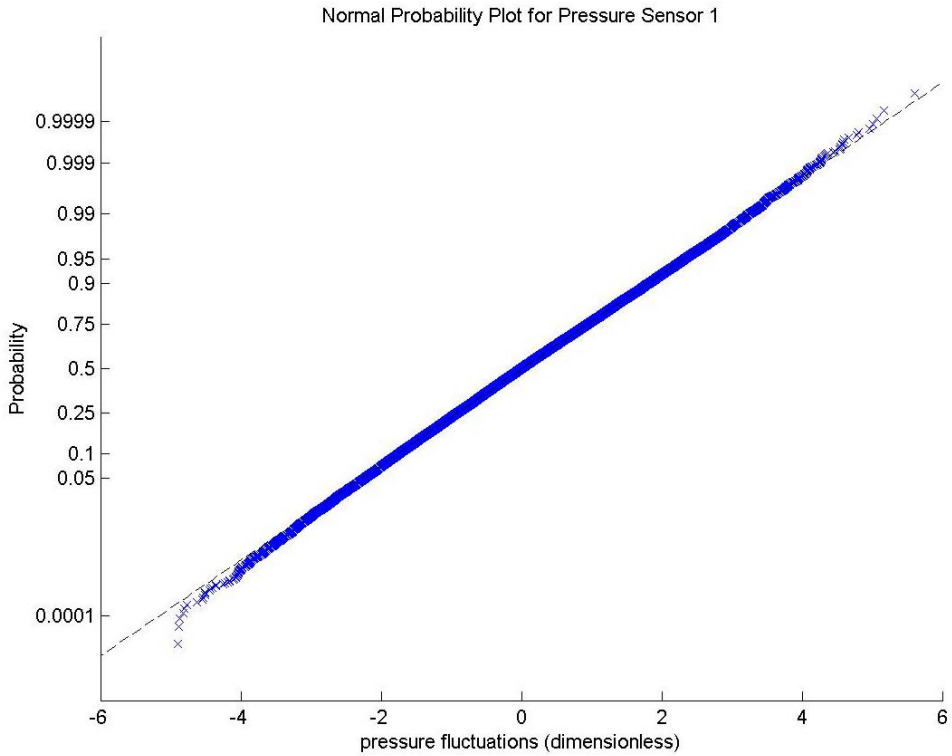
Figure 4.12: Typical distance to bed measurements for the 45-degree case

90-degree Location

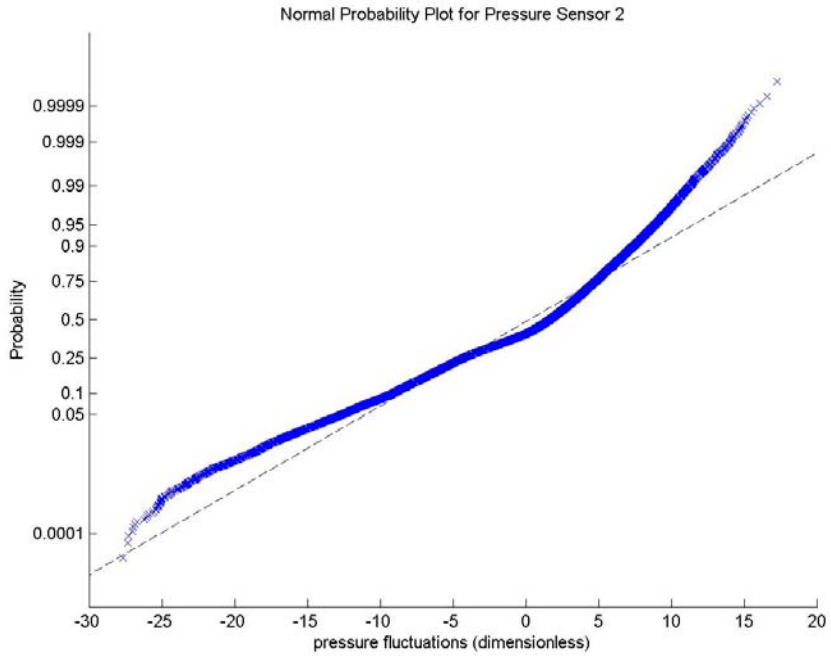
In the 90-degree position, the pressures recorded at the deepest pressure sensor (P1, 15 mm deep) are still approximately normally distributed, Figure 4.13 (a). However the pressures recorded at pressure sensors two and three exhibit bimodal behavior. This behavior is explained by periodic vortex shedding associated with flow around a cylinder and can be seen in normal probability plots and histograms shown in Figure 4.13 (b) –(e). Table 4.6 has statistics similarly calculated for the 0 and 45 degree positions.

Table 4.5: skewness values for pressures and velocity components in the 90-degree location

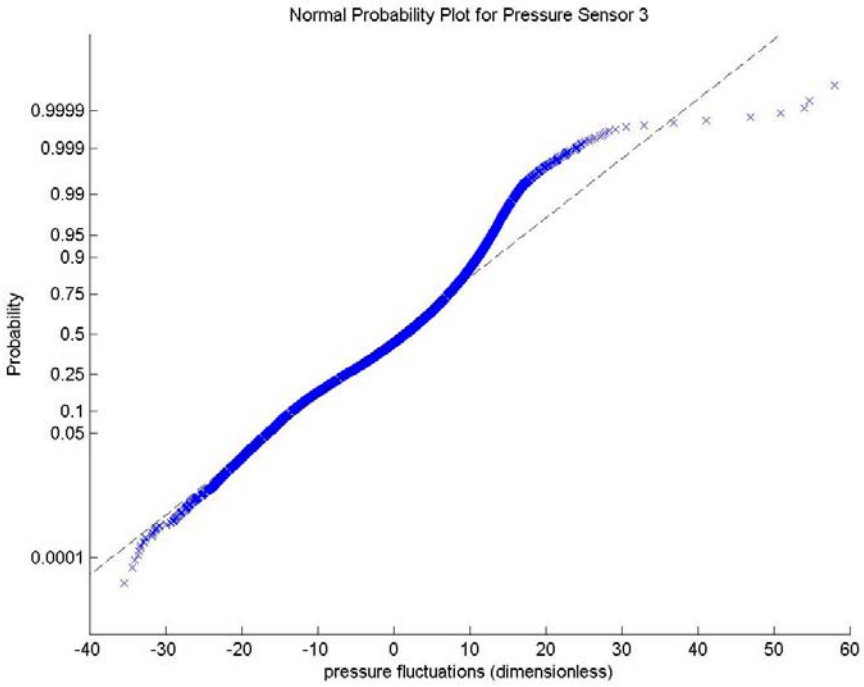
Skew	Trial 1	Trial 2	Trial 3
P1	0.0022	0.0019	0.0019
P2	0.007	0.0086	0.0105
P3	0.0164	0.0158	0.0146
streamwise	0.1589	0.1408	0.194
transverse	0.1123	0.267	0.6699
Vertical	0.413	0.5446	0.2428



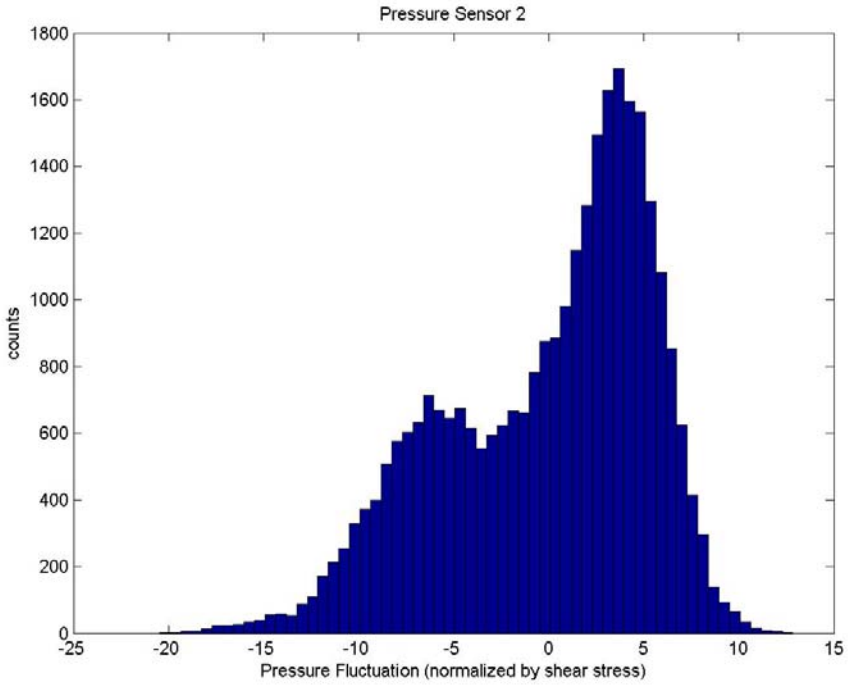
(a) Normal probability plot for subsurface pressure fluctuations at the 90-degree location



(b) Normal probability plot for subsurface pressure fluctuations at the 90-degree location



(c) Normal probability plot for subsurface pressure fluctuations at the 90-degree location



(d) Histograms for subsurface pressure fluctuations at the 90-degree location

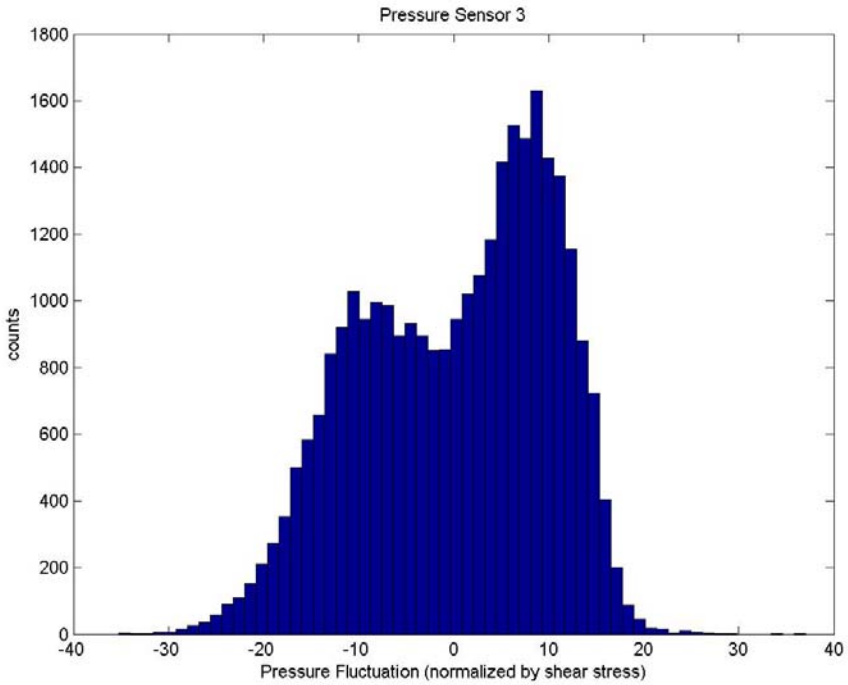


Figure 4.13(e): Histograms for **s**ubsurface pressure fluctuations at the 90-degree location

Figure 4.14 shows the spread of the pressure fluctuations for the 90-degree alignment. In a similar fashion to observations for the other two alignments, pressure sensor one has the smallest inter-quartile range and sensor three the largest. All three radial locations have an approximate range of 60.

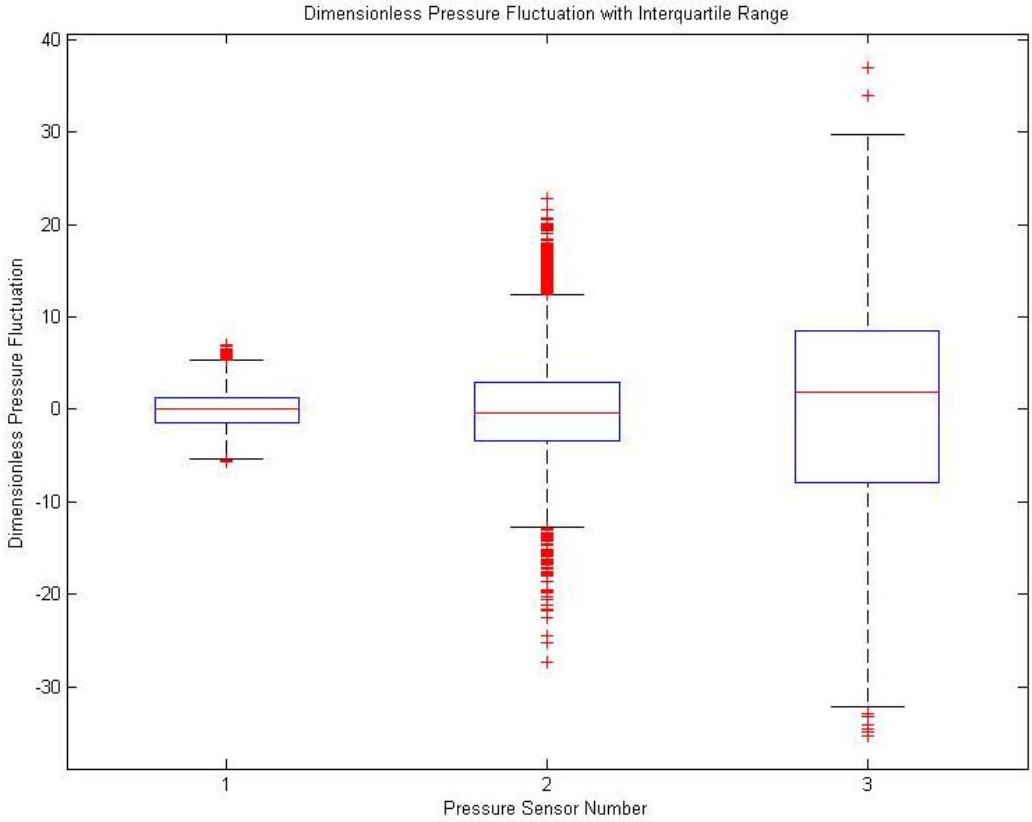


Figure 4.14: Boxplot showing inter-quartile range of all three pressure sensors at the 90-degree location. Each whisker is 1.5IQR long.

Distance to bed measurements were made while simultaneously measuring velocity. Distances are referenced to the face of the velocity profiler. For the 90-degree case point measurements indicate slight aggradation. Over the course of a five-minute, measurement the bed elevation gained three millimeters or slightly more than three, median grain diameters, Figure 4.15.

