A method for measuring time-resolved, path-integrated temperature in a reciprocating internal combustion engine cylinder using ultrasonic thermometry

A THESIS SUBMITTED TO THE FACULTY OF
THE UNIVERSITY OF MINNESOTA
BY

C. T. Weigelt

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

Dr. William Northrop

September 2016

© Chad T. Weigelt 2016

Acknowledgments

+JMJ

First, except by the grace of God, I would not have been able to accomplish this degree. Second, the patience and encouragement of my wife is without measure. She allowed me to pursue graduate school and throughout has been supportive in taking on the extra burden caused by continuing school at the same time as starting our life together. My extended family has been integral as well, namely my mother-in-law for providing residence while I was a full-time graduate student.

I am greatly indebted to Dr. Will Northrop for his accepting me as an research assistant in his lab. His passion and excitement for research is evident in the way he treats his graduate students. He is constantly positive in the face of unsure results and provides the energy needed to finish the sometimes daunting, task-at-hand. Will also has an abundance of patience which has never been clearer than in the situation of my writing this thesis.

Thanks to all of my fellow graduate students in the Center for Diesel Research Lab for their support and camaraderie. Thank you to Darrick Zarling and John Adams for their help in the lab and taking down and re-setting up my test stand, twice. Thank you to Erik Anderson, at Emerson, for your sage advice on how to finish a master's thesis. Finally, I am grateful toward all of the students and professors I have interacted with and learned from throughout my collegiate career.

Abstract

A limitation currently facing internal combustion engines research is the lack of a direct method of measuring the temperature of the gases inside the cylinder. The rate at which a combustion cycle evolves is too rapid for conventional, direct measurement systems such as thermocouples. Other, fast measurement systems like laser-induced fluorescence (LIF) rely too heavily on engine modifications to be effective and cannot withstand typical engine operating conditions. The direct measurement of the gas temperature in an engine cylinder, at realistic operating conditions is needed to better understand the combustion cycle and its effects on cycle efficiency.

The study presented here measures the time of flight of an ultrasonic signal to calculate the temperature of the gas in an engine cylinder. The speed of sound is a function of the temperature of the medium through which it propagates as well as thermodynamic properties for gases. A test setup utilizing a modified engine head with a single transducer-receiver and temperature controlled intake was used to trigger ultrasonic signals at controlled crank angles in a motoring small engine. The signals then traveled through the engine cylinder, echoed off the piston head and returned to the transducer. The time of flight was measured using a Kalman filtering technique. Temperatures could then be calculated from the speed of sound using the time of flight measurement and known path length.

Results show that the method proposed here is capable of measuring in-cylinder temperatures consistently within an accuracy of 10% during engine motoring. The methods used here yield within sample deviations well below 0.25% when a significant number of measurements are available to analyze. Further improvements of the method are recommended to continue this proof of concept testing into a combusting engine environment.

Contents

List of Tables

List of Figures

1 Introduction and Background

1.1 Introduction

The temperature within a reciprocating, firing engine cylinder is not trivial to measure given the short time duration over which a single cycle evolves. To measure temperature in this environment requires a measurement technique that meets a minimum sampling rate akin to the Nyquist theorem for signal processing. An engine running at a speed of 2000 rpm means that a single cycle evolves over 30 ms and each crank angle degree (CAD) is passed in 83 μs. Typically, internal combustion engine research utilizes pressure measurements collected throughout the combustion cycle, in conjunction with the ideal gas law assuming initial conditions at the beginning of the closed cycle to calculate bulk gas temperature [6]. Direct temperature measurement is difficult. Ultrasonic thermometry is a viable method of direct temperature measurement within a reciprocating internal combustion engine cylinder.

Measurement techniques such as thermocouples, thermistors, fluid expansion thermometers and bimetallic devices suffer from lag in their measurement meaning they cannot be used effectively in an engine cylinder environment [7]. Another shortcoming of these methods is their invasive nature for measurement[6]. They require direct thermal contact with the fluid medium and measure only at the point where they are placed. Engine cylinders of course can have large thermal gradients meaning local temperature measurements can be misleading.

Spatially resolved methods such as radiation thermometry [8], laser-induced fluorescence (LIF) and Raleigh scattering have been shown to yield accurate results in gases. Like most optical techniques, these methods require a line-of-sight or highly modified engine cylinder. Optical methods have led to significant discoveries in combustion and fluid flow [9], but they are not ideal because the modifications necessary to conduct these studies do not allow for typical engine operating conditions.

Typical applications of ultrasonics in an air medium are object sensing in robotics and automotive industries. The use of the speed of sound as a pyrometer in gases was suggested in a publication as early as 1873 [10]. This method though was only first meaningfully applied in the 1930's [11, 12] when its applicability to high temperature applications was realized. Advancements in signal processing and the use of microprocessors have allowed for ultrasonic measurement techniques to become widespread in industry and research alike and are reviewed comprehensively in Lynnworth's work [13].

The goal of this work is to show, through proof of concept testing, that ultrasonic thermometry is a practical method for measuring temperature in a reciprocating engine. The method is capable of measurement on timescales adequate to reach single CAD resolution and can produce accurate measurements within 1% of the measured intake temperature while motoring at relevant engine operating speeds.

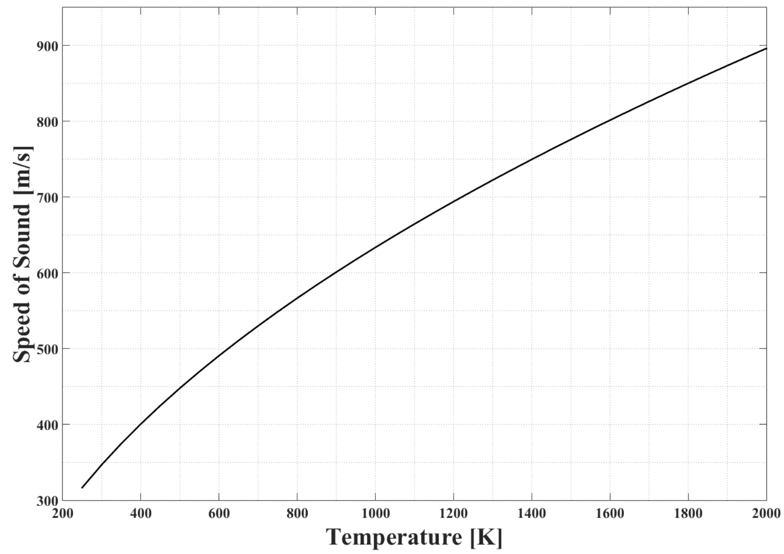


Figure 1.1: *Relationship between the speed of sound and the temperature of the medium where* $R = 8314\left[\frac{J}{kmol-K}\right]$, $\gamma = 1.4$, *and* $M = 29\left[\frac{kg}{kmol}\right]$.

1.2 Background

Ultrasonic thermometry is the measurement of temperature in a medium using the speed of sound's dependence on the temperature of the medium. The speed of sound in air is dependent on temperature as show in equation 1.1 where c is the speed of sound, gamma (γ) is the ratio of specific heats, R is the ideal gas constant, T is the temperature of the gas in absolute units and M is the average molecular weight of the gas [13]. As can be seen, the speed of sound has a square root relationship with temperature. Gamma (γ) can be estimated using known correlations with temperature whereas the other parameters, R and M do not change with temperature.

$$c = \sqrt{\frac{\gamma RT}{M}} \tag{1.1}$$

Assuming values of gamma (γ) and M of 1.4 and 29 kg/kmol respectively, yields a temperature speed of sound relationship as shown in Figure 1.1. When predicting the temperature using the speed of sound, an elementary rate multiplied by time equates to distance calculation is needed. The calculation shown in equation 1.2 is solved for the velocity. Once the speed of the ultrasonic signal is known, equation 1.1 is simply solved for temperature. The flight distance (l) is calculated from the crank angle (CA) data and engine parameters where time of flight (ToF) (τ) is experimentally measured. The flight distance (l), for the experiments presented here, is the path length over which the signal progresses as illustrated in Figure 1.2. The factor of two is necessary because the sent signal propagates through the fluid, reflects off the piston top and returns to the sending location.

$$c = \frac{2l}{\tau_{ToF}} \tag{1.2}$$

As discussed in Section 1.1 this method of temperature measurement is not limited to measuring a single point of the gas medium. The measurement recorded is ultimately a 1D time and space averaged reading of the variation throughout its path. An expression for this is given in equation 1.3. The temperature function is solved for using equation 1.1 where the speed of sound (c) and parameters gamma (γ), M and R are local to each measurement point. For the readings in this work, these parameters are taken as bulk fluid parameters because local measurement is not possible, thus resulting in an average temperature along the signal path.

$$T_{path} = \frac{1}{2l} \cdot \frac{1}{\tau_{ToF}} \int \int T(x,t)\, dx dt \tag{1.3}$$

George Stokes theorized the attenuation of sound in the middle of the 19th century [14].

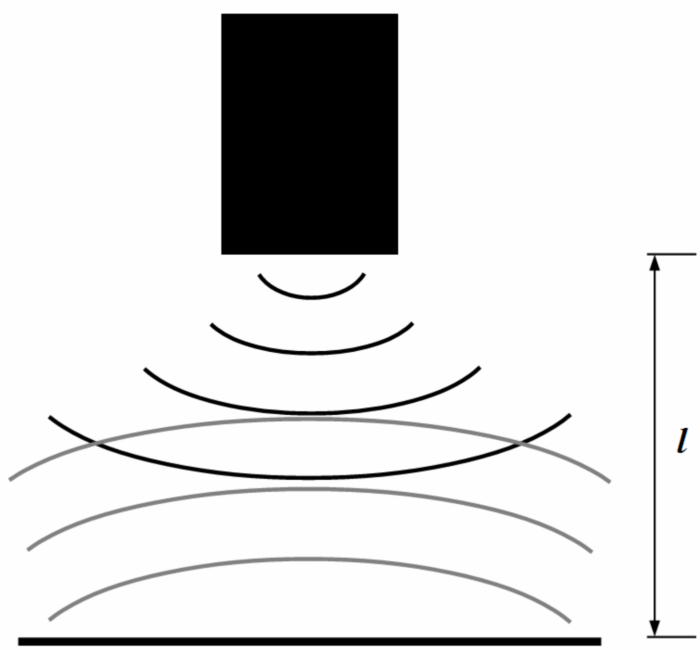


Figure 1.2: *The flight path* (l) *is twice the distance from the transducer to the reflecting surface.*

The derived equation for sound attenuation is given in equation 1.4 where η is the dynamic viscosity coefficient of the fluid, ω is the frequency of the sound, ρ is the fluid density and V is the speed of sound in the medium. From this classical theory of attenuation it can be seen that attenuation of the signal increases with the square of the frequency of the signal. Because of this physical degradation of the signal, long range sensing, in air, is limited by sound attenuation [15]. Clearly though this is not the only dependency relevant to the study here. Additionally, Stokes' absorption applies for dry air.

$$\alpha = \frac{2\eta\omega^2}{3\rho V^3} \tag{1.4}$$

In his review of ultrasonic transducers [1], Massa discusses the dependency of sound absorption on relative humidity as it was investigated by Knudsen [16] and Sivian [17]. Figure 1.3 is from Massa's review and shows the relationship between Stokes' classical absorption (curve B) and the empirical work of Sivian (curve A) at high frequencies at 37% relative humidity. At lower frequencies there is a more significant difference in absorption between the dry air and wet air. When the frequency increases to approximately 200 kHz the effects of humidity are decreased significantly.

The data presented in Figure 1.3 represents the effects relative humidity on sound absorption. Changing temperature is another factor that affects sound attenuation as it is related to the speed of sound. Cyril Harris investigated the effects of temperature on sound absorption, however these results only range up to 12.5 kHz making application to 200 kHz experiments performed here, difficult [18]. In general, increasing temperature increases absorption until a maximum is reached at which point the magnitude of the absorption decreases until it reaches a steady value. Not surprising, and in keeping with the results

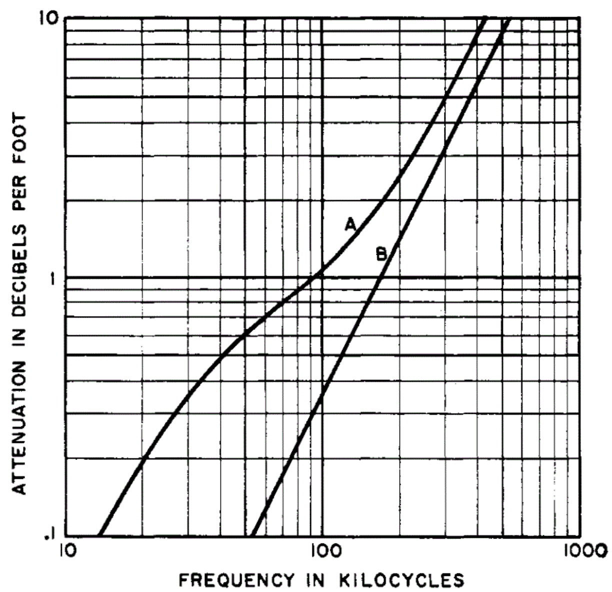


Figure 1.3: *Figure 5 from Massa [1] comparing classical absorption for dry air according to Stokes, curve B, and average experimental data for average atmospheric conditions of 75°F and 37 percent relative humidity, curve A.*

above, increasing the frequency of the signal increases both the maximum and steady values of attenuation.

Overall, Stokes' classical theory serves well as a indicator of the relationship between sound attenuation and frequency and temperature. These relationships are important to keep in mind when evaluating the data presented in this report but more important for future optimization of the method. For the experiments performed here, a reasonable expectation is that the signal returns with enough power to detect.

A final consideration when investigating thermometry is the effects of temperature gradients on the propagation of the sound wave. Temperature gradients will cause refraction of the sound wave in the direction of the temperature gradient [19]. This effect causes the primary intensity of the sound to travel in a non-linear direction, effectively turning the sound wave. When this happens in an engine cylinder, where a linear flight path is expected, the signal either does not return to the transducer or is very faint. Again, this effect will be more significant to combusting engine cylinder testing but is relevant to the testing conducted here, as well.

1.3 Application to internal combustion engines

As stated in section 1.1 there is not currently an effective or direct way to measure temperature in internal combustion engines. Understanding the combustion process is fundamental to internal combustion research and the formation of exhaust pollutants like soot and NOx [20]. Emitted species like carbon dioxide from the combustion process are unavoidable as they are a natural product of the combustion cycle. Emissions such as soot and NOx however can be mitigated through detailed understanding of the combustion cycle [21]. Current understanding of the combustion cycle is that the temperature at which ignition occurs are integral to the mitigation of soot and NOx production [22]. For example, in combustion, the Zel'dovich chemical mechanism shows that the conversion of N_2 and O_2 to NO and NO_2 is strongly dependent on temperature [23].

Although the method developed in this project results in a line-integrated temperature, it may not be an accurate measure of bulk in-cylinder temperature when combustion is occurring due to high spatial temperature gradients. However, one realistic application of the method could be to measure cylinder temperature just after the engine intake valve closes. At this point, the gas in the cylinder is well mixed, not reacting and has minimal thermal gradients. The initial temperature is commonly estimated to calculate the amount of exhaust gases trapped in the cylinder, referred to as the residual gas fraction [24]. This residual amount of exhaust gas has an effect on the subsequent combustion cycle similar to that created by exhaust gas recirculation (EGR). Currently the residual exhaust gases are estimated through pressure and volume analysis and the concentration of CO_2 in a sample of the gases during compression [20]. A method for directly measuring the temperature of the cylinder after the intake valve closes would greatly aid the estimation of residual exhaust gases. Exhaust gas composition can be accurately estimated by the bulk temperature of the gases because pressure and volume in the cylinder are known as is the temperature of the intake gases. Any change in temperature therefore is due to residual exhaust gases being present.

The ultrasonic thermometry technique could also be used to measure the temperature of intake and exhaust gases entering and exiting the cylinder. These measurements could be used by engine researchers to better understand the breathing process and determine heat transferred with the cylinder walls and valves. Determination of signal attenuation could also be used to measure in-cylinder turbulence effects during the breathing process.

In addition to more accurate characterization of residual gas fraction, and the gas exchange process, this method could potentially be applied to measuring in-cylinder temperatures to help validate combustion models and further advance the understanding of particle and pollutant formation. Increased sophistication in signal processing for this type of technique may also enable the ability to detect thermal gradients in the combustion chamber.

1.4 Prior research

A number of patents that have been filed dealing with the measurement of temperature in an internal combustion engine [25, 26, 4, 27, 28]. One of these patents by Allmendinger and his colleagues uses ultrasonic thermometry as the measurement method. The remainder refer to different forms of the direct temperature measurements discussed in Section 1.1.

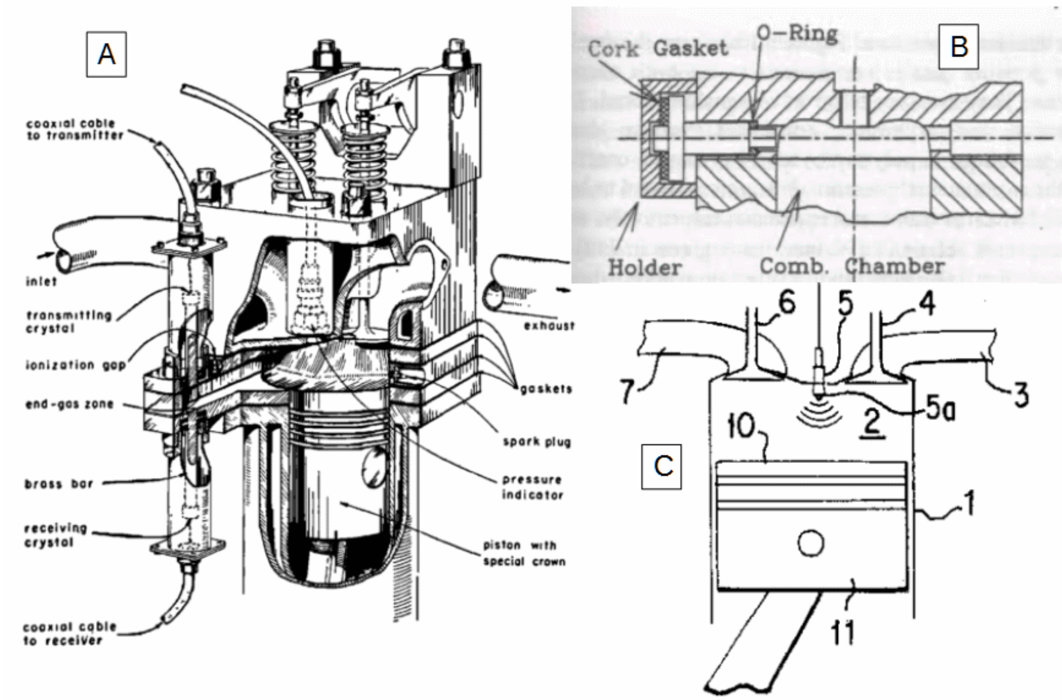


Figure 1.4: *Test setup diagrams from prior research conducted by Livengood et al. [2] (A), Bauer et al. [3] (B) and Allmendinger et al. [4] (C). Livengood uses an end-gas side chamber to measure gas temperatures ultrasonically. Bauer makes his measurement across the top of the engine cylinder. Allmendinger imagines a path from engine head to the top of the piston however this method was not reduced to practice.*

Allmendinger's method is most similar to the method presented in this study. They used a single transmitter-receiver transducer combination (transceiver). Ultrasound is emitted from the transceiver that is co-located with the spark-plug and bounced off the piston top and is read using the same emitting transceiver. A significant difference between this work and the study described here is that Allmendinger imagines using the sensor to continuously monitor the pressure and overlaying the ultrasonic signal at specific points of interest in the engine cycle. Through an exhaustive patent and literature search it was not found that this invention was never reduced to practice. At the time of publication of this work, the status of the patent maintenance fees are lapsed.

Related ultrasonic thermometry methods applied in internal combustion engine research use separate transmitting and receiving transducers. Livengood, Rona and Baruch utilize a separate side chamber to explore the composition and temperature of 'end-gas' or the region that is last to be consumed by the flame front [2]. In this work the researchers were able to achieve a repeatability of ±20 ºF or ±1% using mechanical triggering methods.

Bauer *et al.* have completed the most directly applicable study of internal combustion engine, in-cylinder thermometry to date [3]. They too use independent sending and receiving transducers, however their measurement path is directly across the top of the engine cylinder diameter. The time of flight is measured using the time lapse between the completion of the sent signal and peak of the received signal. This work shows that the method of measuring in cylinder temperatures with ToF measurements can lead to reasonable results. The main source of error cited by the authors is the ToF measurement.

In addition to measuring gas temperatures in piston-cylinder engines, the ultrasonic thermometry method has found other applications in internal combustion engines. For example Mauermann [29] and Hohenberg [30] have applied the method to rotary engines. Hohenberg used a fixed transmitter and receiver as well and uses the measurement to derive correction factors for pressure and gas temperatures. Exhaust gas temperatures are another application of ultrasonic thermometry. Lakshminarayanan and colleagues use an ultrasonic flow meter to monitor flow and temperature pulsations in exhaust [31] whereas Higashino *et al.* measured engine cycle resolved temperatures in exhaust using ultrasonic thermometry [32]. Dadd measured gas temperatures ranging from 300K to 1000K in sterling engines [33].