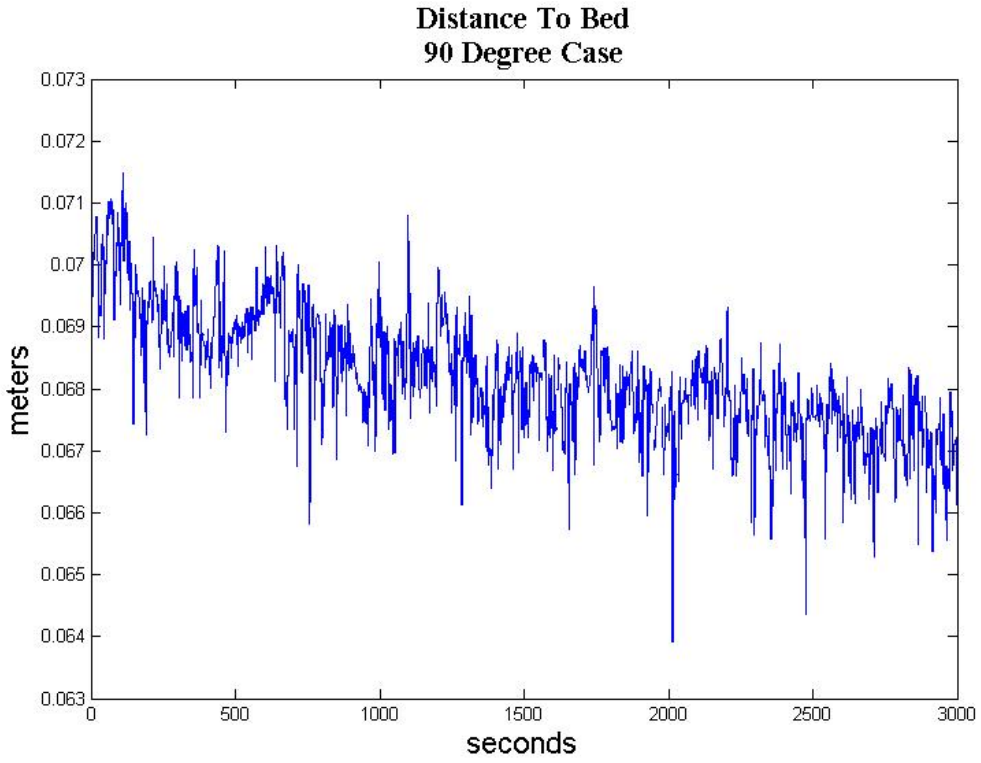


Figure 4.15: Typical distance to bed measurements for the 90-degree case

Chapter 5: Discussion

Discussion of results

The results will be described in relation to the hypotheses of this project, which states: 1.) *the characteristics of the subsurface pressure fluctuations near a vertical cylindrical obstruction to open channel flow vary with the radial position along the obstruction.2.) the pressure fluctuations exponentially decay with depth for all three radial positions,* and the *distribution* of pressure fluctuations change as a function of both depth and radial position. *3.) Pressure is most correlated with vertical velocity fluctuations at the zero degree, and the streamwise or transverse velocity components at the 45 and 90 degree positions*. Correlations are expected to he highest with the vertical velocity at the zero degree position because of the strong downward velocity at the front of the cylinder. As the streamlines bend around the cylinder, Figure 1.3, the streamwise or transverse velocity components are expected to play a more important role in hyporheic pressure fluctuations. Experiments were designed and executed to test these hypotheses. Results from the experiments confirm the radial dependence of the pressure fluctuations .

Pressure sensor one is buried deep enough in the sediment that all pressure fluctuations decay to near zero (0.002 inches of water in the units of the pressure sensor) for all locations around the cylinder as seen in Figure 4.4. The decay of these pressure fluctuations occurs over a depth approximately one order of magnitude larger the characteristic grain size as described by Vollmer (2002). The ratio of decay depth to d_{50} in this study is approximately 20 (decay depth is 15 mm, d_{50} is 0.85mm).

Based on data presented in the box plots of Figures 4.8, 4.11 and 4.14, the dimensionless pressure fluctuations (p'/τ) in the near surface pressure sensors compare well to the fluctuations presented in Detert (2010b). In the Detert work, fluctuations varied from +40 to -20. The 0-degree case is distributed symmetrically from +30 to – 30, the 45-degree case is distributed like Deterts's from +40 to -25 and the 90-degree case distributed from +60 to -40. Also, as in the Detert (2010b) work and as shown in Figure 4.4 the pressure fluctuations decay to the point of a non-zero constant.

Table 5.1: Dimensionless near-bed pressure fluctuations p'/τ

	Detert 2010 work, for comparison	0-degree case	45-degree case	90-degree case
Dimensionless pressure fluctuation	-20 to 40	-30 to 30	-25 to 40	-40 to 60

Correlation coefficients were calculated for each velocity component and pressure sensor three, the sensor closest to the surface. Correlations were generally low and shown in Table 5.2. The best correlations were at the 90-degree location and always negative. These low correlations likely occurred because of a loss of synchronicity between the timer associated with the pressure sensors and the clock associated with the velocimeter. The timer on the data acquisition system used in this study has an associated accuracy of 1.5%. A similar data acquisition product has a timer with an associated accuracy of 30 ppm or 500 times greater. Another possible source for decorrelation is the length of tube connecting the subsurface to the differential pressure sensor, that is, the travel time from the pressure change to occur on the bed and register in the sensor. With low correlations

and inconsistencies between trials, data were smoothed with a moving average (with a span of 500) to seek increased correlation and improve consistency of correlation between trials. Correlation magnitudes increased, Table 5.3. For the zero-degree position, and the streamwise velocity component, all correlation coefficients were small and negative, i.e. a consistent pattern in both tables 5.2 and 5.3. The most consistent correlations came with the z-component velocities and the near-surface pressure sensor. Pressure sensor three is best correlated with the vertical velocity component regardless of the radial position. This correlation is likely best because forces impinging on the bed and particles are most associated with the downward directed flow the cylinder generates. Due to the general inconsistency and lack of repeatability of these correlations, statistical properties of these data will be used in equation development rather than the instantaneous values.

Pressure sensor three is best correlated with the vertical velocity component regardless of the radial position. This correlation is likely best because forces impinging on the bed and particles are most associated with the downward directed flow the cylinder generates. Due to the general inconsistency and lack of repeatability of these correlations, statistical properties of these data were used in equation development rather than the instantaneous values.

Table5.2: Table of correlation coefficients between instantaneous velocity components and pressure fluctuations from pressure sensor three.

		Trial 1	Trial 2	Trial 3
Zero	x-vel and P3	-0.0194	-0.1289	-0.0108
Zero	y-vel & P3	0.268	0.0259	-0.0109
Zero	z-vel & P3	0.0268	0.1439	0.0576
45	x-vel and P3	-0.1106	-0.0956	0.3385
45	y-vel & P3	0.0452	-0.2031	0.2721
45	z-vel & P3	0.1	0.0133	0.1122
90	x-vel and P3	-0.1823	0.1644	0.1176
90	y-vel & P3	-0.0267	0.0686	-0.1996
90	z-vel & P3	-0.3513	-0.2368	-0.1535

Table 5.3: Table of correlation coefficients between smoothed velocity components and pressure fluctuations from pressure sensor three. Smoothing was performed with a moving average with a span of 500 (5 seconds)

		Trial 1	Trial 2	Trial 3
Zero	x-vel and P3	-0.1041	-0.2711	-0.1791
Zero	y-vel & P3	-0.02	0.0841	-0.089
Zero	z-vel & P3	0.2336	0.3028	0.227
45	x-vel and P3	-0.0834	-0.179	0.637
45	y-vel & P3	0.0774	-0.4301	0.5354
45	z-vel & P3	0.099	0.1597	0.2964
90	x-vel and P3	-0.2076	0.2419	0.2573
90	y-vel & P3	-0.0415	0.2912	-0.3103
90	z-vel & P3	-0.6676	-0.5393	-0.3779

Correlation coefficients are low for many reasons: first, the pressure sensors average pressures over an area the size of pressure port openings (i.e. the diameter of the tubes). Additionally any clock drift between the LabJack and the laptop will effect correlation where timing between the signals are important. Clock drift will not however affect global statistical parameters or transformations to the frequency domain. Detert (2010c) also reports low pressure and velocity correlations.

Velocity measurements

Some quantities in the analysis require normalization based on friction velocity. Generally, the friction velocity (0.0198 m/s in this study) is approximately 10-percent of the free stream velocity, the free stream velocity in this case is 0.20 m/s. Zero distance above the bed is difficult to define for rough beds. The Krogstad procedure also determines the velocity shift (ε) associated with a rough bed. This shift is a vertical displacement of the velocity profile and accounts for the distance between the plane of zero velocity and the top of roughened surface where the velocity is not quite zero. Values of ε/k, where k is the roughness height should be approximately 0.25. The Krogstad fit estimates ε/k as 0.16 in this study; however, Breugem (2006) states permeable beds alter the expected range of parameters suggested in literature for log-law fits of data.

The Strouhal number is a dimensionless parameter important in oscillating flows (Munson, *et al.* 2006). It is defined as

$$St = \frac{\omega l}{V} \tag{5.1}$$

where St is the Strouhal number, ω is the frequency of oscillation associated with vortex shedding, l is the characteristic length (diameter of cylinder) and v is the average velocity. At high Reynolds numbers, the Strouhal number is 0.21 for flow around a cylinder.

For the given flow condition, the Strouhal number (Eq. 5.1) predicts an oscillating frequency of 0.38 Hertz. Figure 4.1 shows pressure and velocity spectra with peaks

between 0.35 and 0.45 Hertz. Pressure sensor one was buried deepest (15 mm), pressure sensor two was at a depth of 9 mm, and pressure sensor three was buried 3 mm.

Power spectral density analysis of both the velocity and pressure signals, along with theoretical calculations, show the subsurface pressure sensors are capable of detecting turbulence characteristics associated with the turbulent velocity field and vortex shedding from the cylinder.

Bimodal behavior at the 90-degree location

Bimodal velocity distributions were modeled and described for flow around a cylinder by Kirkil (2008). However, measured velocity in this study is not bimodal because the velocity was measured at a location where the flow was still attached to the cylinder but near the separation point. Velocity decreases as flow approaches a cylinder and two stagnation points of locally high pressure form on the upstream and downstream face of the cylinder 180 degrees apart. As the flow passes the cylinder, it passes through a point of minimal local pressure. It then encounters an adverse pressure gradient created by the downstream stagnation point and the flow begins to separate from the cylinder. The location of the velocity measurements was near the point of flow separation. The observation that the pressure signal is bimodal but the velocity is not could suggest a small misalignment between the pressure sensors and the velocimeter or, more likely, the velocimeter measurements were not sufficiently close to the cylinder to detect this oscillating velocity.

Distance to bed measurements

Measurements to the bed are referenced from the face of the velocimeter transducer. Measurements are an average distance to the bed from within the intersection of the cylindrical sample volume and the bed. The sample volume has a 6 mm diameter. Regardless of the location (0, 45 or 90 degree) there is noise in the measurements to the bed. However, there are definite and opposing trends in both the 45- and 90- degree locations. These trends are likely due to a small amount of erosion over the course of the measurement. The delay is due to the time it takes for the scour/deposition to start immediately adjacent to the cylinder and grow until it intersects the velocity sampling volume.

Equation Development

Several researchers describe an exponential decay of pressure fluctuation with increasing depth of cover. Vollmer *et al*. (2002) describes the decay of pressure fluctuations inside a permeable wall (bed) with the general exponential function shown in equation 5.2 where z is the depth of cover and k is a wave number associated with the structure of the flow (i.e. fraction of sweeps, ejections, etc).

$$f(z) = Ae^{kz} + Be^{-kz} \qquad \text{eq 5.2}$$

Equation 5.2 is a generalized solution to the Navier-Stokes Equation for flow within a granular layer (Vollmer, *et al.* 2002). The positive portion of equation 5.2 has no physical meaning and Vollmer *et al.* (2002) suggest truncating it and using the first term only, z is the depth of cover and is a negative value.

Breugem *et al*. (2006), shows that inside a permeable wall with homogeneous porosity, the Reynolds-averaged Navier-Stokes equation is solved with an exponential decay function. Breugem parameterizes the decay based on porosity and wall permeability.

Detert and Parker (2010) use an exponential function to describe pressure fluctuation decay through a bi-modal sediment mixture, eq 5.3. The 2 in the exponent was determined through a least squares fit and represents a folding depth, not a wave number or porosity as done in the Voller (2002) and Breugem (2006) parameterizations, respectively.

A new equation was developed to describe pressure fluctuation decay as a function of depth of cover as well as fraction of sweeps. Equation development started with the Detert and Parker decay equation, eq 5.3.

$$\frac{\sigma_p}{\tau_0} = 2.88e^{\frac{2*y}{k_s}} \qquad \text{eq 5.3}$$

However, there are some notable differences; the Detert and Parker equation is for bimodal bed material, whereas the present experiments took place in a sand bed with uniform grain size distribution. The present study also included a cylinder in the flow field while the Detert and Parker work was in an unobstructed gravel bed. Since the present work used uniform grain size, the equivalent grain size used in equation 5.3 was changed to the median grain size d_{50} in equations 5.4 to 5.7. Notice equation 5.3 retains a

length scale related to the porous material as suggested by Vollmer (2002). Since the grain size was constant in this experiment, the d_{50} value was fixed in this regression.

To address these differences between previous and current research, modified coefficients were determined to adapt equation 5.3 for use in uniform sediment and near a cylindrical structure. This work represents a significant step forward in hyporheic zone research, as equation 5.4 is the only known equation to take into consideration processes associated with flow around a hydraulic structure.

As shown in Vollmer *et al.* (2002), Breugem *et al*. (2006) and Detert and Parker (2010), pressure fluctuations decay exponentially with the depth of cover, however, the decay functions have slightly different parameterizations.

In this work, the decay function, similar to the Detert and Parker parameterization was modified to include the fraction of sweeps and the turbulence magnitude defined by the standard deviation of the velocity fluctuations. The parameters in equation 5.4 are defined as follows:, σ_p is the standard deviation of the pressure fluctuations; τ_0 is the boundary shear stress; $b_{1,2,3}$ are regression coefficients with i corresponding to the velocity component used in the model ; y is the depth of cover (negative value); d_{50} is the median grain size; f is the fraction of sweeps; $\sigma_{v_i'}$ is the standard deviation of the velocity fluctuation in the i^{th} direction (streamwise, transverse or vertical); and U_τ is the friction velocity. Results from this equation are plotted along with the measured values in Figure 5.1.

$$\frac{\sigma_p}{\tau_0} = b_{1_i} e^{\frac{b_{2_i} y}{d_{50}}} \left(\sin\left(f \frac{\sigma_{v_i'}}{U\tau} + b_{3_i} \right) \right) \qquad \text{eq 5.4}$$

The standard deviation of the velocity fluctuations in each direction (streamwise, transverse or vertical) were examined for the best fit to the data. This factor was added to the exponential decay function proposed by Detert and Parker (2010) to help account for the influence of the cylinder as well as periodicity introduced to the turbulent bursting processes.

Table 5.4: Regression coefficients, mean square error and R-squared values using component turbulence magnitudes for fits to Equation 5.4

Using the streamwise turbulence magnitude as a regressor in Equation 5.4					
	b1	b2	b3	mse	R-squared
Zero degree	9.6	0.0991	0.8754	0.8254	0.892
45 degree	8.7813	0.0798	0.288	0.4237	0.9232
90 degree	15.0988	0.1089	0.649	1.301	0.9321
Using the transverse turbulence magnitude as a regressor in Equation 5.4					
	b1	b2	b3	mse	R-squared
Zero degree	9.608	0.0993	0.9401	0.824	0.8922
45 degree	9.0484	0.0799	0.1718	0.4546	0.9176
90 degree	14.9638	0.1088	0.5772	1.3037	0.932
Using the vertical turbulence magnitude as a regressor in Equation 5.4					
	b1	b2	b3	mse	R-squared
Zero degree	9.5611	0.0992	1.3513	0.8189	0.8929
45 degree	8.6096	0.0798	0.8498	0.3903	0.9292
90 degree	14.9185	0.1089	1.0447	1.302	0.9322

The 95% confidence intervals in equation 5.4 are shown in Table 5.5. Based on the coefficients in Table 5.3, the decay does vary with radial location. This variation is described by comparing the best-fit regression parameters with confidence intervals. Most notable, the best fit b_1 for the 90-degree location is outside the 95% confidence

interval for b_1 for either the zero or 45-degree case using the vertical velocity parameterization.