1.5 Objective of study

The objective of the study presented here is to explore the potential for accurate temperature measurements in a reciprocating internal combustion engine using ultrasonic ToF measurements. The method presented here uses a single transducer to both transmit and receive the signal. The flight path for the ultrasonic signal is from the engine head to the piston and then reflected back. This flight results in a path integrated temperature of the entire volume of the cylinder. The engine is modified such that pressure remains approximately constant and no combustion is occurring. The work here is intended to be proof-of-concept testing upon which further development and testing can be based. This testing focused largely on the triggering and data processing. Due to the scarcity of ultrasonic transducers capable of withstanding temperatures and pressures achieved under normal engine operating conditions, this study considered only engine motoring (i.e., no combustion) with no compression. However, the method developed here could be easily applied to a firing engine if suitable transducers became available.

2 Methods and Materials

2.1 Experimental setup

Testing was conducted on a single-cylinder Briggs & Stratton engine with a modified head to accommodate a single ultrasonic transducer. The engine was motored using a dynamometer to control engine speed. Figure 2.2 shows a diagram of the test setup illustrating the main components. A National Instruments (NI) PXI system was used as the control software for signal generation and data collection. A LabVIEW VI was developed in conjunction with the NI equipment to trigger signal generation and data collection. The program also monitored and recorded piston location and intake temperature.

2.1.1 Equipment

Table 2.1 lists the major components manufacturing and model information for the test setup. The engine, dynamometer, variable frequency drive (VFD), temperature controller and shaft encoder were located in the engine research lab at the Mechanical Engineering Department at the University of Minnesota. The ultrasonic transducer and amplifying circuit were purchased from Airmar Technology Corporation. The majority of the National

Table 2.1: *List of equipment and manufacturers used for this study.*

Equipment	Manufacturer	Model
Engine	Briggs & Stratton	1650 Series Model: 215232
Chassis	National Instruments	PXI-1042
Controller	National Instruments	PXI-8115
RIO	National Instruments	PXI-7813R
Digitizer	National Instruments	PXI-5124
AWG	National Instruments	PXI-5412
Expansion chassis	National Instruments	9151
Differential digital input	National Instruments	9411
Thermocouple input	National Instruments	9211
Ultrasonic transducer	Airmar Technology Corporation	AT200
Amplifying circuit	Airmar Technology Corporation	T1 Development Kit
Shaft encoder	BEI Sensors	HS35
Dynamometer	Midwest Dynamometer & Engineering Company	46VHTS
Variable frequency drive	MagneTek	GPD 503
Temperature controller	Autonics	TC3YT

Instruments equipment was provided through the National Instruments Academic Donations Program.

The thermocouples were Type K and were verified using an ice water bath. The engine speeds were verified using a handheld duemo tachometer.

2.1.2 Temperature control of intake

A heated intake was used to adjust and control the temperature in the engine cylinder. The intake heater consisted of a series of resistive heaters switched by a temperature controller to maintain a desired temperature. There was a thermocouple located at the bottom of the intake heater used as feedback to the temperature controller. To maintain temperature in the engine cylinder, the heater was set at a temperature above the desired cylinder temperature. The temperature at which the heater was set was experimentally determined to maintain the intake temperature measured by a separate thermocouple located in the engine head near the intake valve. The temperature inside the cylinder was assumed to be that which was measured at the intake thermocouple which was a volumetric mixture of the heated volume inside the heater and any extra volume required to fill the cylinder. Intake and exhaust were controlled with check valves on either end of the engine head. In this way, heated air could be inducted into the engine and expelled with the movement of the piston. The cracking pressure of the check valves were 1 psi, such that pressure did not build up in the cylinder, compromising the temperature measurement accuracy.

2.1.3 Engine modifications

A modified engine head was required to accommodate the ultrasonic transducer. Because the performed experiments only required motoring the engine with no compression, no spark plug or traditional intake and exhaust valves were needed. The engine head used in the study is shown in Figure 2.1 and a detailed drawing of the head is included in Appendix B. Compression was also negated through the use of simple check valves on the intake and exhaust allowing for aspiration to the atmosphere. The head consisted of an aluminum plate with a small clearance volume machined into it. It had a slot to insert the transducer that positions the transducer perpendicular to the face of the piston head. The head also had a port for monitoring intake temperature via a thermocouple.

The engine was motored using a dynamometer controlled via a VFD. Without combustion in the engine, oil built up in the cylinder so the engine was motored without the use of oil

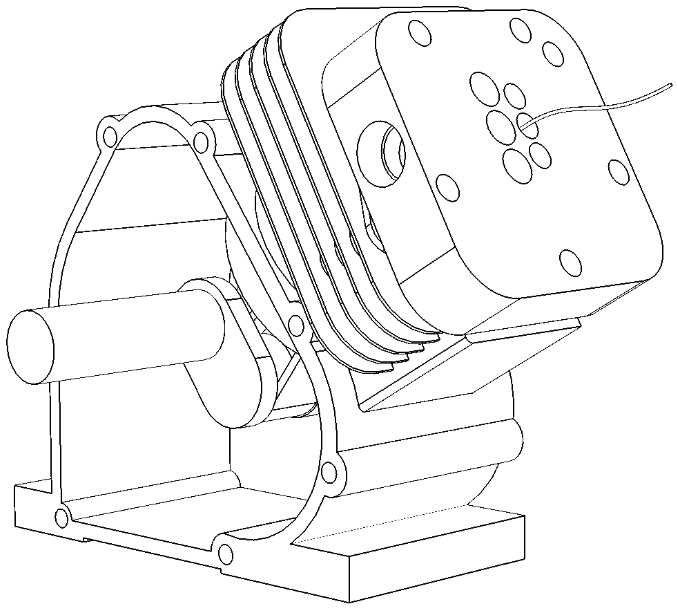


Figure 2.1: *The engine was fitted with a modified head. The head allowed for the placement of multiple ultrasonic transducers. Only a single transducer was used in these experiments, however. The head also provided and intake and exhaust port on opposite sides, respectively.*

as a lubricant. A sacrificial Teflon piston ring was used to prevent binding as well as the application of a heavy petroleum jelly to the cylinder walls before testing.

2.2 Methodology

Three intake temperatures were tested at two engine speeds with experimental temperature measurements being taken at each crank angle (CA) at 45 degree increments starting at 0 CAD through 360 CAD for each case. The incremental optical encoder was used to monitor piston position. When the desired trigger CA was reached the ultrasonic signal was sent. The sampled data was 2000 samples, sampled at a frequency of 2 MHz. This gives 1ms of data per record. The sampling program was setup such that the data was buffered and each record had an offset before the triggered start. CA information was also recorded for each

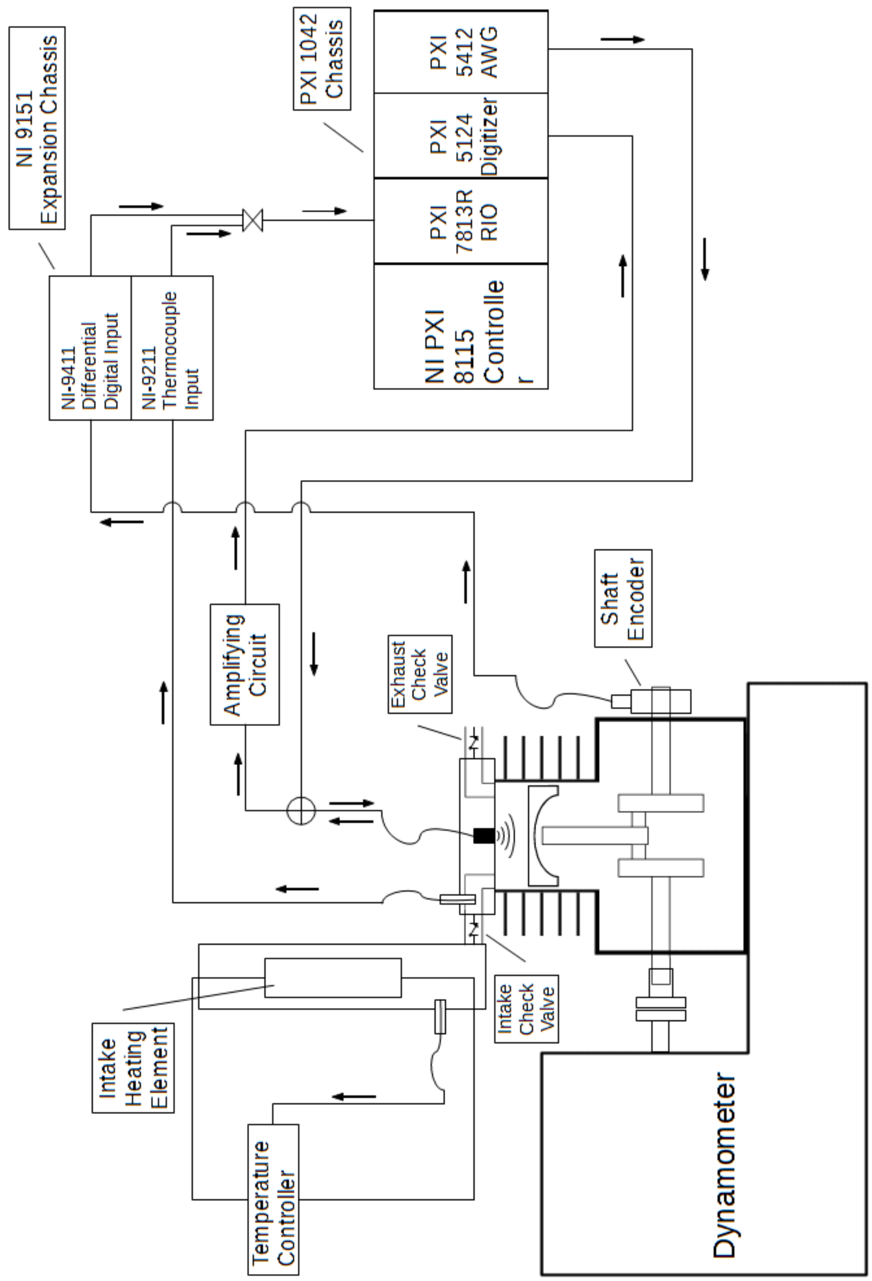


Figure 2.2: *Diagram of test setup.*

test. This data again is buffered with 2500 samples taken at the same 2 MHz frequency with an indicated trigger sample indicating the offset, or start point.

The transducer used for testing was an Airmar AT200 with a resonant frequency of 200 kHz and maximum operating temperature of 90 ºC. The signal sent had an amplitude of 5 V peak to peak with 4 pulse cycles at a frequency of 200 kHz. Initial stationary readings were taken at each trigger crank angle at room temperature (300 K). These readings were used for reference and calibration of the system. For each trigger angle shown in Table 2.2, the piston was positioned at the trigger location and 50 readings were taken in succession.

Temperature data was collected according to the schedule in Table 2.2 with samples being taken over the span of 3 days. Each sample is independent. For room temperature (300 K) samples, the intake heater was not turned on. The engine was set to the required speed by the VFD and the test trigger angle was controlled within the LabVIEW VI control software. Once the engine was motoring, the software was started and recorded the sample. This process was repeated for each sample, trigger and temperature. If the temperature was to be elevated, the intake heater was set to an experimentally determined temperature that kept the intake temperature consistent at the desired test condition. The process described above was then repeated for each of these tests. Due to the lack of engine oil and a need for maintaining constant temperature in the engine cylinder, the setup was only run for 10 minutes at a time before it was shut down to cool off before resuming testing.

Each sample taken consisted of 50 independent readings. Because the engine cycle was faster than the processing time in the LabVIEW VI, each reading was separated by 10 cycles. In other words, when the engine passed through the trigger crank angle, the ultrasonic signal was sent and received; while the control system was pulling the signal from the buffer, the engine was still motoring and did not trigger to send the next signal until 10

Table 2.2: *Data collection routine. Each sample consists of 50 readings taken at 10 engine cycle intervals.*

Temperature	Room Temperature [300K]		Elevated Temperature [350K]		High Temperature [380K]	
Engine Speed	300 rpm	1200 rpm	300 rpm	1200 rpm	300 rpm	1200 rpm
Trigger Angle	Samples Taken					
0 CAD (TDC)	4	2	2	2	2	3
45 CAD	4	2	2	2	2	2
90 CAD	4	2	2	2	2	4
135 CAD	4	2	2	3	2	2
180 CAD (BDC)	5	3	2	3	4	12
225 CAD	4	2	2	3	2	2
270 CAD	4	2	2	2	2	4
315 CAD	4	1	2	2	2	2
360 CAD (TDC)	4	1	2	2	2	3

engine cycles had passed. This gave sufficient time to record the signal in a file. A sample therefore required 500 engine cycles to complete. The samples were saved into a comma separated values file to be post processed.

2.3 Data processing

Data files contain the intake temperature reading, engine speed calculated in the VI, signal offset, time step between signals and the raw signal in the odd numbered lines. The trigger point number and crank angle information is contained in the even numbered lines. The file information lines at the top of the files contain the time-stamp, date and test condition information. Data processing was done in Matlab after all samples had been collected.

Each data file represents a sample and contains 50 independent readings or signals. Each signal was independently processed to determine the temperature prediction from the time of flight. Not all signals were able to be included in the analysis. Many signals contained too much noise or did not have a sufficient signal return to process. Through taking multiple readings or signals, some of this uncertainty was minimized because the predicted temperatures were averaged together to produce a data point. Unfavorable conditions for some of the tests did lead to zero readings being sufficient to produce a sample. The test setup was dismantled shortly after collecting the data preventing the re-taking of the bad data sets.

Using a single transducer as both the transmitter and receiver presented some difficulties in data processing. Figure 2.3 shows the differences in received signals dependent on the trigger crank angle. For most trigger angles, the reflected signal can be seen multiple times. Ringing is also clearly evident for approximately a quarter of a millisecond. The four cases where the piston was closest to top dead center (TDC) (0 CAD, 45 CAD, 315 CAD, 360 CAD), the first few echos occur within the ringing portion of the signal. These echo signals also overlap each other resulting in difficulty identifying signal location. Because of this an algorithm for selecting and recording which signal was processed was required.

The raw signals had a high frequency chirp noise on the signal. This was removed through a low-pass filter of 800 kHz. The Hilbert transform was then taken to get the analytic signal envelope [34]. To select the signals the Hilbert transform signal was then passed through a Savitzky-Golay filter with a large amount of averaging, to produce a smooth signal envelope [35]. The temperature and trigger angle information were then used to estimate the theoretical echo start locations for multiple echo returns. The signal was then selected by searching for a maximum around the theoretical echo location. The ringing noise was avoided by searching beyond a minimum time. The echo number was determined by the theoretical calculation and time around which the search begins.

Signals without a strong enough echo to detect were discarded by analyzing the first deriva-

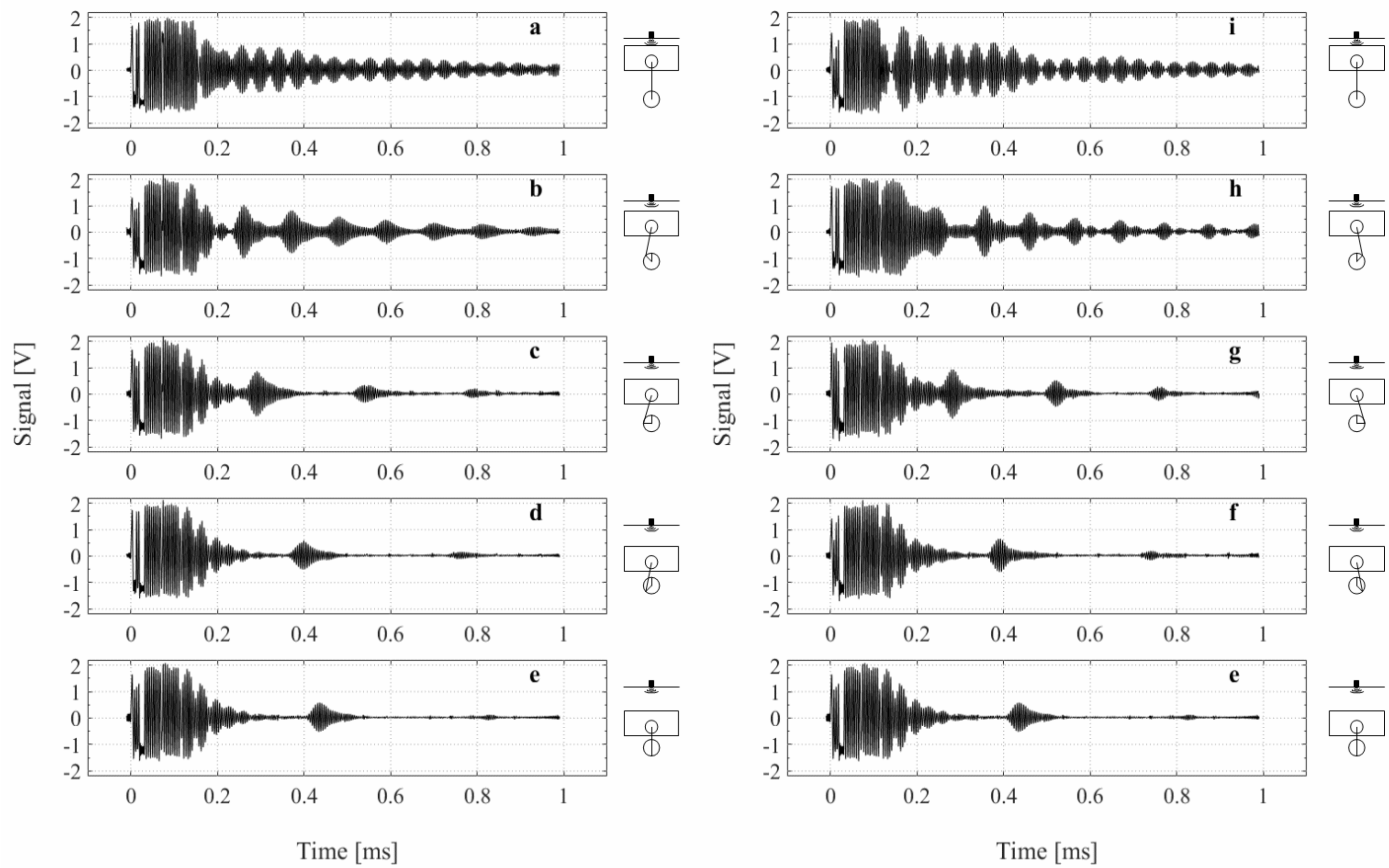


Figure 2.3: *The number of echoes received at each trigger CA. The trigger angles 0, 45, 315 and 360 the first echoes occur before the ringing has damped out. For each sample the first echo available after 0.35 ms was analyzed.*

Table 2.3: *Engine parameters piston location parameters. Crank radius is derived from the engine stroke, connecting rod length was given when asked for by the manufacturer and the clearance was measured manually.*

Parameter	Value [m]
Crank radius	0.031
Connecting Rod	0.098
Clearance	0.0076

tive of the smoothed signal. If the first derivative did not achieve a minimum value indicating a strong echo front, the signal was discarded. The position of the signal was then passed to the Kalman filtering program to determine the echo envelope of the signal. The Kalman filter is covered in more detail in section 2.3.1.

The Kalman filter was used to fit the echo envelope of each signal that was passed to it by the signal selection algorithm with the goal to accurately determine the ToF for a given signal. Not all of the signals passed to the filter were able to be fit successfully. This was likely due to additional noise or other anomalies in the signal not removed by the pre-processing. Signals where the fit parameters were non-real or negative could easily be removed. Finally, each signal was plotted manually to discard any remaining signals that did not fit correctly.

Remaining signals in each sample group were then used to calculate the predicted temperature based on the time of flights measure from each signal. The flight distance was calculated based on the trigger angle and engine parameters given in Table 2.3. Equation 2.1 is the equation for piston position where r is the crank radius, A is the crank angle and l is the connecting rod length[20]. The crank radius is half of the engine stroke which is given in the data sheet for the engine and the connecting rod length was given by the manufacturer when requested. When the crank angle was 0 degrees the piston was at TDC. The flight distance was then calculated by taking the difference between the maximum extension of the piston (x_{max}) and the actual position (x) plus the clearance distance as shown in equation 2.2.

$$x = r\cos A + \sqrt{l^2 - r^2 \sin^2 A} \tag{2.1}$$

$$FD = (x_{max} - x) + clearance \tag{2.2}$$

Equation 1.1 was solved for temperature with values of M = 29 kg/kmol, γ = 1.4 and R = 8314 J/kmol-K and equation 1.2 with the path length calculated using equation 2.2 the temperature of the medium can be estimated. For each sample the estimated temperatures from the individual signals were averaged together. They then could be compared with the measured temperatures based on the intake thermocouple.

2.3.1 Kalman Filtering

Kalman filtering is a technique commonly used in navigation applications such as spacecraft reentry [36], robotic simultaneous location and mapping (SLAM) [37] and global positioning systems [38, 39]. The method has also been applied in moment estimation in chemically reacting systems [40]. Kalman filtering is used for state and parameter estimation algorithm for dynamic systems [41]. When Kalman first published the recursive technique in 1960, it was applied to discrete-data linear filtering situations as a way of removing white noise [42]. As computing power increased through the second half of the century, research interests in the Kalman filter only increased as it is useful for many systems. The basics of the Kalman filter are summarized well by Welch and Bishop [43].