Table 5.5: 95% Confidence intervals for regression parameters in equation 5.4

	0 degree		45 degree		90 degree	
	lower CI	Upper CI	lower CI	Upper CI	lower CI	Upper CI
b1	6.32	12.8022	6.6202	10.5989	10.1127	19.7244
b2	0.0559	0.1425	0.0531	0.1065	0.0692	0.1485
b3	-3.968	6.6706	0.1214	1.5782	-1.3191	3.4084

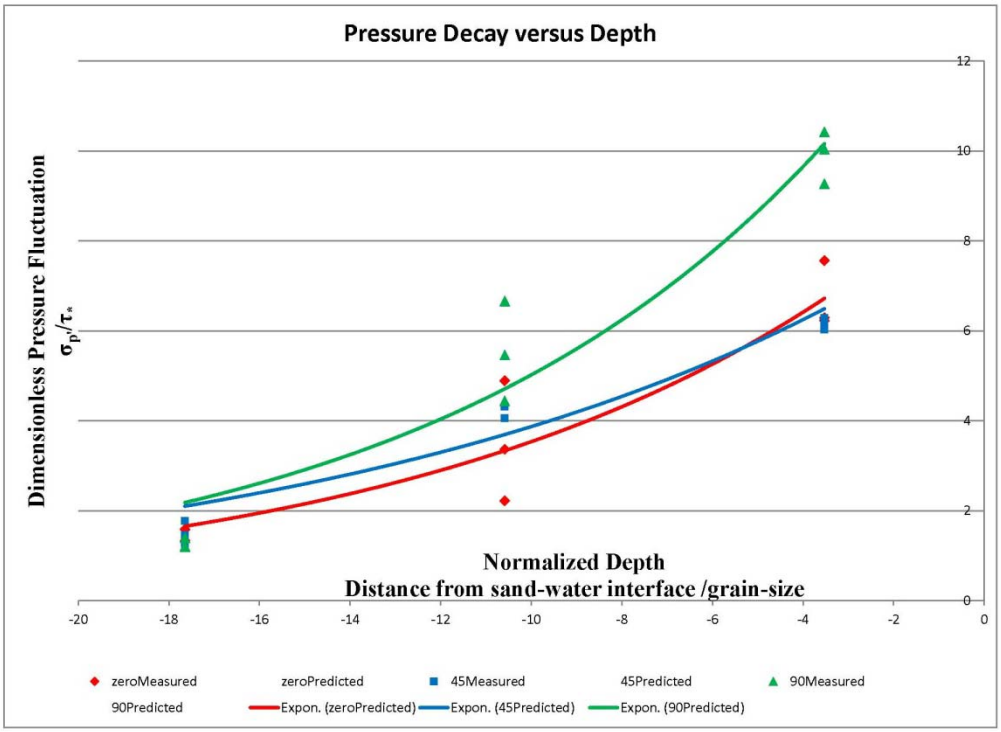


Figure 5.1: pressure fluctuation as a function of depth for all three locations and all three elevations for each trial using equation 5.4 and vertical velocity parameterization**.**

Other fits were attempted with similar parameters and/or functional form as equation 5.4, where parameters are defined the same way; k in equation 4.6 is the wave number associated with sweep frequency.

$$\frac{\sigma_p}{\tau_0} = b_1 e^{\frac{b_2 y}{d_{50}}} \left(\frac{\sigma_{v_i'}}{U\tau} * \sin(f) + \frac{\sigma_{v_i'}}{U\tau} * \cos(f) \right) \text{.....eq 5.5}$$

Equation 5.5 is plotted in Figure 5.2, general patterns observed in equation 5.4 are present in equation 5.5 but results from equation 5.5 do not have the lowest mean square error or the highest R-squared, Table 5.6.

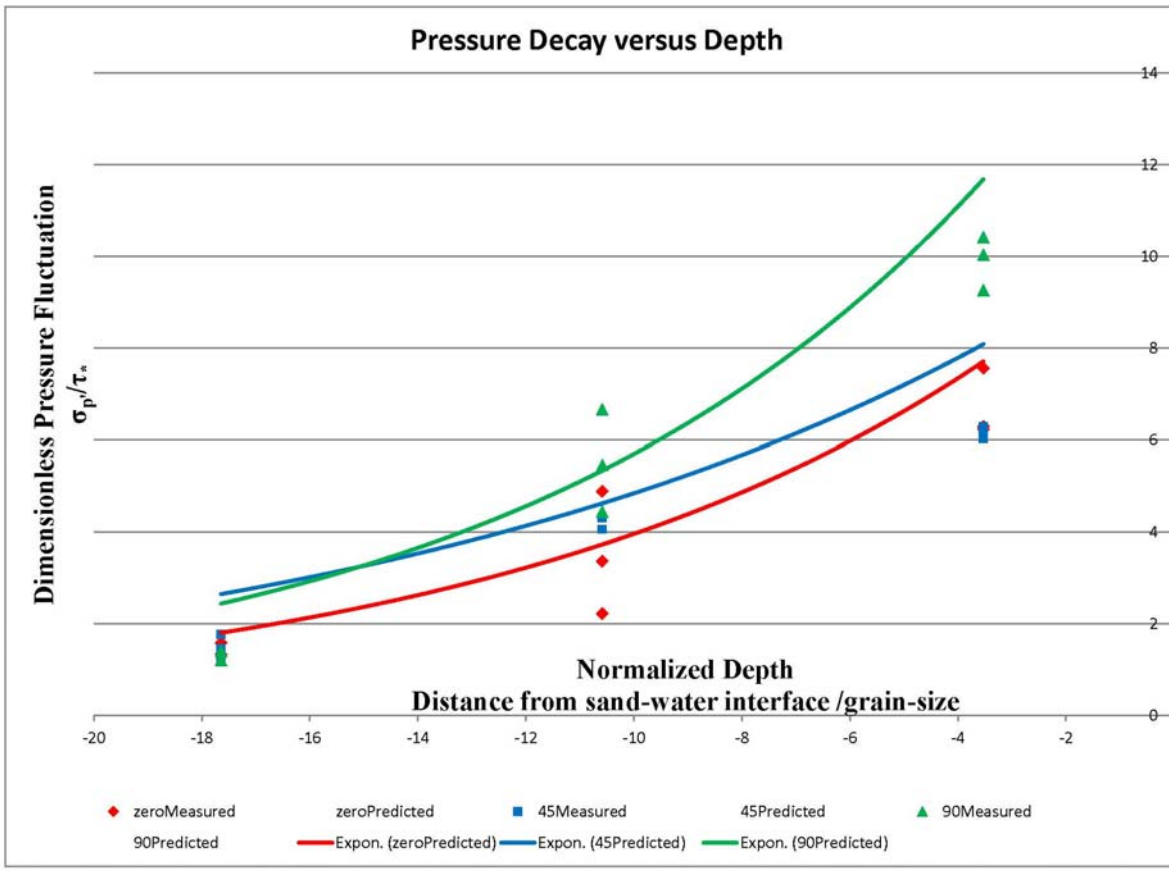


Figure 5.2: Equation 5.5 plotted using vertical velocity parameterization

Table 5.6: Regression coefficients, mean square error and R-squared values using component turbulence magnitudes for fits to Equation 5.5				
Using the streamwise turbulence magnitude as a regressor in Equation 5.5				
	b1	b2	mse	R-squared
Zero degree	5.3505	0.1027	1.091	0.8335
45 degree	2.2743	0.0792	1.4453	0.6943
90 degree	7.537	0.1117	3.2053	0.805
Using the transverse turbulence magnitude as a regressor in Equation 5.5				
	b1	b2	mse	R-squared
Zero degree	7.0058	0.1035	1.2086	0.8156
45 degree	2.0715	0.079	1.9444	0.5887
90 degree	6.9935	0.1104	1.942	0.8818
Using the vertical turbulence magnitude as a regressor in Equation 5.5				
	b1	b2	mse	R-squared
Zero degree	16.0105	0.1028	1.0865	0.8342
45 degree	4.7328	0.079	1.8138	0.6164
90 degree	14.812	0.1109	2.0345	0.8762

$$\frac{\sigma_p}{\tau_0} = b_1 e^{\frac{b_2 y}{d_{50}}} \left(\sin\left(k \frac{\sigma_{v_i'}}{U\tau} + b_3 \right) \right) \ldots\ldots \text{eq 5.6}$$

Results from Equation 5.6 are in Table 5.7. As seen in the mean square error and R-squared columns, the wave number parameterization yields poor results compared to other formulations. Note, the negative values for R-squared indicate a straight average of the data is a better predictor of normalized pressure fluctuations. This parameterization is given no further consideration and results were not plotted.

Table 5.7: Regression coefficients, mean square error and R-squared values using component turbulence magnitudes for fits to Equation 5.6					
Using the streamwise turbulence magnitude as a regressor in Equation 5.6					
	b1	b2	b3	mse	R-squared
Zero degree	12.1433	0.0977	2.1532	1.4943	0.8046
45 degree	13.9537	0.0796	2.0767	0.4503	0.9184
90 degree	-17.7849	0.1084	1.6224	1.4117	0.9264
Using the transverse turbulence magnitude as a regressor in Equation 5.6					
	b1	b2	b3	mse	R-squared
Zero degree	-4.0803	0.0587	1.5822	26.9593	-2.526
45 degree	3.1384	0.0902	2.2663	26.823	-3.8628
90 degree	-2.7589	0.0825	2.17	64.2449	-2.3507
Using the vertical turbulence magnitude as a regressor in Equation 5.6					
	b1	b2	b3	mse	R-squared
Zero degree	4.6916	0.0704	1.3707	27.1634	-2.5527
45 degree	2.8586	0.0734	2.6511	26.6498	-3.8314
90 degree	30.1543	0.1124	1.0758	19.556	-0.0199

$$\frac{\sigma_p}{\tau_0} = b_1 e^{\frac{b_2 y}{d_{50}}} * f\frac{\sigma_{v_i'}}{U\tau} + b_3 \ldots\ldots \text{eq 5.7}$$

Table 5.8 shows results from equation 5.7. Results from equation 5.7 plot similarly as equation 5.4 but the mean square error and R-squared error do not fit the data as well.

Table 5.8: Regression coefficients, mean square error and R-squared values using component turbulence magnitudes for fits to Equation 5.7					
Using the streamwise turbulence magnitude as a regressor in Equation 4.7					
	b1	b2	b3	mse	R-squared
Zero degree	12.5647	0.1598	1.3342	1.0991	0.8562
45 degree	5.8124	0.1296	1.6776	1.2453	0.7742
90 degree	16.5711	0.153	1.5818	3.313	0.8272
Using the transverse turbulence magnitude as a regressor in Equation 5.7					
	b1	b2	b3	mse	R-squared
Zero degree	16.688	0.1735	1.5068	1.1346	0.8516
45 degree	5.2178	0.1442	1.9677	1.4738	0.7328
90 degree	15.6327	0.1164	0.3007	2.2575	0.8823
Using the vertical turbulence magnitude as a regressor in Equation 5.7					
	b1	b2	b3	mse	R-squared
Zero degree	37.6347	0.1607	1.3442	1.0884	0.8576
45 degree	11.9538	0.1406	1.9023	1.4175	0.743
90 degree	32.9155	0.1225	0.548	2.3435	0.8778

In addition to equations 5.4 through 5.7, all maintaining an exponential decay factor, a simple second-order polynomial relating the normalized depth to the normalized pressure fluctuation was fit to the data at the 45-degree radial position, Figure 5.3. This was done to capture the concavity of the data at this location; the exponential functions have a slight positive concavity while the data from the 45-degree location have slight negative concavity. However, when compared to the zero-degree location and the 90-degree location, the 45-degree location has a narrow band of normalized pressure fluctuations, especially at pressure sensor two and three, Figure 4.4. While these data were measured with the same accuracy as all the other data, these data span an unusually tight range and the idea of the pressure fluctuations having negative concavity at the 45-degree location should be viewed with some skepticism. This research found no physical reason why a second-order polynomial should produce better results than the physically

based models in equations 5.4 through 5.7 but it provides a purely empirically driven model option until additional data can validate a physically based solution. Regression coefficients and fit characteristics are shown in Table 5.9.

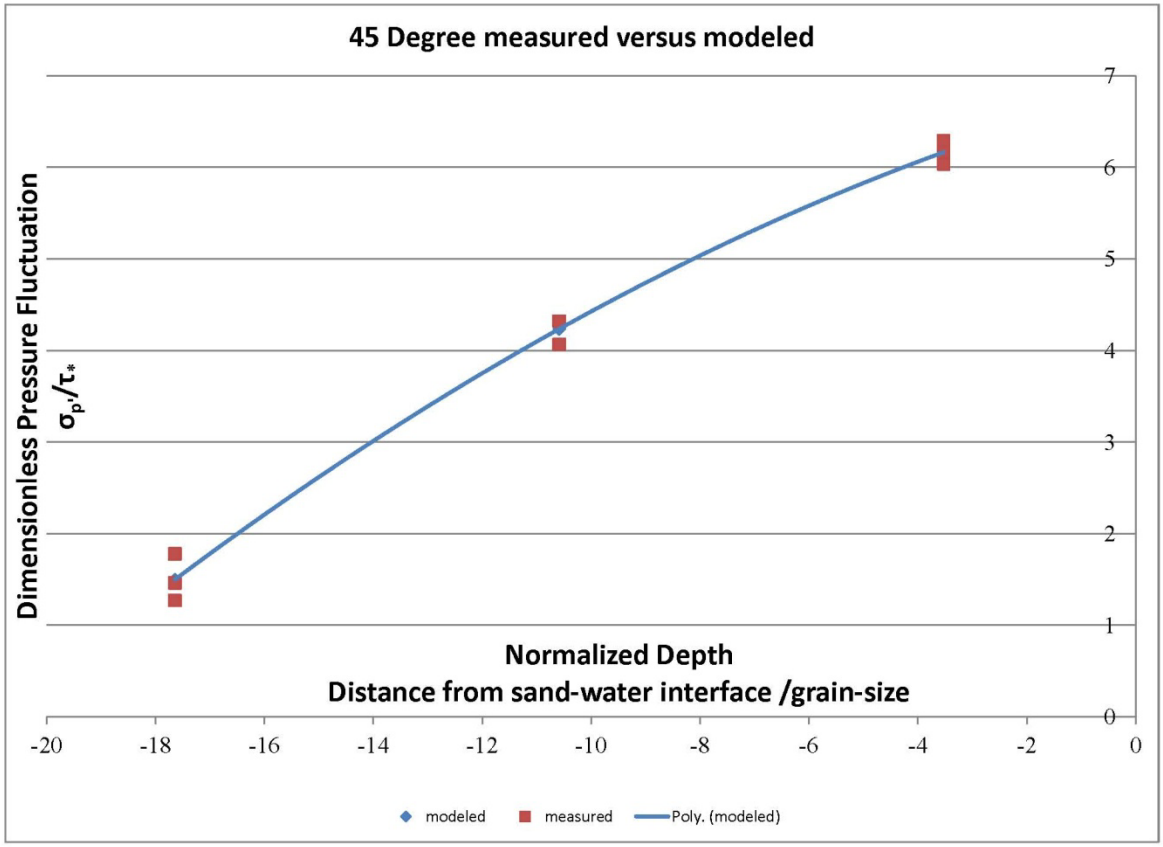


Figure 5.3: Second degree polynomial describing pressure fluctuation decay with increasing depth of cover

Table 5.7: coefficients and fit statistics for the second order polynomial					
	Quadratic	Linear	Constant	mse	R-squared
Polynomial	-0.0081	0.159	6.8249	0.15	0.999

Discussion of other work

As described by Higashino (2008) many processes are capable of inducing pressure fluctuations in the subsurface including standing waves, flow separations due to bedforms and turbulence associated with the flow. The cylinder in this work produces a zone of high pressure enabling relatively deep pore water pressure fluctuations. This standing surface roller attached to the cylinder produces a longitudinal pressure gradient similar to the one described in Khezri (2012); the force associated with this pressure gradient can be the largest force acting on a particle at the inception of motion. Equation 5.8 represents force generated by the longitudinal pressure gradient where p is pressure A_s the surface area of particle, h_s is the characteristic dimension of the particle and x is the longitudinal direction

$$F_p = \frac{\partial p}{\partial x} A_s h_s \tag{5.8}$$

The experimental setup for this research worked to minimize surface waves interference through the use of differential pressure sensors, a flow straightner at the head of the flume as well as by floating a piece of foam just downstream of the straightner. The bed was smoothed over at the start of each run; however, a slight scour hole did develop near the 90-degree measuring position. Despite the small scour, the dominant source of the pressure fluctuations is the presence of the hydraulic structure. According to work by Higashino (2009), standing waves (like a surface roller associated with flow around a cylinder) generate pressure fluctuations an order of magnitude larger than pressure fluctuations associated with a turbulent velocity field without a hydraulic structure. Additionally, pressure fluctuations an order of magnitude smaller than those

measured would prove even more difficult to detect with the pressure sensors used in this study or with other commercially available pressure sensors. This indicates the pressure fluctuations measured in this work are associated with the cylinder rather than general turbulence, bed forms or surface waves.

The force described in equation 5.8 can be modified and included in a force balance on individual particles. Forces are expected to have similar distributions as presented in the results section. This task will require additional laboratory or numerical work to asses.

Chapter 6: Conclusions

This work extends the ideas of hyporheic processes into the field of sediment transport. Specifically, forces on particles due to hyporheic process have only been investigated for material with relatively high hydraulic conductivity and only in flat beds with no obstruction. It was shown that hyporheic pressure fluctuations can be measured in medium sand, at least in the presence of a hydraulic structure. This is the first time hyporheic pressure fluctuations were measured in a medium sand with a hydraulic structure. These fluctuations play an important role in understanding the force balance associated with incipient motion of individual grains and these measurement help identify the turbulent processes responsible for generating additional lift.

Normalized, subsurface pressure fluctuations were measured and statistically modeled in this investigation. Functional forms were chosen based on previous research into pressure fluctuations in the hyporheic zone; however, models were modified to better reflect the data and processes specific to this research such as the inclusion of a hydraulic structure. Equation 5.4 with the vertical turbulence parameter was the best, physically rooted model for each radial position; specific coefficients for each position are located in Table 5.3.

The modeled data clearly follow the exponential decay model proposed by Detert and Parker (2010), Bregum et al. (2006) and Vollmer et al. (2002) at the zero and 90-degree locations. However, the model required modification to include the effects of the cylinder. The 45-degree location requires further investigation. In this case, a second-

order polynomial was fitted to the data as a purely empirical model. This polynomial was developed to address the apparent shift in concavity at the 45-degree location. There is no known or speculated physical reason for a shift in concavity at the 45-degree location. Therefore, it is expected to be an artifact of error in measurements. The measured pressures with pressure sensor two, all fall higher than predicted by the model. However, with more data, the range of normalized pressure fluctuations is expected to be more in line with the range produced at both the zero and 90-degree position. Additional data are expected to confirm the exponential decay model proposed in equation 5.4 at the 45-degree location.

Based on the coefficients in Table 5.4, the decay does vary with radial location. This variation is described by comparing the best-fit regression parameters with confidence intervals. Most notable, the best fit b_1 for the 90-degree location is outside the 95% confidence interval for b_1 for either the zero or 45-degree case using the vertical velocity parameterization.

Pressure fluctuations for pressure sensor three (the least depth of cover) were poorly correlated (unless a relatively long moving average was used). However, based on the correlations in Table 5.2, and the minimal mse and R-squared values in Table 5.4, the standard deviation of the vertical velocity fluctuations provided the best model parameterization for all three radial positions. With better correlations of instantaneous values, perhaps a more dominant process can be identified based on radial position. For this study, the vertical velocity fluctuations work best.

Three velocity zones are included in hyporheic investigations: the surface water velocity is characterized by the classic open channel profile. Hyporheic investigations occur in permeable beds, so the no slip condition is lost at the water-sediment interface. In this zone, the velocity profile decays exponentially to zero or the third zone where Darcian velocity is present. This transition region (hyporheic zone) between the open channel flow and the porous media flow is the least studied of the three regions and relationships between pressure and velocity here are only recently developed. There are many sources of pressure fluctuations in the hyporheic zone, including those associated with hydraulic structures. The present work has clearly documented the exponential decay of the subsurface pressure fluctuations associated with a circular cylinder as well as developed a statistical description of pressure fluctuations most associated with sediment transport and conditioned by the turbulent velocity field. The functional form of the Detert and Parker (2010) equation was modified to accommodate conditions of the present investigation, including uniform grain size in the medium sand fraction and the cylindrical obstruction.

Many studies investigate near surface pressure fluctuations and relate the magnitude of the root mean square of the pressure fluctuations to shear stress at a point using a simple empirical relation. The coefficient varies but is approximately 3. Other related investigations provide little insight for the present research, as those works do not include the effects of a flow obstruction (which is the focus of the present investigation). Near surface pressure measurements in the present work show the root mean square of

the pressure fluctuations are 6 to 10 times the shear stress depending on the radial posistion.

Gumbel distributions are a natural choice to model large-scale pressure fluctuations throughout the depth of observations. Restricting pressures to those greater than 1.5 standard deviations of observed values lead to a statistically significant Gumbel fits. P-values for each trial in each location and depth are listed in Table 4.3. All values are significant except those associated with two sensors in the first trial at the 90-degree location, as described earlier.

The experimental results were used to develop models to describe the decay of pressure fluctuations through the hyporheic zone and are consistent in form to pressure decay models previously proposed using similar observations in a somewhat different experimental environment: a bimodal substrate and an absence of flow obstruction. Models developed in the present investigation share the same form as existing models and fit the data well when appropriately modified to include the effects of the hydraulic structure. Equation 5.4

Chapter 7: Future Work

Additional work is recommended to better define pressure fluctuation decay at the 45-degree location. The model in equation 5.4 with the vertical velocity parameterization does a good job based on the fit statistics but the concavity is better defined with a second order polynomial. A physical reason for the second order polynomial should be sought or more likely additional data be collected in similar conditions to validate equation 5.4 for the 45-degree location.

Further work remains to link the hyporheic pressure fluctuations and the initiation of sediment transport. In terms of experimental data collection, the following list of topics, though by no means exhaustive, represent important themes to build upon the present research:

How do hyporheic pressure fluctuations around a hydraulic structure change…

- …in mixed sediment
- …with various channel blockages
- …with various shapes
- …as a function of Reynolds number or turbulence intensity

Like many new research topics, this work started with a basic case investigating what is thought to be the simple case of uniform grain-size sediment. However, due to the pervasiveness of well graded material these ideas require testing in a natural sediment before being applied in practice. A mixed grain-size will have different porosity and

permeability. Pressure fluctuation penetration depth is expected to be reduced and the rate of decay is expected to increase. Another consideration in mixed grain-size systems is the effect of bed armoring. If the bed becomes armored, the upper layer of sediment will likely be a more uniform grain-size, however the particle will be larger and such modest increases in lift and drag associated with pressure fluctuation changes are likely to be diminished. These ideas can be investigated with the same experimental setup used in this study.

The ratio of channel blockage is important in pier scour investigations, especially in flume studies and it is expected to be important in studies similar to this one. However, a limiting consideration is the ability to install pressure sensors inside the cylinder. Larger cylinders can be investigated in wider flumes or a method to install the pressure sensors in small structures needs development. Cutting a panel in the cylinder seems plausible however, roughness on the cylinder will slightly increase due to the joint and the access panel. The increased roughness would be near the pressure sensors and vulnerable to criticism.

Hydraulic structures of various shapes deserve consideration in similar research. Common pier scour equations consider square, round or sharp piers. These different shapes effect the location of flow separation on the structure and as shown in this study the flow separation point is likely important in determining the distribution of subsurface pressure fluctuation. This work could be investigated with the same setup and equipment used in this study. Care is required to accurately determine the flow separation point but

perhaps a PIV (particle imaging velocimetry) setup may work better in this case since the entire flow field could be captured, albeit at a slower frequency. A combined PIV and acoustic analysis used in the Detert (2010a) might be considered so to capture both spatial and temporal variation in flow.

Varying the turbulence intensity might also have an effect on pressure fluctuation intensity, penetration depth and rate of decay. Varying the turbulence intensity can be achieved by varying the bed treatment or adding a thin wire near the boundary to disturb the typical logarithmic velocity profile similar to work done by Sumer *et al.* (2003).