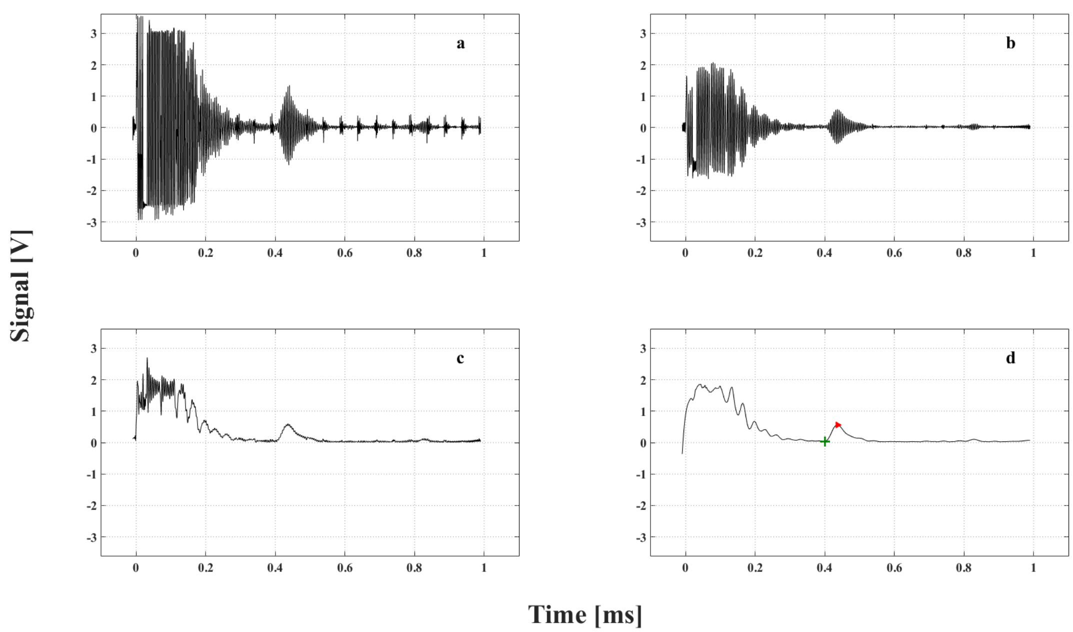


Figure 2.4: *Signal selection process where (a) is the raw signal; (b) is the signal after a low-pass filter is applied at 800 kHz; (c) is the analytic signal envelope from Hilbert Transform; (d) is the echo envelope with a Savitzky-Golay filter applied for smoothing the signal. The green + indicates the theoretical time of flight whereas the red ▸ is the maximum point of the echo envelope and subsequent start location for the Kalman Filter.*

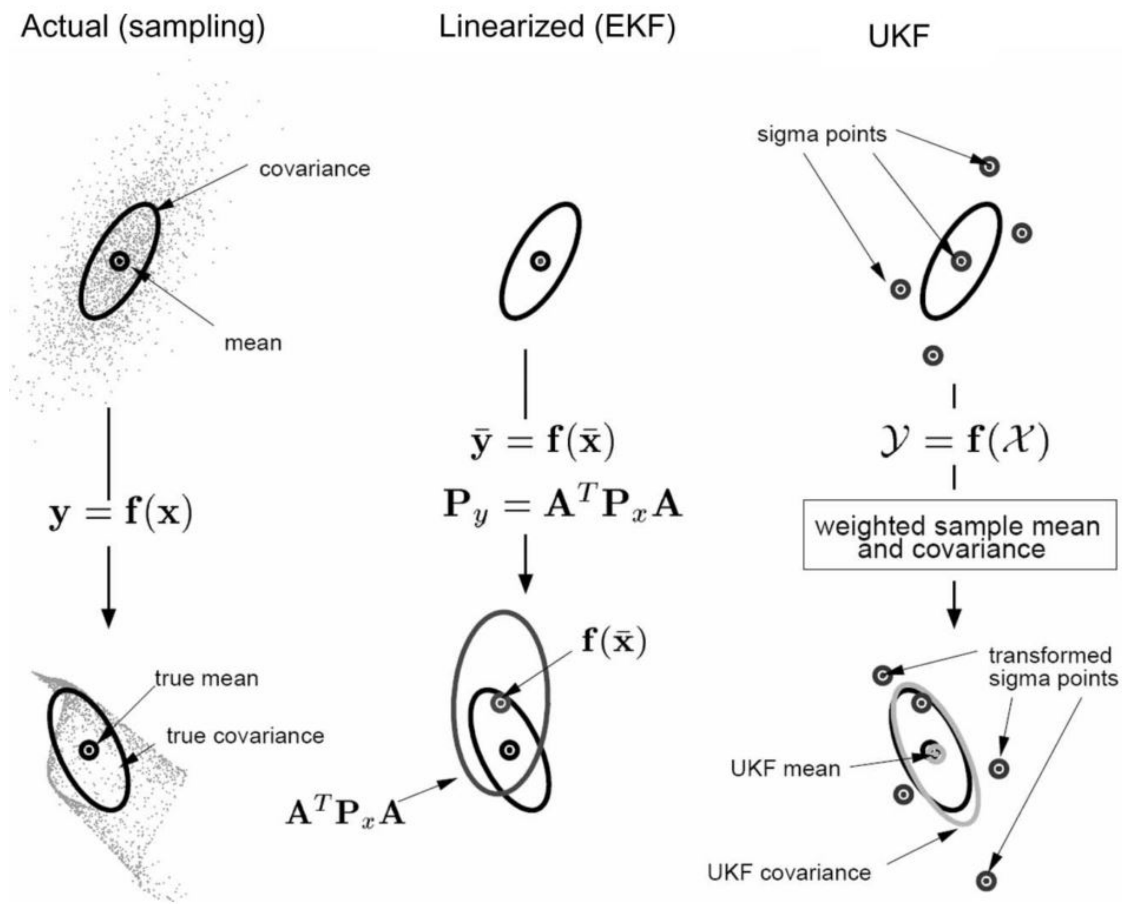


Figure 2.5: *Depiction from [5] showing the advantages of the unscented transform compared to traditional linearization techniques.*

The Kalman filter is intended for linear systems. To be used with nonlinear systems it relies on a linearization of the system known as the Extended Kalman filter (EKF). Linearization however can lead to errors in estimation. Julier and Uhlmann, in 1997, theorized the Unscented Kalman filter (UKF) for use on nonlinear systems [44, 45]. Wan and van der Merwe created the depiction shown in Figure 2.5 of the advantages of the unscented transform for broad applications of nonlinear systems [5]. The UKF uses weighted sample points for the estimation of the mean and covariance which leads to much more accurate linear estimations of the actual state and parameters.

Angrisani et al. first applied the EKF to ultrasonic signals as a means of fitting an echo envelope in 2004 [46] and the results were published in 2006 [47]. The same authors then later applied the UKF to the same problem yielding even more accurate results [48]. The method involves treating the echo envelope given in equation 2.3 as a nonlinear process where $A_0$is the echo amplitude, α and T are transducer and condition dependent parameters, τ is the ToF and $t_s$is the sampling period. k is the iteration indice. From this, the state vector is the combination of the shape parameters [A_0, α, T, τ] and equation 2.3 is the measuring equation.

$$A(kt_s) = A_0 \left(\frac{kt_s - \tau}{T} \right)^{\alpha} e^{\left(\frac{kt_s - \tau}{T} \right)} \tag{2.3}$$

To process the signals in this study, the methods of [47] and [48] were followed using a modified Matlab code produced by Yi Cao of Cranfield University in 2008 who references [45]. Figure 2.6 shows the progression of the signal processing from raw signal to echo envelope. The raw signal (a) was first put through a low pass filter to remove the high frequency noise (b). The signal envelope was then found by taking the Hilbert transform and removing the ringing found at the beginning of the signal (c). Finally, using the signal location information found in a previous processing step, described in section 2.3, the signal was isolated for fitting the echo envelope by selecting a window around the specific location (d). It was found through through experimentation that the Kalman filter worked more successfully when the Hilbert transform was smoothed through a Savitzky-Golay filtering operation (shown in (d)). This step helped to remove noise from the signal. Plot (e) shows the final result of the echo envelope overlayed on the signal. Initial values taken for the state vector are shown in Table 2.4. The initial value of τ was determined from the time corresponding with the start of the evaluation window.

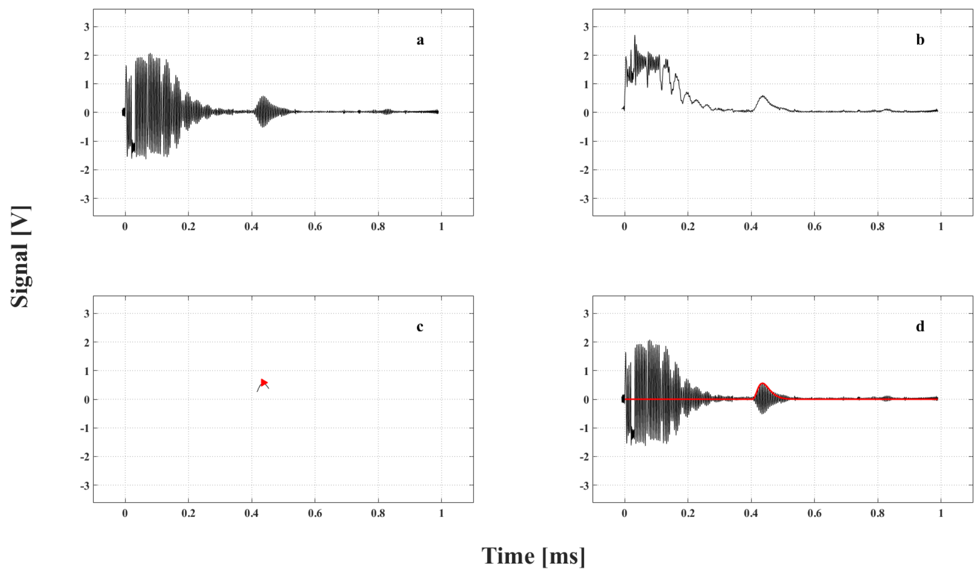


Figure 2.6: *Signal processing showing the fitting of the echo envelope given in Equation 2.3 using a Kalman Filter method. (a) is the low-pass filtered signal with an 800 kHz cutoff. (b) is the analog signal envelope given by the Hilbert Transform. (c) is the portion of the signal selected for the Kalman filter fitting algorithm with the indicated red ▸ as the echo location position selected previously. (d) is the resulting echo envelope overlayed on the low-pass filtered signal.*

Table 2.4: *Initial state vector values for the echo envelope.*

A_0	α	T	τ
0.41	2.7	$11e^{-6}$	$0.95 * windowStart$

3 Results

3.1 Stationary reference

Processing the data with the engine parameters given in Table 2.3 yields the predicted temperature results shown in Figure 3.1. These measurements are from a single sample taken at room temperature with the piston stationary at a given crank angle. The reference line is the measured room temperature of 299 K. All of the predicted temperatures were within 10% of the reference measured temperature for the stationary piston.