APPENDIX A –MATLAB CODE

Main Code

```
clc;
%clear all;

% %create a fresh directory

fileNames = dir('*Vectrino*.mat');
counter = length(fileNames); %no of files

[stat, mess, id] = rmdir('despiked','s');
mkdir('despiked');
%write to a output file
[stat, mess, id] = rmdir('matlab_meanData','s');
mkdir('matlab_meanData');
%directory for spectra
[stat, mess, id] = rmdir('spectra','s');
mkdir('spectra');

[stat, mess, id] = rmdir('spectraCorrected','s');
mkdir('spectraCorrected')

[stat, mess, id] = rmdir('spectraRaw','s');
mkdir('spectraRaw')

[stat, mess, id] = rmdir('pressureSpectra','s');
mkdir('pressureSpectra')

[stat, mess, id] = rmdir('pressureStuff','s');
mkdir('pressureStuff')

[stat, mess, id] = rmdir('quadrant','s');
mkdir('quadrant')


loadWeights;

LoadPressureData;
pressureResample;
pressSpectOutput = pressureSpectra(resampPressureFluct.one, ...
    resampPressureFluct.two, resampPressureFluct.three);

cmap = hsv(6);  %# Creates a 6-by-3 set of colors from the HSV colormap
%#########################################################################
####
```

```
% Use this section if you want to loop through many files. For example
did
% you collect data then move the profiler up or down some distance
creating
% overlap? That is the intention of this code
 for j=1:counter %This is for as many files that are in the
execution...
% directory
%      clear Data Filtered deNoised ReynoldsStressInst ReynoldsStress;
%     clear Raw;
%     load the data
     load(fileNames(j,1).name);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

     %load(fileNames(1).name);
     first = 1;
     last = size(Data.Profiles_VelX,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     % for looping through many files
     outputfile1 = sprintf('despiked/%s.xlsx',fileNames(j,1).name);
     outputfile2 =
sprintf('matlab_meanData/%s.xlsx',fileNames(j,1).name);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   %preallocation runs me out of memory
    tempX = zeros(size(Data.Profiles_VelX,1)); %preallocate
    tempY = tempX; tempZ1 = tempX; tempZ2= tempX; %preallocate

    for i=first:last
         tempX = Data.Profiles_VelX(:,i);
         tempY = Data.Profiles_VelY(:,i);
         tempZ1 = Data.Profiles_VelZ1(:,i);
         tempZ2 = Data.Profiles_VelZ2(:,i);

       [Filtered.Profiles_VelX(:,i), Filtered.Profiles_VelY(:,i),...
       Filtered.Profiles_VelZ1(:,i), Filtered.Profiles_VelZ2(:,i),...
       Filtered.Profiles_goodIndicies(:,i)] = ...
       simpleGaussian3d(tempX, tempY, tempZ1, tempZ2);

         ip = zeros(size(Data.Profiles_VelX,1),1);
        %[Raw.Profiles_VelX(:,i), Raw.Profiles_VelY(:,i),...
        %  Raw.Profiles_VelZ1(:,i)] = func_despike_phasespace3d_3var...
        %  (Data.Profiles_VelX(:,i), Data.Profiles_VelY(:,i),...
        %  Data.Profiles_VelZ1(:,i), 0 );% 2 for cubic spline
interpolation

%          [Filtered.Profiles_VelX(:,i),
Filtered.Profiles_VelY(:,i),...
%            Filtered.Profiles_VelZ1(:,i)] = ...
```

```
%            func_despike_phasespace3d_3var(Data.Profiles_VelX(:,i),...
%            Data.Profiles_VelY(:,i), Data.Profiles_VelZ1(:,i), 2 );
%          % 2 for cubic spline interpolation in line above and below
%          Filtered.Profiles_VelZ2(:,i) = ...
%            func_despike_phasespace3d(Data.Profiles_VelZ2(:,i), 2 );

       % ip(:,i) = ~isnan(Raw.Profiles_VelX(:,i));

%        [Filtered.Profiles_VelX(:,i), Filtered.Profiles_VelY(:,i),...
%        Filtered.Profiles_VelZ1(:,i),
Filtered.Profiles_VelZ2(:,i),...(:,i)
%        Filtered.Profiles_goodIndicies(:,i)] = ...
%        func_despike_phasespace3d_3var(tempX, tempY, tempZ1, 2 );% 2
for
%        cubic spline interpolation

    end


    clear tempX tempY tempZ1 tempZ2 %save memory
%##########################################################################
####
    beamVel = beam(Data.Profiles_VelX, Data.Profiles_VelY, ...
      Data.Profiles_VelZ1, Data.Profiles_VelZ2, ...
      Config.ProbeCalibration_calibrationMatrix);

    for i=first:last
         tempX = beamVel.one(:,i);
         tempY = beamVel.two(:,i);
         tempZ1 = beamVel.three(:,i);
         tempZ2 = beamVel.four(:,i);

         [beamVel.Filtered.one(:,i), beamVel.Filtered.two(:,i),...
           beamVel.Filtered.three(:,i), beamVel.Filtered.four(:,i),...
           beamVel.Filtered.goodIndicies(:,i)] = ...
           simpleGaussian3d(tempX, tempY, tempZ1, tempZ2);

    end

    clear tempX tempY tempZ1 tempZ2 %save memory
%##########################################################################
####
%exportingdata

%####################Uncomment below when I'm finished
% xlswrite(outputfile1,Filtered.Profiles_VelX,'xvel');
% xlswrite(outputfile1,Filtered.Profiles_VelY,'yvel');
% xlswrite(outputfile1,Filtered.Profiles_VelZ1,'zvel1');
% xlswrite(outputfile1,Filtered.Profiles_VelZ2,'zvel2');

%the idea of removing bad SNR and corelations first is why consider
```

```
%data I know to be bad in the Gaussian filter, it will just increase
%the size of the standard deviation? Remove the data I know
%to be bad then in the Gaussian filter the standard deviations will be
%smaller and the filter will remove less data. Either way a similar
%amount of datat is removed but this way makes more sense to me -- TJC

%The problem with the above comment is if I remove data with bad data
%quality indicators, I'll end up with columns of differnt lengths --
%TJC


%Determine good if SNR is sufficiently high
    goodSNRx = pickGoodData(Data.Profiles_SNRBeam1, 25);
    goodSNRy = pickGoodData(Data.Profiles_SNRBeam2, 25);
    goodSNRz1 = pickGoodData(Data.Profiles_SNRBeam3, 25);
    goodSNRz2 = pickGoodData(Data.Profiles_SNRBeam4, 25);

    %get the index of the good SNR
    goodSNRxIdx = ~isnan(goodSNRx);
    goodSNRyIdx = ~isnan(goodSNRy);
    goodSNRz1Idx = ~isnan(goodSNRz1);
    goodSNRz2Idx = ~isnan(goodSNRz2);
    goodSNR = goodSNRxIdx & goodSNRyIdx & goodSNRz1Idx & goodSNRz2Idx;

    clear goodSNRx goodSNRy goodSNRz1 goodSNRz2
    clear goodSNRxIdx goodSNRyIdx goodSNRz1Idx goodSNRz2Idx

    %Determine good if correlation is sufficiently high
    goodCorrX = pickGoodData(Data.Profiles_CorBeam1, 75);
    goodCorrY = pickGoodData(Data.Profiles_CorBeam2, 75);
    goodCorrZ1 = pickGoodData(Data.Profiles_CorBeam3, 75);
    goodCorrZ2 = pickGoodData(Data.Profiles_CorBeam4, 75);

   %get the index of good correlations
    goodCorrXIdx = ~isnan(goodCorrX);
    goodCorrYIdx = ~isnan(goodCorrY);
    goodCorrZ1Idx = ~isnan(goodCorrZ1);
    goodCorrZ2Idx = ~isnan(goodCorrZ2);
    goodCorr = goodCorrXIdx & goodCorrYIdx & goodCorrZ1Idx &
goodCorrZ2Idx;

    clear goodCorrX goodCorrY goodCorrZ1 goodCorrZ2
    clear goodCorrXIdx goodCorrYIdx goodCorrZ1Idx goodCorrZ2Idx


%##########################################################################
####
%This section is used to reduce noise in the vertical velocity
component.
% sigmaSquared = zeros(1,size(Data.Profiles_VelX,2));%preallocate
% verticalNoise = zeros(1,size(Data.Profiles_VelX,2));%preallocate
% for i = first:last
```

```
%     deNoisedData(:, i) = deNoise(Filtered.Fluct.x(:,i),...
%         Filtered.Fluct.y(:,i));
%
%     deNoisedZ1Z2(:, i) = deNoise(Filtered.Fluct.z1(:,i),...
%         Filtered.Fluct.z2(:,i));
%
%     deNoisedXY(:, i) = deNoise(Filtered.Fluct.x(:,i),...
%         Filtered.Fluct.y(:,i));
%
% %     verticalNoise(:,i) = verticalNoiseSpectra...
% %         (Filtered.Profiles_VelX(:,i),Filtered.Profiles_VelY(:,i));
%
% sigmaSquared(i) = abs(deNoisedXY(i).sigSqrdOut)./weights(i,2)';
% end

%sigmaSquared = abs(deNoisedZ1Z2.sigSqrdOut)./weights(:,2)';
%##########################################################################
####

   %read bottom depth from Data and gaussian filter it
    [Filtered.BottomDistance, goodBottomDistanceIndex] =...
        simpleGaussian(Data.BottomCheck_BottomDistance);
    avgBottomDist = nanmean(Filtered.BottomDistance);
    stdBottomDist = nanstd(Filtered.BottomDistance);
   %xlswrite('despiked/data.xls',Filtered.BottomDistance,'depth');

   %read profile range
    depths_computed = avgBottomDist-Data.Profiles_Range;
%
   %temperature
   %xlswrite('despiked/data.xls',...
   %    Data.Profiles_Temperature(first:last,1),'temperature');

%##########################################################################
####
%I took out this entire block because I'm not sure why I need this
stuff in
%Excel. I think it is just slowing down execution. Uncomment if I ever
%need it

% %Correlations
% %we have 31 sets of despiked data, put them in one excel file
% %most top depth to most bottom depth
% xlswrite('despiked/data.xls',Data.Profiles_CorBeam1,'xcor');
% xlswrite('despiked/data.xls',Data.Profiles_CorBeam2,'ycor');
% xlswrite('despiked/data.xls',Data.Profiles_CorBeam3,'zcor1');
% xlswrite('despiked/data.xls',Data.Profiles_CorBeam4,'zcor2');
%
% %SNR
% %we have 31 sets of despiked data, put them one excel file
% %most top depth to most bottom depth
% xlswrite('despiked/data.xls',Data.Profiles_SNRBeam1,'xSNR');
% xlswrite('despiked/data.xls',Data.Profiles_SNRBeam2,'ySNR');
```

```
% xlswrite('despiked/data.xls',Data.Profiles_SNRBeam3,'z1SNR');
% xlswrite('despiked/data.xls',Data.Profiles_SNRBeam4,'z2SNR');
%
% %Amplitudes
% % %we have 31 sets of despiked data,put them in one excel file
% % %most top depth to most bottom depth
% xlswrite('despiked/data.xls',Data.Profiles_AmpBeam1,'xAmp');
% xlswrite('despiked/data.xls',Data.Profiles_AmpBeam2,'yAmp');
% xlswrite('despiked/data.xls',Data.Profiles_AmpBeam3,'z1Amp');
% xlswrite('despiked/data.xls',Data.Profiles_AmpBeam4,'z2Amp');
%#########################################################################
####
% %FILTER DATA
%
% %complete analysis of velocity data
    for i = first:last
        Filtered.Inst = struct('x', Filtered.Profiles_VelX, 'y',...
            Filtered.Profiles_VelY, 'z1', Filtered.Profiles_VelZ1, ...
            'z2', Filtered.Profiles_VelZ2);
       %Filtered.Mean = struct('x', mean(Filtered.Inst.x(:,i)), 'y',
...
       %    mean(Filtered.Inst.y(:,i)), 'z1', ...
       %    mean(Filtered.Inst.z1(:,i)), 'z2',
mean(Filtered.Inst.z2(:,i)));
        Filtered.Mean.x(:,i) = mean(Filtered.Inst.x(:,i));
        Filtered.Mean.y(:,i) = mean(Filtered.Inst.y(:,i));
        Filtered.Mean.z1(:,i) = mean(Filtered.Inst.z1(:,i));
        Filtered.Mean.z2(:,i) = mean(Filtered.Inst.z2(:,i));
       %Filtered.Fluct = struct('x', Filtered.Inst.x(:,i) - ...
       %    Filtered.Mean.x(i), 'y',Filtered.Inst.y(:,i) - ...
       %    Filtered.Mean.y(i) , 'z1', Filtered.Inst.z1(:,i) - ...
       %    Filtered.Mean.z1(i), 'z2', Filtered.Inst.z2(:,i) - ...
       %    Filtered.Mean.z2(i));
        Filtered.Fluct.x(:,i) = Filtered.Inst.x(:,i) -
Filtered.Mean.x(:,i);
        Filtered.Fluct.y(:,i) = Filtered.Inst.y(:,i) -
Filtered.Mean.y(:,i);
        Filtered.Fluct.z1(:,i) = Filtered.Inst.z1(:,i) - ...
            Filtered.Mean.z1(:,i);
        Filtered.Fluct.z2(:,i) = Filtered.Inst.z2(:,i) - ...
            Filtered.Mean.z2(:,i);
       %velocityFiltered(i) = struct('inst', velInstFiltered(i),
'mean',...
       %    velMeanFiltered(i), 'fluct', velFluctFiltered(i));

%     Raw.Inst = struct('x', Filtered.Profiles_VelX, 'y',...
%           Filtered.Profiles_VelY, 'z1', Filtered.Profiles_VelZ1,...
%           'z2', Filtered.Profiles_VelZ2);
%     %Filtered.Mean = struct('x', mean(Filtered.Inst.x(:,i)), 'y', ...
%           mean(Filtered.Inst.y(:,i)), 'z1', ...
%           mean(Filtered.Inst.z1(:,i)), 'z2',
mean(Filtered.Inst.z2(:,i)));
%     Raw.Mean.x(:,i) = mean(Raw.Inst.x(:,i));
```

```
%     Raw.Mean.y(:,i) = mean(Raw.Inst.y(:,i));
%     Raw.Mean.z1(:,i) = mean(Raw.Inst.z1(:,i));
% %     Raw.Mean.z2(:,i) = mean(Raw.Inst.z2(:,i));
%     %Filtered.Fluct = struct('x', Filtered.Inst.x(:,i) - ...
%           Filtered.Mean.x(i), 'y',Filtered.Inst.y(:,i) - ...
%           Filtered.Mean.y(i) , 'z1', Filtered.Inst.z1(:,i) - ...
%           Filtered.Mean.z1(i), 'z2', Filtered.Inst.z2(:,i) - ...
%           Filtered.Mean.z2(i));
%     Raw.Fluct.x(:,i) = Raw.Inst.x(:,i) - Raw.Mean.x(:,i);
%     Raw.Fluct.y(:,i) = Raw.Inst.y(:,i) - Raw.Mean.y(:,i);
%     Raw.Fluct.z1(:,i) = Raw.Inst.z1(:,i) - Raw.Mean.z1(:,i);
% %     Raw.Fluct.z2(:,i) = Raw.Inst.z2(:,i) - Filtered.Mean.z2(:,i);
%     %velocityFiltered(i) = struct('inst', velInstFiltered(i), 'mean',
...
%           velMeanFiltered(i), 'fluct', velFluctFiltered(i));


         beamVel.Filtered.mean.one(:,i) =
mean(beamVel.Filtered.one(:,i));
         beamVel.Filtered.mean.two(:,i) =
mean(beamVel.Filtered.two(:,i));
         beamVel.Filtered.mean.three(:,i) =
mean(beamVel.Filtered.three(:,i));
         beamVel.Filtered.mean.four(:,i) =
mean(beamVel.Filtered.four(:,i));

         beamVel.Fluct.one(:,i) = beamVel.Filtered.one(:,i) - ...
             beamVel.Filtered.mean.one(:,i);
         beamVel.Fluct.two(:,i) = beamVel.Filtered.two(:,i) - ...
             beamVel.Filtered.mean.two(:,i);
         beamVel.Fluct.three(:,i) = beamVel.Filtered.three(:,i) - ...
             beamVel.Filtered.mean.three(:,i);
         beamVel.Fluct.four(:,i) = beamVel.Filtered.four(:,i) - ...
             beamVel.Filtered.mean.four(:,i);
    end



    for i = first:last
         deNoisedData(:, i) = deNoise(Filtered.Fluct.x(:,i), ...
             Filtered.Fluct.y(:,i), Filtered.Fluct.z1(:,i),...
             Filtered.Fluct.z2(:,i),i,j);

%     deNoisedZ1Z2(:, i) = deNoise(Filtered.Fluct.z1(:,i),...
%         Filtered.Fluct.z2(:,i));
%
%     deNoisedXY(:, i) = deNoise(Filtered.Fluct.x(:,i),...
%         Filtered.Fluct.y(:,i));

         sigmaSquared(i) = deNoisedData(i).sigSqrdOut./weights(i,2)';

         deNoised.Fluct.x(:,i) = sqrt((Filtered.Fluct.x(:,i)).^2 - ...
```

```
            weights(i,1)*sigmaSquared(i));
        deNoised.Fluct.y(:,i) = sqrt((Filtered.Fluct.y(:,i)).^2 - ...
            weights(i,1)*sigmaSquared(i));
        deNoised.Fluct.z1(:,i) = sqrt((Filtered.Fluct.z1(:,i)).^2 -
...
            weights(i,2)*sigmaSquared(i));
        deNoised.Fluct.z2(:,i) = sqrt((Filtered.Fluct.z2(:,i)).^2 -
...
            weights(i,2)*sigmaSquared(i));

        reduceA(i) = weights(i,1)*sigmaSquared(i);
        reduceB(i) = weights(i,2)*sigmaSquared(i);


%     deNoised.Fluct.x(:,i) = deNoised.Inst.x(:,i) -
deNoised.Mean.x(:,i);
%     deNoised.Fluct.y(:,i) = deNoised.Inst.y(:,i) -
deNoised.Mean.y(:,i);
%     deNoised.Fluct.z1(:,i) = deNoised.Inst.z1(:,i) -
deNoised.Mean.z1(:,i);
%     deNoised.Fluct.z2(:,i) = deNoised.Inst.z2(:,i) -
deNoised.Mean.z2(:,i);
%
%     %correlations
%     correlationFiltered(i) = struct...
%               ('x', Filtered.Profiles_CorBeam1(first:last,i), ...
%               'y', Filtered.Profiles_CorBeam2(first:last,i), ...
%               'z1', Filtered.Profiles_CorBeam3(first:last,i), ...
%               'z2', Filtered.Profiles_CorBeam4(first:last,i));
%     %SNR
%     snrFiltered(i) = struct...
%               ('x', Filtered.Profiles_SNRBeam1(first:last,i), ...
%               'y', Filtered.Profiles_SNRBeam2(first:last,i), ...
%               'z1', Filtered.Profiles_SNRBeam3(first:last,i), ...
%               'z2', Filtered.Profiles_SNRBeam4(first:last,i));
%     %Amp
%     amplitudeFiltered(i) = struct...
%               ('x', Filtered.Profiles_AmpBeam1(first:last,i),...
%               'y', Filtered.Profiles_AmpBeam2(first:last,i), ...
%               'z1', Filtered.Profiles_AmpBeam3(first:last,i), ...
%               'z2', Filtered.Profiles_AmpBeam4(first:last,i));

%
% %perform a spectral analysis for each vertical velocity bin, This
creates
% graphs only
% for i = first:last
%         PwelchAll;
% end
%
%
%
%     %collect the velocities that meet the quality control conditions
```

```
%     cleanVelocityInst(i) = struct...
%               ('x', velInstFiltered(i).x(goodDataIdxX & goodDataIdxY
...
%               & goodDataIdxZ1 & goodDataIdxZ2), ...
%               'y', velInstFiltered(i).y(goodDataIdxX & ...
%               goodDataIdxY & goodDataIdxZ1 & goodDataIdxZ2), ...
%               'z1', velInstFiltered(i).z1(goodDataIdxX & ...
%               goodDataIdxY & goodDataIdxZ1 & goodDataIdxZ2), ...
%               'z2', velInstFiltered(i).z1(goodDataIdxX & ...
%               goodDataIdxY & goodDataIdxZ1 & goodDataIdxZ2));
%     %determine turbulence intensities
%      turbulentIntnst.x(i) = nanstd(deNoised.Fluct.x(:,i));
%      turbulentIntnst.y(i) = nanstd(deNoised.Fluct.y(:,i));
%      turbulentIntnst.z1(i) = nanstd(deNoised.Fluct.z1(:,i));
%      turbulentIntnst.z2(i) = nanstd(deNoised.Fluct.z2(:,i));

%      turbulentIntnst.x(i) = nanstd(Filtered.Fluct.x(:,i));
%      turbulentIntnst.y(i) = nanstd(Filtered.Fluct.y(:,i));
%      turbulentIntnst.z1(i) = nanstd(Filtered.Fluct.z1(:,i));
%      turbulentIntnst.z2(i) = nanstd(Filtered.Fluct.z2(:,i));

        turbulentIntnst.x(i) = sqrt(sum(deNoised.Fluct.x(:,i).^2));
        turbulentIntnst.y(i) = sqrt(sum(deNoised.Fluct.y(:,i).^2));
        turbulentIntnst.z1(i) = sqrt(sum(deNoised.Fluct.z1(:,i).^2));
        turbulentIntnst.z2(i) = sqrt(sum(deNoised.Fluct.z2(:,i).^2));

%         turbulentIntnst.all(i) = sqrt(mean([Filtered.Fluct.x(:,i).^2
...
%             Filtered.Fluct.y(:,i).^2  ...
%             Filtered.Fluct.z1(:,i).^2]));
%
%     cleanVelocityMean(i) = struct...
%               ('x', nanmean(cleanVelocityInst(i).x),...
%               'y', nanmean(cleanVelocityInst(i).y),...
%               'z1', nanmean(cleanVelocityInst(i).z1),...
%               'z2', nanmean(cleanVelocityInst(i).z2));
%     cleanVelocityFluct(i) = struct...
%               ('x', cleanVelocityInst(i).x-cleanVelocityMean(i).x,
...
%               'y', cleanVelocityInst(i).y-cleanVelocityMean(i).y, ...
%               'z1', cleanVelocityInst(i).z1-cleanVelocityMean(i).z1,
...
%               'z2', cleanVelocityInst(i).z2-cleanVelocityMean(i).z2);
%
%     %skewness of velocity fluctuations
        skw.x(i) = skewness(deNoised.Fluct.x(:,i));
        skw.y(i) = skewness(deNoised.Fluct.y(:,i));
        skw.z1(i) = skewness(deNoised.Fluct.z1(:,i));
        skw.z2(i) = skewness(deNoised.Fluct.z2(:,i));
%
%     %snr
%     cleanSnrInst(i) = struct('x', snrFiltered(1,i).x(goodDataIdxX &
...
```

```
%           goodDataIdxY & goodDataIdxZ1 & goodDataIdxZ2), ...
%           'y', snrFiltered(1,i).y(goodDataIdxX & goodDataIdxY & ...
%           goodDataIdxZ1 & goodDataIdxZ2), 'z1',...
%           snrFiltered(1,i).z1(goodDataIdxX & goodDataIdxY & ...
%           goodDataIdxZ1 & goodDataIdxZ2), 'z2', ...
%           snrFiltered(1,i).z1(goodDataIdxX & goodDataIdxY & ...
%           goodDataIdxZ1 & goodDataIdxZ2));
%     cleanSnrMean(i) = struct('x', nanmean(cleanSnrInst(i).x), ...
%           'y', nanmean(cleanSnrInst(i).y), 'z1', ...
%           nanmean(cleanSnrInst(i).z1), 'z2',
nanmean(cleanSnrInst(i).z2));
%     %correlation
%     cleanCorrelationInst(i) = struct('x', ...
%           correlationFiltered(1,i).x(goodDataIdxX & goodDataIdxY &
...
%           goodDataIdxZ1 & goodDataIdxZ2), 'y', ...
%           correlationFiltered(1,i).y(goodDataIdxX & goodDataIdxY &
...
%           goodDataIdxZ1 & goodDataIdxZ2), 'z1', ...
%           correlationFiltered(1,i).z1(goodDataIdxX & goodDataIdxY ...
%           & goodDataIdxZ1 & goodDataIdxZ2) , 'z2',...
%           correlationFiltered(1,i).z1(goodDataIdxX...
%           & goodDataIdxY & goodDataIdxZ1 & goodDataIdxZ2));
%     cleanCorrelationMean(i) = struct('x', ...
%           nanmean(cleanCorrelationInst(i).x), ...
%               'y', nanmean(cleanCorrelationInst(i).y), ...
%               'z1', nanmean(cleanCorrelationInst(i).z1), ...
%               'z2', nanmean(cleanCorrelationInst(i).z2));
%
%     %amplitude
%     amplitudeFilteredInst(i) = struct...
%           ('x', amplitudeFiltered(1,i).x(goodDataIdxX & goodDataIdxY
...
%           & goodDataIdxZ1 & goodDataIdxZ2), 'y', ...
%           amplitudeFiltered(1,i).y(goodDataIdxX & goodDataIdxY & ...
%           goodDataIdxZ1 & goodDataIdxZ2), ...
%           'z1', amplitudeFiltered(1,i).z1(goodDataIdxX &
goodDataIdxY...
%           & goodDataIdxZ1 & goodDataIdxZ2), ...
%           'z2', amplitudeFiltered(1,i).z2(goodDataIdxX & ...
%           goodDataIdxY & goodDataIdxZ1 & goodDataIdxZ2));
%     cleanAmplitudeMean(i) = struct...
%           ('x', nanmean(amplitudeFilteredInst(i).x), ...
%           'y', nanmean(amplitudeFilteredInst(i).y), ...
%           'z1', nanmean(amplitudeFilteredInst(i).z1), ...
%           'z2', nanmean(amplitudeFilteredInst(i).z2));
%


%     %Just use z1 for Reynolds stress computations, otherwise seems to
messy
%     %Create Reynolds Stresses
        ReynoldsStressInst.x(:,i) = ...
            deNoised.Fluct.x(:,i).*deNoised.Fluct.y(:,i);
```

```
            ReynoldsStressInst.y(:,i) = ...
                deNoised.Fluct.x(:,i).*deNoised.Fluct.z1(:,i);
            ReynoldsStressInst.z1(:,i) = ...
                deNoised.Fluct.y(:,i).*deNoised.Fluct.z1(:,i);

            ReynoldsStress.x(:,i) = ...
nanstd(deNoised.Fluct.x(:,i)).*nanstd(deNoised.Fluct.y(:,i));
            ReynoldsStress.y(:,i) = ...
nanstd(deNoised.Fluct.x(:,i)).*nanstd(deNoised.Fluct.z1(:,i));
            ReynoldsStress.z1(:,i) = ...
nanstd(deNoised.Fluct.y(:,i)).*nanstd(deNoised.Fluct.z1(:,i));


    end

    for i = first:last;
        %NORMALIZED PRESSURES
        myPressMap3(i) = jPDFCondNormPress(Filtered.Fluct.x(:,i),...
            Filtered.Fluct.y(:,i), Filtered.Fluct.z1(:,i), ...
            resampPressureFluct.three, depths_computed(i));

%             myPressMap3(i) =
jPDFCondNormPress(Filtered.Fluct.x(:,i),...
%               Filtered.Fluct.y(:,i), Filtered.Fluct.z1(:,i), ...
%               resampPressureFluct.three, depths_computed(i), ...
%               Filtered.Inst.x(:,i));
%
        figure;
        hold on
        waterfall(myPressMap3(i).xAxisValuesNorm, ...
            myPressMap3(i).zAxisValuesNorm,
myPressMap3(i).PressureMapZ)
        xlabel({'streamwise (cm/s)'});ylabel({'vertical(cm/s)'});
        zlabel({'pressure (dimensionless)'});
        grid on

        figure;
        hold on
        waterfall(myPressMap3(i).xAxisValuesNorm, ...
            myPressMap3(i).yAxisValuesNorm,
myPressMap3(i).PressureMapY)
        xlabel({'streamwise (cm/s)'});ylabel({'transverse(cm/s)'});
        zlabel({'pressure (dimensionless)'});
        grid on

        someString1 =
['SaveAllFigures(''pressureStuff\pressureThree'...
            int2str(i)];
        someString2 = [''',''jpg'')'];
```

```
        someString3 = [someString1 someString2];
        eval(someString3);

        close all;

        myPressMap2(i) = jPDFCondNormPress(Filtered.Fluct.x(:,i),...
        Filtered.Fluct.y(:,i), Filtered.Fluct.z1(:,i), ...
        resampPressureFluct.two, depths_computed(i));



        figure;
        hold on
        waterfall(myPressMap2(i).xAxisValuesNorm, ...
            myPressMap2(i).zAxisValuesNorm,
myPressMap2(i).PressureMapZ)
        xlabel({'streamwise (cm/s)'});ylabel({'vertical(cm/s)'});
        zlabel({'pressure (dimensionless)'});
        grid on

        figure;
        hold on
        waterfall(myPressMap2(i).xAxisValuesNorm, ...
            myPressMap2(i).yAxisValuesNorm,
myPressMap2(i).PressureMapY)
        xlabel({'streamwise (cm/s)'});ylabel({'transverse(cm/s)'});
        zlabel({'pressure (dimensionless)'});
        grid on

        someString1 = ...
            ['SaveAllFigures(''pressureStuff\pressureTwo' int2str(i)];
        someString2 = [''',''jpg'')'];
        someString3 = [someString1 someString2];
        eval(someString3);

        close all;

        myPressMap1(i) = jPDFCondNormPress(Filtered.Fluct.x(:,i),...
        Filtered.Fluct.y(:,i), Filtered.Fluct.z1(:,i), ...
        resampPressureFluct.one, depths_computed(i));
        figure;
        hold on
        waterfall(myPressMap1(i).xAxisValuesNorm, ...
            myPressMap1(i).zAxisValuesNorm,
myPressMap1(i).PressureMapZ)
        xlabel({'streamwise (cm/s)'});ylabel({'vertical(cm/s)'});
        zlabel({'pressure (dimensionless)'});
        grid on

        figure;
        hold on
        waterfall(myPressMap1(i).xAxisValuesNorm, ...
```

```
            myPressMap1(i).yAxisValuesNorm,
myPressMap1(i).PressureMapY)
        xlabel({'streamwise (cm/s)'});ylabel({'transverse(cm/s)'});
        zlabel({'pressure (dimensionless)'});
        grid on

        someString1 = ['SaveAllFigures(''pressureStuff\pressureOne'...
            int2str(i)];
        someString2 = [''',''jpg'')'];
        someString3 = [someString1 someString2];
        eval(someString3);

        close all;

        finalBins = [length(myPressMap1(1,i).xAxisValues), ...
            length(myPressMap1(1,i).yAxisValues), ...
            length(myPressMap1(i).zAxisValues)];


        %Quadrant Analysis
        quadOut(i) = quadrant(Filtered.Fluct.x(:,i),...
            Filtered.Fluct.y(:,i), Filtered.Fluct.z1(:,i), ...
            Filtered.Fluct.z2(:,i),finalBins,0);

        figure;
        hold on
        hist3(quadOut(i).M, quadOut(i).nBins1);
        xlabel({'streamwise (m/s)'});ylabel({'vertical(m/s)'});
        zlabel({'counts'});
        grid on
        view(3);

        figure;
        hold on
        hist3(quadOut(i).M2, quadOut(i).nBins2);
        xlabel({'streamwise (m/s)'});ylabel({'transverse(m/s)'});
        zlabel({'counts'});
        grid on
        view(3);

        someString1 = ['SaveAllFigures(''quadrant\hist' int2str(i)];
        someString2 = [''',''jpg'')'];
        someString3 = [someString1 someString2];
        eval(someString3);

        close all;

%         figure;
%         hold on
%         %contour(quadOut(i).xAxisValues, ...
%               quadOut(i).yAxisValuesVertVel,
quadOut(i).quadAnalysisZ);
```

```
%         contour(quadOut(i).yAxisValuesVertVel, ...
%               quadOut(i).xAxisValues,  quadOut(i).quadAnalysisZ);
%         xlabel({'streamwise'; 'fluctuations (cm/s)'});...
%               ylabel({'vertical';'fluctuations (cm/s)'});
%         grid on
%
%         figure;
%         hold on
%         contour(quadOut(i).xAxisValues, quadOut(i).yAxisValues, ...
%               quadOut(i).quadAnalysisY);
%         xlabel({'streamwise'; 'fluctuations (cm/s)'});...
%               ylabel({'transverse';'fluctuations (cm/s)'});
%         grid on
%
%         someString1 = ['SaveAllFigures(''quadrant\contour'
int2str(i)];
%         someString2 = ['''',''jpg'')'];
%         someString3 = [someString1 someString2];
%         eval(someString3);
%
%         close all;



%RAW PRESSURES
%         myPressMap(i) = jPDFCondPress(Filtered.Fluct.x(:,i),...
%             Filtered.Fluct.y(:,i), Filtered.Fluct.z1(:,i), ...
%             resampPressureFluct.three, resampPressureFluct.two, ...
%             resampPressureFluct.one);


        sweepPressure = zeros(size(myPressMap3(1,i).uPressNorm));
        %pick out pressures which occurred during a sweep or ejection
  %sweepPressure = myPressMap3(i).uPressNorm(quadOut(i).sweepIdx);
        %ejectPressure(i) =
myPressMap3(i).uPressNorm(quadOut(i).ejectIdx);

    end

    sigma3 = std(resampPressureFluct.three);
    sigma2 = std(resampPressureFluct.two);
    sigma1 = std(resampPressureFluct.one);



%
   %mean vel
    headMeanVel = {'AvgVx (m/s)','AvgVy (m/s)',...
        'AvgVz1 (m/s)','AvgVz2 (m/s)'};
   %put velocities arrays in a matrix
    meanVel = [Filtered.Mean.x; Filtered.Mean.y;...
        Filtered.Mean.z1;Filtered.Mean.z2]';
   %turbulent intensities
```

```
    headVelIntnst = {'RMSVx" (m/s)','RMSVy" (m/s)',...
        'RMSVz1" (m/s)','RMSVz2" (m/s)'};
   %put turbulent intensity arrays in a matrix
    velIntnst = [turbulentIntnst.x; turbulentIntnst.y;...
        turbulentIntnst.z1;turbulentIntnst.z2]';
   %Reynolds stresses
    headRstress = {'ReynoldsStress uv','ReynoldsStress uw',...
        'ReynoldsStress vw'};
   %put Reynolds stress arrays in a matrix
    rstress =[ReynoldsStress.x; ReynoldsStress.y;ReynoldsStress.z1]';
 % %AvgSNR
% HeadavgSnr = {'AvgSnrX','AvgSnrY','AvgSnrZ1','AvgSnrZ2'};
% %put SNR arrays in a matrix
% avgSnr =
[cleanSnrMean.x;cleanSnrMean.y;cleanSnrMean.z1;cleanSnrMean.z2]';
% %avg correlation
% HeadavgCor = {'AvgCorX','AvgCorY','AvgCorZ1','AvgCorZ2'};
% %put correlation arrays in a matrix
% avgCor = [cleanCorrelationMean.x;cleanCorrelationMean.y;...
%           cleanCorrelationMean.z1;cleanCorrelationMean.z2]';
% %ave amplitude
% HeadavgAmp = {'AvgAmpX','AvgAmpY','AvgAmpZ1','AvgAmpZ2'};
% %put amplitude arrays in a matrix
% avgAmp = [cleanAmplitudeMean.x;cleanAmplitudeMean.y;...
%           cleanAmplitudeMean.z1;cleanAmplitudeMean.z2]';
   %totalHeader
%totalHeader = ...
%           ['depth (m)',headMeanVel,headVelIntnst,headRstress,...
%
HeadavgSnr,HeadavgCor,HeadavgAmp,'skewness.x','skewness.y',...
%           'skewness.z1','skewness.z2'];
    totalHeader = ['depth (m)',headMeanVel,headVelIntnst,headRstress];
%put all above matrices into a big matrice
%converting sorted depth into meters
   %skew = [skw.x;skw.y;skw.z1;skw.z2]';
%dataout = [depths_computed',meanVel,velIntnst,rstress,skew];
    dataout = [depths_computed',meanVel,velIntnst,rstress];
%
%write to a output file
    xlswrite(outputfile2,totalHeader);
    xlswrite(outputfile2,dataout,'Sheet1','A2');
 end



clear Data Config someString* s outputfile* namestring mess pressLength
clear pressSampFreq* good* pressTime



save postProcessedData;
```

```
[stat, mess, id] = rmdir('pressHistograms','s');
mkdir('pressHistograms');

resampPressLength = length(resampPressureFluct.three);
quadOutLength = length(quadOut(10).sweepIdx);
minIdx = min(resampPressLength, quadOutLength);

binNum = 17;

rPf3 = resampPressureFluct.three(101:minIdx-300);
quadEject = quadOut(binNum).ejectIdx(101:minIdx-300);
quadSweep = quadOut(binNum).sweepIdx(101:minIdx-300);
xxx=rPf3(quadEject);
fig1 = figure(1);
hist(xxx(1:end-100),60);
xlabel({'Ejection Events'});ylabel({'counts'});
title('Pressure Sensor 3');
xxx=rPf3(quadSweep);
fig2 = figure(2);
hist(xxx(1:end-100),60);
xlabel({'Sweep Events'});ylabel({'counts'});
title('Pressure Sensor 3');
fig3 = figure(3);
hist(rPf3,60);
xlabel({'Pressure Fluctuation (inches of water)'});ylabel({'counts'});
title('Pressure Sensor 3');

someString1 = ['SaveAllFigures(''pressHistograms\Psensor3_'
int2str(i)];
someString2 = ['''',''jpg'')'];
someString3 = [someString1 someString2];
eval(someString3);
close all;

rPf2 = resampPressureFluct.two(101:minIdx-300);
quadEject = quadOut(binNum).ejectIdx(101:minIdx-300);
quadSweep = quadOut(binNum).sweepIdx(101:minIdx-300);
xxx=rPf2(quadEject);
fig1 = figure(1);
hist(xxx(1:end-100),60);
xlabel({'Ejection Events'});ylabel({'counts'});
title('Pressure Sensor 2');
xxx=rPf2(quadSweep);
fig2 = figure(2);
hist(xxx(1:end-100),60);
xlabel({'Sweep Events'});ylabel({'counts'});
title('Pressure Sensor 2');
fig3 = figure(3);
hist(rPf2,60);
xlabel({'Pressure Fluctuation (inches of water)'});ylabel({'counts'});
title('Pressure Sensor 2');
```

```
someString1 = ['SaveAllFigures(''pressHistograms\Psensor2_'
int2str(i)];
someString2 = [''',''jpg'')'];
someString3 = [someString1 someString2];
eval(someString3);
close all;

rPf1 = resampPressureFluct.one(101:minIdx-300);
quadEject = quadOut(binNum).ejectIdx(101:minIdx-300);
quadSweep = quadOut(binNum).sweepIdx(101:minIdx-300);
xxx=rPf1(quadEject);
figure(7);
hist(xxx(1:end-100),60);
xlabel({'Ejection Events'});ylabel({'counts'});
title('Pressure Sensor 1');
xxx=rPf1(quadSweep);
figure(8);
hist(xxx(1:end-100),60);
xlabel({'Sweep Events'});ylabel({'counts'});
title('Pressure Sensor 1');
figure(9);
hist(rPf1,60);
xlabel({'Pressure Fluctuation (inches of water)'});ylabel({'counts'});
title('Pressure Sensor 1');

someString1 = ['SaveAllFigures(''pressHistograms\Psensor1_'
int2str(i)];
someString2 = [''',''jpg'')'];
someString3 = [someString1 someString2];
eval(someString3);
close all;

resampPressureFluct.intensity.three = ...
    std(resampPressureFluct.three(101:minIdx-300));
resampPressureFluct.intensity.two = ...
    std(resampPressureFluct.two(101:minIdx-300));
resampPressureFluct.intensity.one = ...
    std(resampPressureFluct.one(101:minIdx-300));
save postProcessedData;

stream = Filtered.Fluct.x(101:minIdx-300, binNum);
trans = Filtered.Fluct.y(101:minIdx-300, binNum);
vert1 = Filtered.Fluct.z1(101:minIdx-300, binNum);

moreProbabilityStuff
```