The data does show a parabolic shape similar to the shape of a piston position vs. time plot. Based on this, a sensitivity analysis was done to investigate whether the sinusoidal trend could be removed from the stationary piston temperature data. Using the 180 degree trigger angle data point as a reference, it was determined that the flight distance, for this sample, needed to be 0.384 mm longer to have the measured and predicted values agree. Increasing the clearance distance alone would exacerbate the disagreement between the measured temperature and the other samples. It was therefore determined that two parameters would likely need to be adjusted. Through trial and error it was determined that the most reasonable results were achieved when the crank radius is increased by the 0.384 mm and the clearance is decreased by 0.384 mm meaning the assumed crank radius was shorter

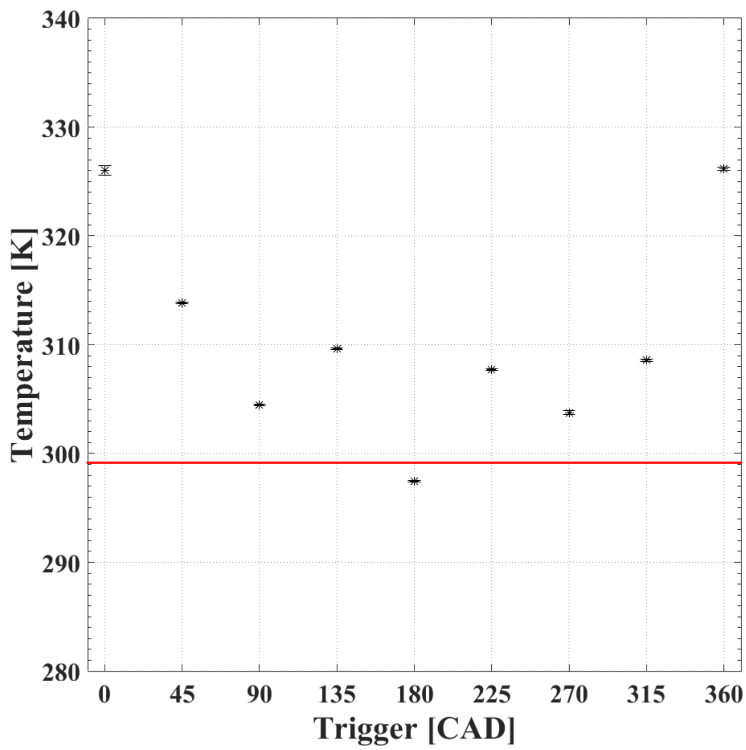


Figure 3.1: *Stationary reference predicted temperatures using the piston location parameters given in Table 2.3.*

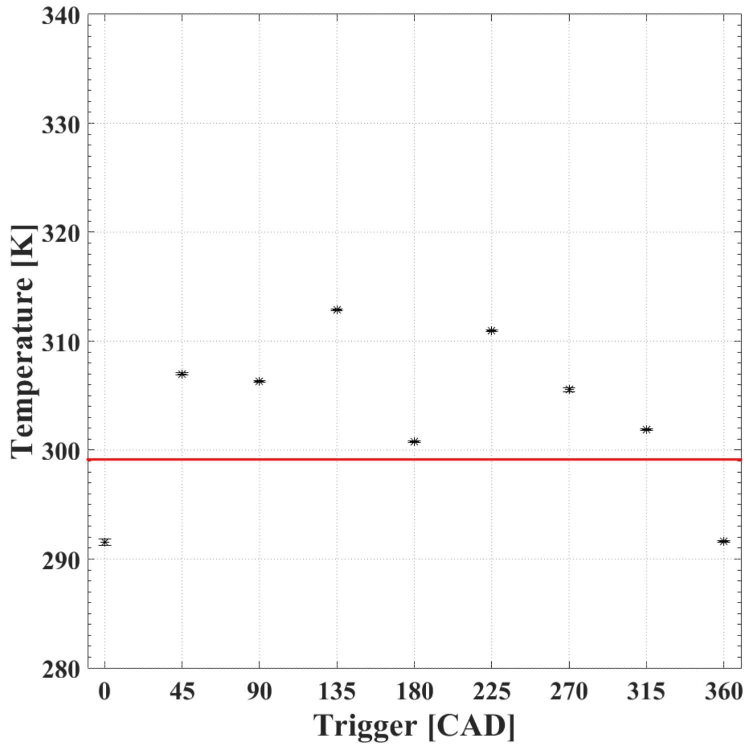


Figure 3.2: *Predicted temperatures for stationary reference samples using the corrected crank radius and clearance volume.*

than the results indicate and the measured clearance was too long. This change effectively applies the adjustment at 2x to the 180 degree trigger data.

Figure 3.2 shows the stationary data recalculated with the adjustment discussed above incorporated. It is clear here that the geometry parameter adjustment had the greatest effect on the TDC triggered data (0 degree and 360 degree). Also notable is the effect of the changes on the BDC 180 degree trigger, because the change is applied double to this data point, the prediction goes from being below the measured value in Figure 3.1 to above in Figure 3.2.

3.2 Motoring measurements

Intake temperatures were measured for each reading taken. These too were averaged for each sample. Figure 3.3 shows the measured temperature averages for each sample by trigger angle. The reference lines are the average temperature across the trigger angles and the 95% confidence interval (CI) about those means. Error bars on each data point are also 95% CI. Plot a is the room temperature (300 K) 300 rpm case, b is 300 K 1200 rpm. Plots c and d are 50 °C (323 K) at 300 rpm and 1200 rpm respectively. The final plots, e and f, are high temperature 80 °C (353 K) again at 300 rpm and 1200 rpm engine speeds. Some test conditions are not shown in the graphs because there was not enough data collected to be significant.

Predicted temperatures are shown in Figure 3.4 for the motoring cases. Again the lettered plots are the same as in Figure 3.3. The reference lines shown are the same plotted lines as those in the Figure 3.3 as well. In general, the lower speed test cases are in better agreement with the measured temperatures and show better stability over throughout the duration of the test. For the slower speed recordings (a,c,e) the TDC (0 degree and 360 degree) predicted temperatures are significantly less accurate than the recordings when the piston is farther from the sensor. In the corresponding high speed plots this trend appears to have reversed itself somewhat.

Figure 3.5 shows the difference between the measured intake temperature and predicted temperature values for each individual sample. The letters of the charts correspond with Figures 3.3 and 3.4 and the error bars are 95% CI of each sample. It is seen here that many of the trends discussed above hold when looking at the difference between measured intake temperature and the predicted ToF temperature. All of the plots, with exception of e, the

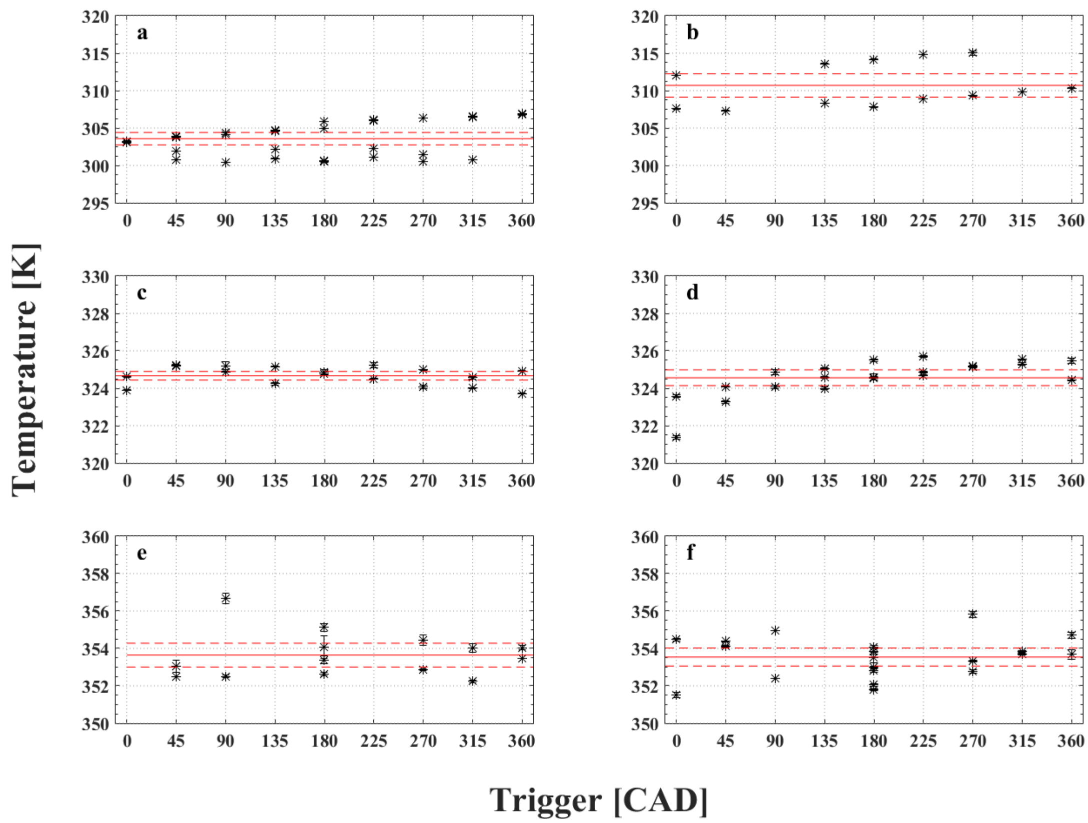


Figure 3.3: *Thermocouple measured intake temperatures for (a) room temperature, 300 rpm (b) room temperature, 1200 rpm (c) elevated temperature, 300 rpm (d) elevated temperature, 1200 rpm (e) high temperature, 300 rpm and (f) high temperature 1200 rpm. The reference lines are the average measured intake temperature and 95% CI bounds. Error bars are 95% CI.*

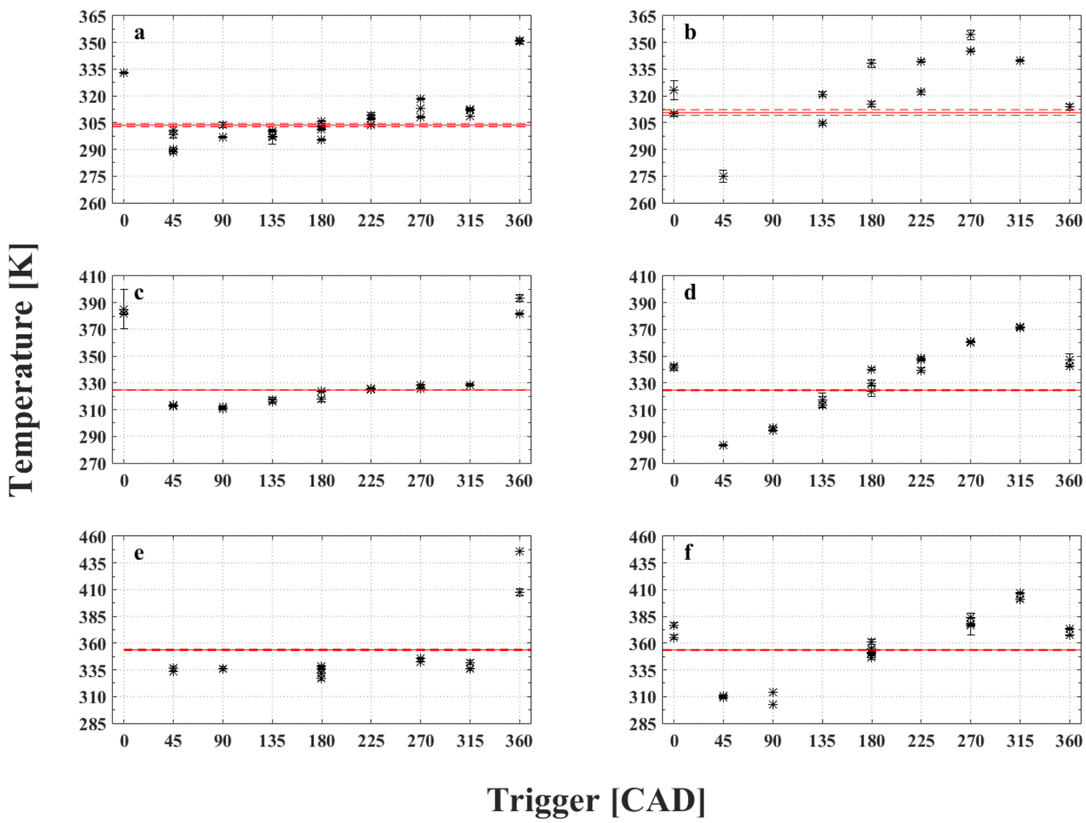


Figure 3.4: *Ultrasonic thermometry predicted temperatures for (a) room temperature, 300 rpm (b) room temperature, 1200 rpm (c) elevated temperature, 300 rpm (d) elevated temperature, 1200 rpm (e) high temperature, 300 rpm and (f) high temperature 1200 rpm. The horizontal reference lines are the average measured intake temperature and 95% CI bounds. Error bars are 95% CI.*

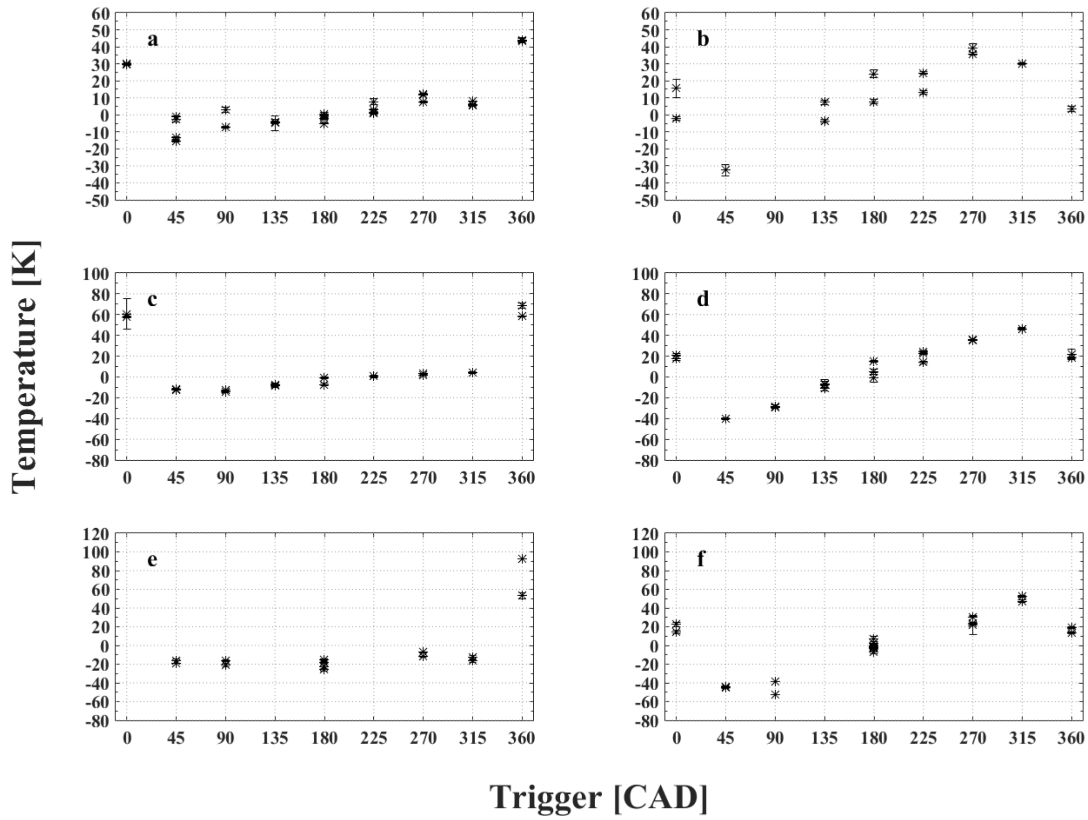


Figure 3.5: *Difference of predicted and measured temperatures for (a) room temperature, 300 rpm (b) room temperature, 1200 rpm (c) elevated temperature, 300 rpm (d) elevated temperature, 1200 rpm (e) high temperature, 300 rpm and (f) high temperature 1200 rpm. Error bars are 95% CI.*

high temperature 300 rpm plot, show a general trend of increasing predicted temperatures with increasing trigger angle despite the fact that they were not all collected in this order.

Tables A.1 through A.6, in Appendix A summarize the numerical and statistical data for the experimental conditions. The average measured and predicted temperatures are given as $\overline{x}_{measured}$ and $\overline{x}_{predicted}$ respectively. The relative standard error of the mean is given for each value as well. Finally the percent error between the two values is reported as $\epsilon_{\%}$. It is important to note in this data the very small deviations within the measurement samples indicating a high level of repeatability between measurements.

Table 3.1: *Summarized percent error by the root sum of squares method for each test condition. The gradient shading shows the relative error between the different test conditions.*

Temperature	Room Temperature [300K]		Elevated Temperature [350K]		High Temperature [380K]	
Engine Speed	300 rpm	1200 rpm	300 rpm	1200 rpm	300 rpm	1200 rpm
Trigger Angle	RSS percent errors					
0	14%	5.1%	26%	8.6%	-	7.5%
45	6.9%	11%	5.2%	18%	7.2%	18%
90	3.6%	-	5.9%	13%	7.4%	19%
135	2.9%	2.6%	3.5%	4.7%	-	-
180	1.9%	8.0%	2.2%	4.8%	12%	3.2%
225	2.8%	8.9%	0.2%	11%	-	-
270	7.2%	17%	1.1%	16%	3.9%	13%
315	3.7%	9.7%	1.8%	20%	5.9%	20%
360	20%	1.1%	28%	8.7%	30%	6.6%

The percent error for each test condition is summarized using the root sum of squares method in Table 3.1. In general, error is lower at the longer flight distances and lower engine motoring speed. Over 30% of the tests have a combined error of less than 5% and over 60% of the tests have a combined error of less than 10%.

4 Discussion

The results in Tables A.1 through A.6 show that the ultrasonic measurements technique used here is capable of highly repeatable measurements. This is likely due to the Kalman filtering method used for determining the ToF for echo return. Angrisani et al. [48] showed that this processing technique can lead to extremely accurate measurements reporting ~0.25% bias and standard deviations experimentally for ultrasonic distance determination using a ToF method. Using the Kalman filter as a measure of the ToF in thermometry applications is one of the key findings of this work.

Bauer, Tam, Heywood and Ziegler successfully measured the temperatures in an engine cylinder using ultrasonic ToF measurements [3]. They used a signal detection method based on the peak of the received signal and report this technique alone accounts for 2% uncertainty in their measurement and is the largest source of error, they believe. Utilizing a Kalman filter technique to measure their ToF could further validate their findings.

Both of the previous implementations of ultrasonic ToF measurements in internal combustion engines have used independent, fixed transmitter and receiver. This method has a distinct advantage in that it allows researchers to measure the precise flight path length over which the flight time is being measured. Admittedly, the method used here suffers due to the variable flight distance because the piston is moving during the measurement. This is

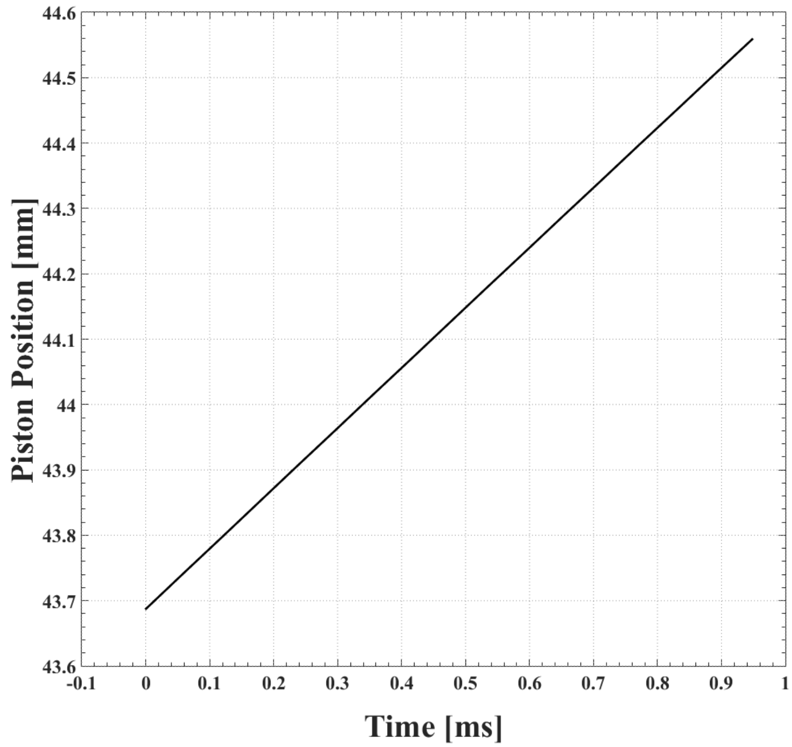


Figure 4.1: *Flight distance versus time for a signal triggered at 90 CAD motoring at 300 rpm.*

an issue that can be resolved through careful measurement and assembly of the test setup. Temperature measurement by ultrasound ToF is extremely sensitive to the flight path. Figure 4.1 shows the piston position versus time for the duration of a sample triggered at 90 CAD. Between the beginning and the end of the recording is 0.87 mm which, in the example used here, is over 12 K change in predicted temperature or 4% error at 300 K. This large change results in the need for the signal to traverse the distance 4x due to the transducer ringing interference.