LoadWeights

```
%column 1 is a column 2 is b.
%These are the weights for the Nortek VectrinoII These weights were
%calculated in an Excel file and are based on cad drawings of the
```

```
%VectrinoII. They are of similar magnitude when compared to the weights
on
%Hurther and Lemmin 2001. These weights need to be changed if the
%resolution of the velocity bins is changed

weights = [
5.612611223 0.548898692;
5.791299631 0.547247372;
5.974287331 0.545668045;
6.161577734 0.544157302;
6.353174114 0.542711868;
6.549079613 0.541328601;
6.749297234 0.540004498;
6.953829848 0.538736689;
7.16268019  0.537522437;
7.375850867 0.536359136;
7.59334436  0.535244306;
7.815163027 0.534175588;
8.041309107 0.533150743;
8.271784728 0.532167643;
8.506591907 0.531224272;
8.745732558 0.530318713;
8.989208498 0.529449153;
9.237021446 0.52861387;
9.489173033 0.527811235;
9.745664806 0.5270397;
10.00649823 0.526297801;
10.27167469 0.525584151;
10.5411955  0.524897434;
10.81506191 0.524236403;
11.09327511 0.523599878;
11.3758362  0.522986738;
11.66274626 0.522395922;
11.95400628 0.521826424;
12.24961723 0.521277289;
12.54958001 0.520747611;
12.85389547 0.520236532
];
```

LoadPressureData

```
%get the name of the pressure files
pressFileNames = dir('press*.dat');
pressCounter = length(pressFileNames);
for i = 1:pressCounter
    nameString=pressFileNames(i,1).name;
    s=['readPressureData ' pressFileNames(i,1).name];
    eval(s);

    %clear,clc;
end
```

Pressure Resample

```
rbfQ=10; %reduce by factor used in resample function
rbfP=1;  %reduce by factor used in resample function

%create a folder in the present folder
currentFolder = pwd;
f1=fullfile(currentFolder,'matlab_output');
if (exist(f1) == 0)
    mkdir (f1);
end

resampledPressure1 = resample(pressure1,rbfP,rbfQ);  % Now resample
it...
%       from 1000Hz to 100Hz
resampledPressure2 = resample(pressure2,rbfP,rbfQ);  % Now resample it
resampledPressure3 = resample(pressure3,rbfP,rbfQ);  % Now resample it
resampPressure = struct('one',resampledPressure1, ...
    'two',resampledPressure2, 'three', resampledPressure3);
clear resampledPressure1 resampledPressure2 resampledPressure3

resampPressureFluct = struct('one',resampPressure.one -...
    mean(resampPressure.one),...
    'two',resampPressure.two - mean(resampPressure.two),...
    'three',resampPressure.three - mean(resampPressure.three));

pressLength = length(pressure1);
pressSampFreq = 1000;
pressTime =
1/pressSampFreq:1/pressSampFreq:pressLength*1/pressSampFreq;
pressTime2 = pressTime';

%resampPressLength = length(resampledPressure1);
%resampPressFreq = 1000/10;
resampPressTime = 1/pressSampFreq:(1/pressSampFreq)*...
    (rbfQ/rbfP):pressLength*1/pressSampFreq;
resampPressTime2 = resampPressTime';

figure('Name','Pressure Sensor 1','NumberTitle','off');
hold on
xlabel({'time (seconds)'});ylabel({'differential pressure head (in of
h2o)'});
plot(pressTime2,
pressure1,'*',resampPressTime2,resampPressure.one,'o');
title({'Raw & Resampled Pressure Data',' Sensor1'})
SaveAllFigures('matlab_output\p1Resample', 'jpg')
close all

figure('Name','Pressure Sensor 2','NumberTitle','off');
hold on
```

```
xlabel({'time (seconds)'});ylabel({'differential pressure head (in of
h2o)'});
plot(pressTime2,
pressure2,'*',resampPressTime2,resampPressure.two,'o');
title({'Raw & Resampled Pressure Data',' Sensor2'})
SaveAllFigures('matlab_output\p2Resample', 'jpg')
close all

figure('Name','Pressure Sensor 3','NumberTitle','off');
hold on
xlabel({'time (seconds)'});ylabel({'differential pressure head (in of
h2o)'});
plot(pressTime2,
pressure3,'*',resampPressTime2,resampPressure.three,'o');
title({'Raw & Resampled Pressure Data',' Sensor3'})
SaveAllFigures('matlab_output\p3Resample', 'jpg')
close all

fig3 = figure('Name','All Pressure Sensors','NumberTitle','off');
axes1 = axes('Parent',fig3,'YMinorTick','on',...
    'YMinorGrid','on',...
    'YGrid','on',...
    'XMinorTick','on',...
    'XMinorGrid','on',...
    'XGrid','on');

hold on
xlabel({'time (seconds)'});ylabel({'differential pressure head (in of
h2o)'});
plot(resampPressTime2,resampPressure.one,'+',...
    resampPressTime2,resampPressure.two,'*',...
    resampPressTime2,resampPressure.three,'o');
title({'Resampled Pressure Data',' All Sensors'})

legend(axes1,'show', 'pressure1', 'pressure2','pressure3',...
    'location', 'NorthWest');

SaveAllFigures('matlab_output\allResample', 'jpg')
close all
```

PressureSpectra

```
function output = pressureSpectra(p1, p2, p3)

L = length(p1);
Fs = 100; %sample frequency
NFFT = 2^nextpow2(L);

numValues = 2048;
valuesOverlap = 512;
```

```
%cpsd is cross power spectral density
%[cpsdPxx, cpsdW]=cpsd(streamVel, streamVel, [], [],NFFT, Fs);


[psdP11, cpsdW]=pwelch(p1,numValues,valuesOverlap,NFFT, Fs);
[psdP22, cpsdW]=pwelch(p2,numValues,valuesOverlap,NFFT, Fs);
[psdP33, cpsdW]=pwelch(p3,numValues,valuesOverlap,NFFT, Fs);

% [psdP11]=fft(p1,NFFT, Fs);
% [psdP22]=fft(p2,NFFT, Fs);
% [psdP33]=fft(p3,NFFT, Fs);
[cpsdP12, cpsdW]=cpsd(p1, p2,numValues,valuesOverlap,NFFT, Fs);
[cpsdP13, cpsdW]=cpsd(p1,p3,numValues,valuesOverlap,NFFT, Fs);
[cpsdP23, cpsdW]=cpsd(p2,p3,numValues,valuesOverlap,NFFT, Fs);



% someWindow = hamming(L);
%
% [cpsdPxx,cpsdW] = periodogram(streamVel,someWindow,NFFT);
% [cpsdPxy, cpsdW]=cpsd(streamVel, transVel, [], [],NFFT, Fs);

% %myNoise = 0.5.*(cpsdPxx1+cpsdxx2) - abs(cpsdPxy);
% myNoise = (0.5.*(psdPz1z1+psdPz2z2) - abs(cpsdPz1z2))./weights(i,2);
% horizNoise = weights(i,1).*myNoise;
%
% %Corrected Spectra
% correctedX = psdPxx - horizNoise;
% correctedY = psdPyy - horizNoise;

%CorrectedVelocity

% A = [min(cpsdW):max(cpsdW)/length(cpsdW):max(cpsdW)];

% sigSquared = trapz(cpsdW,myNoise);
% correctedStream = sqrt(streamVel.^2 + weights(i,1)*sigSquared);
% correctedTrans = sqrt(transVel.^2 + weights(i,1)*sigSquared);


% [p,S,mu] = polyfit(cpsdW, psdPxx,1);


output.freqs= cpsdW;
output.P11= psdP11;
output.P22= psdP22;
output.P33= psdP33;
output.P12= cpsdP12;
output.P13= cpsdP13;
output.P23= cpsdP23;
```

```
% output.sigSqrdOut = trapz(cpsdW, myNoise);
% output.noiseSpectra = myNoise;
% output.horizNoiseSpectra = horizNoise;


% output.p = p;
% output.S = S;
% output.mu = mu;

% output.corrected.Pxx = correctedX;
% output.corrected.Pyy = correctedY;
% output.corrected.sigSquared = sigSquared;
% output.corrected.stream = correctedStream;
% output.corrected.trans = correctedTrans;
%
% fittedY=exp(p(1).*cpsdW + p(2));
%
% kolmogorovX = [0.1:0.1:50];
% kolmogorovY = kolmogorovX.^(-5/3)*exp(-4);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
fig1 = figure('Name','four component spectra','NumberTitle','off');
% Create axes
axes1 = axes('Parent',fig1,'YScale','log','YMinorTick','on',...
    'YMinorGrid','on',...
    'YGrid','on',...
    'XScale','log',...
    'XMinorTick','on',...
    'XMinorGrid','on',...
    'XGrid','on');
box('on');
hold('all');
xlabel({'frequency (Hz)'});ylabel({'spectral units'});
%loglog(cpsdW,2*abs(spectX(1:NFFT/2))); %Plots power spectra
loglog(cpsdW,psdP11,cpsdW,psdP22,cpsdW,psdP33,cpsdW,cpsdP12, ...
    cpsdW,cpsdP13,cpsdW,cpsdP23);%, cpsdW,fittedY Plots power spectra

% set(loglog(1),'DisplayName','streamwise');
% set(loglog(2),'DisplayName','transverse');
% set(loglog(3),'DisplayName','z1');
% set(loglog(4),'DisplayName','z2');
% set(loglog(5),'DisplayName','noise');
% set(loglog(6),'DisplayName','kolmogorov');

title({'Spectral Analysis',' Velocity'});

legend(axes1,'show', 'P11', 'P22','P33', 'P12', 'P13',...
    'P23', 'location', 'NorthWest');
```

```
%j counts which file is being written (i.e. the profiles were shifted
up
%and up and up and have seperate files. i counts the bins within a file
someString1 = ['SaveAllFigures(''pressureSpectra\SpectLog' ...
    int2str(j) '_' int2str(i)];
someString2 = ['','jpg'')'];
someString3 = [someString1 someString2];
eval(someString3);
%SaveAllFigures('spectra\xSpectLog', 'jpg')
close all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
fig2 = figure('Name','four component spectra','NumberTitle','off');
% Create axes
axes2 = axes('Parent',fig2,'YScale','log','YMinorTick','on',...
    'YMinorGrid','on',...
    'YGrid','on',...
    'XScale','log',...
    'XMinorTick','on',...
    'XMinorGrid','on',...
    'XGrid','on');
box('on');
hold('all');
xlabel({'frequency (Hz)'});ylabel({'spectral units'});
%loglog(cpsdW,2*abs(spectX(1:NFFT/2))); %Plots power spectra
loglog(cpsdW,psdP11,cpsdW,psdP22,cpsdW,psdP33);

% set(loglog(1),'DisplayName','streamwise');
% set(loglog(2),'DisplayName','transverse');
% set(loglog(3),'DisplayName','z1');
% set(loglog(4),'DisplayName','z2');
% set(loglog(5),'DisplayName','noise');
% set(loglog(6),'DisplayName','kolmogorov');

title({'Spectral Analysis',' Velocity'});

legend(axes2,'show', 'P11', 'P22','P33', 'location', 'NorthWest');

%j counts which file is being written (i.e. the profiles were shifted
up
%and up and up and have seperate files. i counts the bins within a file
someString1 = ['SaveAllFigures(''pressureSpectra\partialSpectLog' ...
    int2str(j) '_' int2str(i)];
someString2 = ['','jpg'')'];
someString3 = [someString1 someString2];
eval(someString3);
%SaveAllFigures('spectra\xSpectLog', 'jpg')
close all;
```

Simple GaussianFilter

```
function [outX, outY, outZ1,outZ2, outIdx] = simpleGaussian3d(inputX,
inputY, inputZ1, inputZ2)

%Performs simpe Gaussian filtering for four related variables. It was
%written to filter velocities from a Nortek Vectrino.
%Written by Tim Calappi 19 November 2011
%tcalappi@gmail.com

%Determine mean velocity
meanVel.x = mean(inputX);
meanVel.y = mean(inputY);
meanVel.z1 = mean(inputZ1);
meanVel.z2 = mean(inputZ2);

%Determine fluctuations
fluct.x = inputX - meanVel.x;
fluct.y = inputY - meanVel.y;
fluct.z1 = inputZ1 - meanVel.z1;
fluct.z2 = inputZ2 - meanVel.z2;


%Max acceptable velocity is the mean plus 3 times the standard
deviation
%of the fluctuations
accept.max.x = meanVel.x + 3*std(fluct.x);
accept.min.x = meanVel.x - 3*std(fluct.x);

accept.max.y = meanVel.y + 3*std(fluct.y);
accept.min.y = meanVel.y - 3*std(fluct.y);

accept.max.z1 = meanVel.z1 + 3*std(fluct.z1);
accept.min.z1 = meanVel.z1 - 3*std(fluct.z1);

accept.max.z2 = meanVel.z2 + 3*std(fluct.z2);
accept.min.z2 = meanVel.z2 - 3*std(fluct.z2);

%good indices are between the acceptable limits
good.x = inputX < accept.max.x & inputX > accept.min.x;
good.y = inputY < accept.max.y & inputY > accept.min.y;
good.z1 = inputZ1 < accept.max.z1 & inputZ1 > accept.min.z1;
good.z2 = inputZ2 < accept.max.z2 & inputZ2 > accept.min.z2;


%if I'm trhowing away a particular x velocity, I need to throw out the
y
%and z velocities of the corresponding index. So the good indices are
at
%the intersection of the good indices for each velocity component.
goodIdx = good.x & good.y & good.z1 & good.z2;

%filtered velocity is exactly equal to the input velocity but wherever
```

```
%goodIdx is equal to zero (i.e. ~goodIdx) replace those instantaneous
%velocities with NaN

tempX = inputX;
tempX(~goodIdx) = NaN;

tempY = inputY;
tempY(~goodIdx) = NaN;

tempZ1 = inputZ1;
tempZ1(~goodIdx) = NaN;

tempZ2 = inputZ2;
tempZ2(~goodIdx) = NaN;



%crap variables used for input to interpolation only
crapX1 = find(~isnan(tempX));
crapY1 = tempX(crapX1);
x1  = 1:max(length(tempX));
outX = interp1(crapX1, crapY1, x1, 'cubic')';

crapX2 = find(~isnan(tempY));
crapY2 = tempY(crapX2);
x2  = 1:max(length(tempY));
outY = interp1(crapX2, crapY2, x2, 'cubic')';

crapX3 = find(~isnan(tempZ1));
crapY3 = tempZ1(crapX3);
x3  = 1:max(length(tempZ1));
outZ1 = interp1(crapX3, crapY3, x3, 'cubic')';

crapX4 = find(~isnan(tempZ2));
crapY4 = tempZ2(crapX4);
x4  = 1:max(length(tempZ2));
outZ2 = interp1(crapX4, crapY4, x4, 'cubic')';

% increment = [1:1:length(inputX)];
% plot(increment,outZ1, increment,outZ2)

outIdx = goodIdx;
```

Beam Velocity

```
function beamFinalOut = beam(x, y, z1, z2, calMatrix)

noSamples = size(x,1); noBins = size(x,2);

for j = 1:noBins
```

```
    for i = 1:noSamples
        T = reshape(calMatrix(j, : ), 4, 4 )'; %use j index

        %bv stands for beam velocity
        bv =  inv( T ) * [ x(i,j); y(i,j); z1(i,j); z2(i,j) ];

        beamOut.one(i,j) = bv(1);
        beamOut.two(i,j) = bv(2);
        beamOut.three(i,j) = bv(3);
        beamOut.four(i,j) = bv(4);
    end
end

beamFinalOut = beamOut;
```

More Probability Stuff

```
close all

figure(1);
probplot(rPf1ND);
title('Normal Probability Plot for Pressure Sensor 1');
xlabel({'pressure fluctuations (inches of water)'});
figure(2);
qqplot(rPf1ND,stream);
title('QQ Plot for Pressure Sensor 1');
xlabel({'Pressure Quantiles'});ylabel({'Streamwise Velocity
Quantiles'});
figure(3);
qqplot(rPfND1,trans);
title('QQ Plot for Pressure Sensor 1');
xlabel({'Pressure Quantiles'});ylabel({'Transverse Velocity
Quantiles'});
figure(4);
qqplot(rPf1ND,vert1);
title('QQ Plot for Pressure Sensor 1');
xlabel({'Pressure Quantiles'});ylabel({'Vertical Velocity Quantiles'});
someString1 = ['SaveAllFigures(''pressHistograms\distributionPlotP1_'
int2str(i)];
someString2 = ['''',''jpg'')'];
someString3 = [someString1 someString2];
eval(someString3);
close all;

figure(5);
probplot(rPf2ND);
title('Normal Probability Plot for Pressure Sensor 2');
xlabel({'pressure fluctuations (inches of water)'});
figure(6);
qqplot(rPf2ND,stream);
```

```
title('QQ Plot for Pressure Sensor 2');
xlabel({'Pressure Quantiles'});ylabel({'Streamwise Velocity
Quantiles'});
figure(7);
qqplot(rPf2ND,trans);
title('QQ Plot for Pressure Sensor 2');
xlabel({'Pressure Quantiles'});ylabel({'Transverse Velocity
Quantiles'});
figure(8);
qqplot(rPf2ND,vert1);
title('QQ Plot for Pressure Sensor 2');
xlabel({'Pressure Quantiles'});ylabel({'Vertical Velocity Quantiles'});
someString1 = ['SaveAllFigures(''pressHistograms\distributionPlotP2_'
int2str(i)];
someString2 = [''',''jpg'')'];
someString3 = [someString1 someString2];
eval(someString3);
close all;

figure(9);
probplot(rPf3ND);
title('Normal Probability Plot for Pressure Sensor 3');
xlabel({'pressure fluctuations (inches of water)'});
figure(10);
qqplot(rPf3ND,stream);
title('QQ Plot for Pressure Sensor 3');
xlabel({'Pressure Quantiles'});ylabel({'Streamwise Velocity
Quantiles'});
figure(11);
qqplot(rPf3ND,trans);
title('QQ Plot for Pressure Sensor 3');
xlabel({'Pressure Quantiles'}
);ylabel({'Transverse Velocity Quantiles'});
figure(12);
qqplot(rPf3ND,vert1);
title('QQ Plot for Pressure Sensor 3');
xlabel({'Pressure Quantiles'});ylabel({'Vertical Velocity Quantiles'});
someString1 = ['SaveAllFigures(''pressHistograms\distributionPlotP3_'
int2str(i)];
someString2 = [''',''jpg'')'];
someString3 = [someString1 someString2];
eval(someString3);
close all;

figure(13);
hist(stream,40);
xlabel({'Streamwise Velocity Fluctuation'});ylabel({'counts'});

figure(14);
hist(trans,40);
xlabel({'Transverse Velocity Fluctuation'});ylabel({'counts'});

figure(15);
```

```
hist(vert1,40);
xlabel({'Vertical Velocity Fluctuation'});ylabel({'counts'});

someString1 =
['SaveAllFigures(''pressHistograms\velocityDistributions_' int2str(i)];
someString2 = [''',''jpg'')'];
someString3 = [someString1 someString2];
eval(someString3);
close all;
```

REFERENCES

Bean, H. S. (1971). "Fluid meters: their theory and applications." American Society of Mechanical Engineers.

Breugem, W. P., *et al.* (2006). "The influence of wall permeability on turbulent channel flow." *Journal of Fluid Mechanics*, 562(Compendex), 35-72.

Davidson, P. A. (2009). *Turbulence An introduction for scientists and engineers*, Oxford, New York.

Detert, M., *et al.* (2010a). "Synoptic velocity and pressure fields at the water-sediment interface of streambeds." *Journal of Fluid Mechanics*, Not yet available.

Detert, M., and Parker, G. (2010). "Estimation of washout depth of fine sediments from a granular bed." *Journal of Hydraulic Engineering*, 136(10).

Detert, M., *et al.* (2010b). "Laboratory Measurements on turbulent pressure fluctuations in and above gravel beds." *Journal of Hydraulic Engineering*, 136(10).

Detert, M., *et al.* "Pressure fluctuations within subsurface gravel bed caused by turbulent open-channel flow." *Proc., River Flow*, 695-701.

Dey, S., and Papanicolaou, A. (2008). "Sediment Threshold under Stream Flow: A State-of-the-Art Review." *KSCE J. Civ. Eng.*, 12(1), 45-60.

Einstein, H. A., and El-Samni, E.-S. A. (1949). "Hydrodynamic Forces on a Rough Wall." *Reviews of Modern Physics*, 21(Copyright (C) 2010 The American Physical Society), 520.

Higashino, M., and Stefan, H. (2008). "Velocity pulse model for turbulent diffusion from flowing water into a sediment bed." *J. Environ. Eng.-ASCE*, 134(7), 550-560.

Higashino, M., *et al.* (2009). "Pore water flow due to near-bed turbulence and associated solute transfer in a stream or lake sediment bed." *Water Resources Research*, 45.

Hodi, B. (2009). "The effect of blockage and densimetric froude number on circular bridge pier local scour." Master of Science Master's, University of Windsor, Windsor.

Hofland, B.*, et al.* (2005). "Measurement of fluctuating pressures on coarse bed material." *J. Hydraul. Eng.-ASCE*, 131(9), 770-781.

Hofland, B., and Battjes, J. A. (2006). "Probability density function of instantaneous drag forces and shear stresses on a bed." *J. Hydraul. Eng.-ASCE*, 132(11), 1169-1175.

Keshavarzy, A., and Ball, J. E. (1997). "Analysis of the characteristics of rough bed turbulent shear stresses in an open channel." *Stoch. Hydrol. Hydraul.*, 11(3), 193-210.

Khezri, N., and Chanson, H. (2012). "Inception of bed load motion beneath a bore." *Geomorphology*, 153–154(0), 39-47.

Kirkil, G.*, et al.* (2008). "Coherent Structures in the Flow Field around a Circular Cylinder with Scour Hole." *Journal of hydraulic engineering (New York, N.Y.)*, 134(5), 572-587.

Kline, S. J.*, et al.* (1967). "Structure of turbulent boundary layers." *Journal of Fluid Mechanics*, 30(4), 33.

Krogstad, P. A.*, et al.* (1992). "Comparison between rough-wall and smooth-wall turbulent boundary layers." *Journal of Fluid Mechanics*, 245, 599-617.

Manes, C.*, et al.* (2011). "Turbulent boundary layers over permeable walls: scaling and near-wall structure." *Journal of Fluid Mechanics*, FirstView, 1-30.

McClerren, M. (2009). "Assessment of the Correlations between erodibility and geothenical characteristics of fine grained fluvial soils from lower Michigan." Master of Science, Lawrence Technological Ubniversity, Southfield.

Munson, B.*, et al.* (2006). *Fundamentals of Fluid Mechanics*, Wiley, New York.

Nassif, H.*, et al.* (2002). "Evaluation of bridge scour monitoring methods." FHWA, ed.

Nelson, J. M., *et al.* (1995). "Role of turbulence structure in bed-load transport and bed form mechanics." *Water Resources Research*, 31(8), 2071-2086.

Ruff, J. F., and Gelhar, L. W. (1972). "Turbulent shear flow in porous boundary." *Journal of Engineering Mechanics*, 44, 1-25.

Schmeeckle, M. W., and Nelson, J. M. (2003). "Direct numerical simulation of bedload transport using a local, dynamic boundary condition." *Sedimentology*, 50(2), 279-301.

Schmeeckle, M. W., *et al.* (2007). "Forces on stationary particles in near-bed turbulent flows." *J. Geophys. Res.-Earth Surf.*, 112(F2).

Smart, G. M., and Habersack, H. M. (2007). "Pressure fluctuations and gravel entrainment in rivers." *Journal of Hydraulic Research*, 45(5), 661-673.

Sumer, M., *et al.* (2003). "Influence of turbulence on bed load sediment transport." *Journal of Hydraulic Engineering*, 129(8).

Thomas, A. S. W., and Bull, M. K. (1983). "On the role of wall-pressure fluctuations in deterministic motions in the turbulent boundary layer." *Journal of Fluid Mechanics*, 128(-1), 283-322.

Vollmer, S., *et al.* (2002). "Micro scale exchange processes between surface and subsurface water." 269, 3-10.

Vollmer, S., and Kleinhans, M. G. (2007). "Predicting incipient motion, including the effect of turbulent pressure fluctuations in the bed." *Water Resources Research*, 43(5).

Zagini, A. F. E., and Smith, K. V. H. (1976). "Channel flow over permeable beds of graded spheres." *Journal of Hydraul.*(102), 207-222.

Zanke, U. C. E. (2003). "On the influence of turbulence on the initiation of sediment motion." *International Journal of Sediment Research*, 18(1), 17-17.

ABSTRACT

INVESTIGATION OF PRESSURE FLUCTUATIONS IN THE HYPORHEIC ZONE IN RESPONSE TO FLOW AROUND A HYDRAULIC STRUCTURE

by

TIMOTHY J. CALAPPI

December 2013

Advisor: Dr. Carol Miller

Major: Civil Engineering

Degree: Doctor of Philosophy

Erosion around a cylinder is a well studied field. Particles erode when lift and drag forces overcome a critical threshold. These forces are typically studied from above the water-riverbed interface. This study maps hyporheic pressure fluctuations as they are related to surface water velocity. The pressure map is used to evaluate lift enhancement and destabilization forces on the riverbed. High pressure events in the subsurface help generate a destabilizing force from within the riverbed. This work develops a probability distribution function relating turbulent velocity fluctuations and subsurface pressure fluctuations.

A cylinder was fitted with differential pressure transducers such that the pressure ports were flush with the cylinder surface and below the water-sand interface. Three-component velocities were recorded synchronously with differential pressure fluctuations measured over a 15 mm depth. As expected, results show decay in pressure fluctuations as a function of depth. The standard deviation of the pressure fluctuation in the upper hyporheic zone scales well with shear stress.

AUTOBIOGRAPHICAL STATEMENT

I am currently a hydraulic engineer at the US Army Corps of Engineers. I specialize in Great Lakes connecting channels issues. Work I find most interesting lies at the intersection of hydroacoustics and sediment transport. I'm an Aquarius.