Piston movement will also cause a Doppler effect in the ultrasonic frequency. Equation 4.1 describes the change in frequency observed for a moving object. At worst case conditions here, the piston is moving at 1200 rpm giving a linear velocity of 3.6 [m/s] and the speed of sound is 347 [m/s] at room temperature. This yields a frequency shift of around 2 kHz for the 200 kHz signal; a change of about 1%. This shift is not likely to cause gross errors in the time of flight measurement made by the Kalman filter. Any other potential shifts from the data processing were not corrected for in the final measurements.

$$\Delta f = \frac{\Delta v}{c} f_0 \tag{4.1}$$

It is interesting that the results in Figure 3.5 show a trend where the predicted measurements taken when the piston is moving away from the sensor (45 CAD, 90 CAD, 135 CAD) predict a lower temperature than the samples taken when the piston is moving toward the transducer (225 CAD, 270 CAD, 315 CAD). The trend here would stem from the principle that increasing the flight distance, increases the time of flight and therefore decreases the predicted temperature. The converse is true as well.

One might attribute the trend of increasing predicted temperature as trigger CAD increases to the piston movement itself however a conservative correction for this piston movement indicates the piston movement correction does not account for the trend itself. These results are shown in Figure 4.2 and should be compared with those in Figure 3.5.

Another possible contribution to this trend is that the intake temperature measurement is not representative of the temperature in the cylinder. The trend of predicted temperature increasing with CAD trigger generally holds across test conditions. Given these test were conducted over multiple days with test conditions being repeated at different times and that the CAD triggers were not necessarily run consecutively, there is likely another aspect contributing to this trend; such as measured temperature inaccuracies.

As the piston moves away from TDC, it pulls in ambient air which is at, theoretically, intake temperature. This intake temperature would likely represent the lowest temperature

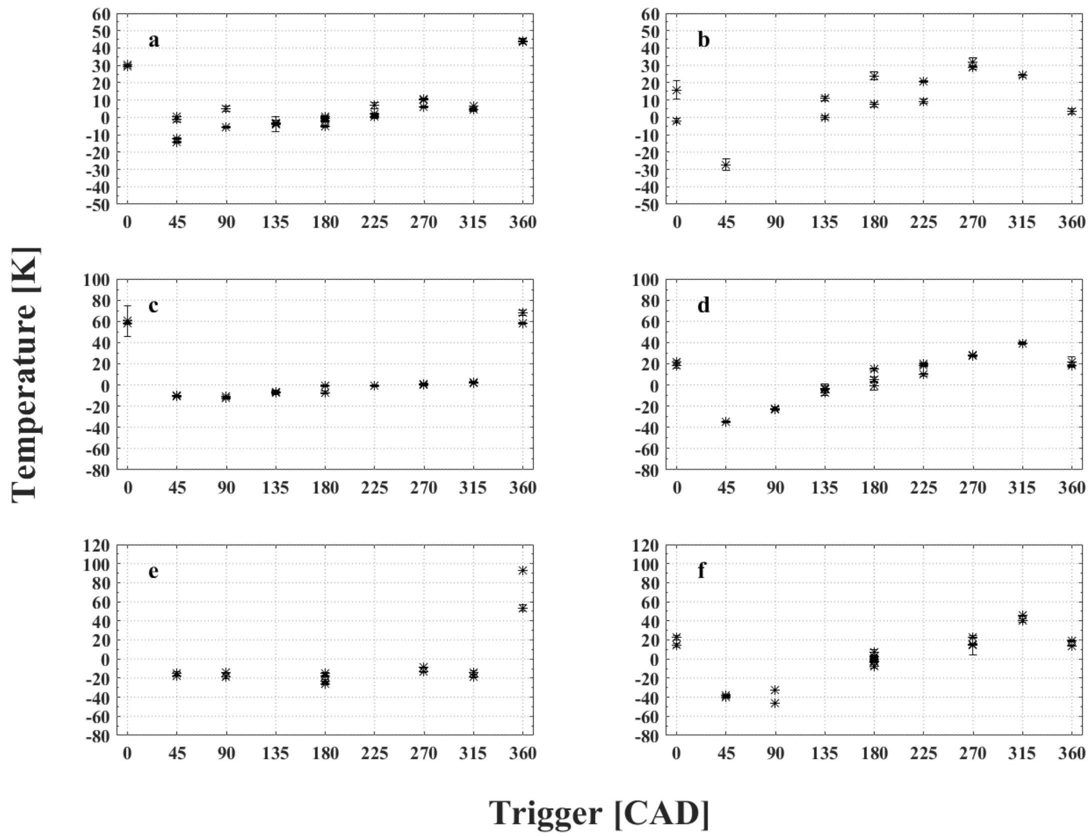


Figure 4.2: *Difference of predicted and measured temperatures for (a) room temperature, 300 rpm (b) room temperature, 1200 rpm (c) elevated temperature, 300 rpm (d) elevated temperature, 1200 rpm (e) high temperature, 300 rpm and (f) high temperature 1200 rpm using a corrected flight distance based on the movement of the piston during data collection. Error bars are 95% CI.*

test conditions. Additionally, intake conditions will represent a low pressure environment which can affect the prediction based on ToF. For each subsequent CAD trigger, the gas has had more and more time to reside in the cylinder where it can be radiantly heated. When the piston turns around, after BDC, the opposite conditions exits. There is now only the gas inside the cylinder, which continues to be heated by the piston walls and the pressure increases due to compression. To adjust the temperature from 330 K to 310 K, using the ideal gas law, would require less than 1 psi pressure change on the intake stroke. This is very likely a cause of error in this study. Unfortunately pressure was not monitored throughout the tests.

Also supporting the pressure bias error effect is the that the 180 CAD trigger samples are generally the most accurate; when the pressure is most stable due to the piston dwell at BDC. The higher speed data also appears to exaggerate the slant of the data, likely due to larger pressure fluctuations on intake and exhaust.

One final point of discussion is the data collected at 0 CAD and 360 CAD trigger points. These points proved to be difficult to complete these measurements at due to their extremely short flight path. Using Equation 1.1 at a temperature of 353 K with the other parameters found in Section 2.3, the speed of sound in air is 376.4 m/s giving a wavelength of 1.88 mm for the 200 kHz signal. The flight distance at these positions is 7.21 mm. The wavelength should be sufficiently short to not have an effect on the measurement principle. More likely, the total length of the echo signal is overlapping with subsequent echo's causing a possible measurement error. Interestingly, for the lower speed data, these points have the largest amount of error. At the higher motoring speeds they are more successful. This phenomenon is not well understood here and deserves further investigation in future studies.

5 Conclusions & Recommendations

5.1 Summary and Conclusions

Overall, the method of pyrometry first suggested by Mayer in 1873 is not only a serviceable means of temperature measurement in internal combustion engines, it also has the potential to open new avenues of combustion and energy research. The proof of concept experiments performed in this work used inexpensive acoustic equipment and was able to achieve results within an accuracy of 1% compared to the reference thermocouple measurement. Additional work needs to be done on correcting the measurements for pressure and piston location but the work presented here supports the claim that ultrasonic thermometry can be used to take path and time averaged temperatures throughout a reciprocating internal combustion engine cylinder.

The study used LabVIEW software and hardware to trigger ultrasonic signals at specific CAD positions in a reciprocating engine. The engine was being motored by a dynamometer and used a modified head with temperature controlled intake. The modified head holds the ultrasonic transducer that is used as both a transmitter and receiver. The ultrasonic signals are sent when the trigger crank angle is reached, propagate through the space in the engine cylinder, reflect off the piston head and return to the transducer. They are saved in a file for post processing operations.

A Kalman filter is used to measure the time of flight of the ultrasonic signal. The signals are processed by removing high frequency noise with a low pass filter. A signal envelope is then found with the Hilbert transform. From this point the echo is isolated through an algorithm that uses the theoretical time of flight based on temperature and engine parameters. The echo envelope is then iteratively fit to the signal envelope to derive the ultrasonic ToF.

Results for this study show that the method is successful at predicting the temperature to within an accuracy of less than 10% for the majority of test conditions used here. Not all tests had this level of accuracy; however, engine parameter correction and pressure compensation would likely improve the method's accuracy.

There are two significant advantages of the method used in this research over those used by previous researchers. First, this method inherently takes a more complete temperature of the cylinder space because it samples a path from the top to the bottom. Livengood relies on a side chamber for temperature measurement whereas Bauer is sampling across the top of the cylinder. Both of these are subject to possible convection or mixing irregularities where the sampled portion is not representative of the whole.

Second, the ToF measurement used in this study marks a significant improvement in the repeatability over the aforementioned two studies. Within sample deviations are well below 0.25% when the there are a significant number of samples to analyze. The work here should be seen as a step forward for this area of research by using more advanced control and processing techniques.

5.2 Recommendations

There are a number of recommendations that can be made for future research in this area. First, general improvements to the control and measurement of the motoring engine are always recommended. A more precise shaft encoder would help greatly to increase piston location and flight distance accuracy. At the timescales over which these experiments evolve it is likely that less than a CAD is passed. Increasing the engine speed for the experiments would also help to alleviate this but not correct the issue entirely. Future work should also include the measurement of the engine components by a high accuracy measurement device to ensure the flight path can be accurately deduced.

Pressure correction is needed in these experiments to ensure no bias is introduced from increased or decreased pressure. When performing the experiments presented here pressure was not monitored as there was no compression intended. Even the small amount of expansion and compression experienced from the check valves was enough to induce temperature errors in this experiment. These types of bias errors should be easily compensated for through pressure monitoring.

The quality of the acoustic equipment used in this type of study should be increased. These proof of concept tests were done using a cost effective transducer and amplifying circuit. High fidelity equipment that introduces more noise rejection and more robust signals is needed to increase the reliability of this method. A transducer that can withstand the conditions within a combusting engine cylinder should also be developed. Bauer uses a water-jacket encased transducer to protect from overheating the piezo crystal in their setup.

Clearly the next step for this work is to increase the temperature and engine speeds to even more relevant operating conditions. Ideally an engine could be designed such that further proof of concept testing could be completed at 2000 rpm without the concern of overheating as was found in this research. A water cooled engine block may be useful for this type of test to control temperature. Finally, completing testing in a combusting engine should be achievable and then more complex applications such as spatial mapping of temperature gradients within the cylinder.

Other methods of signal analysis can also be explored for this type of measurement. There are methods of speed of sound measurements that involve looking at frequency shift information [49, 50]. These methods rely on the same principle of the dependence of the speed of sound in air, they analyze continuous wave information though rather than a ToF measurement. Continuous wave frequency shift methods may provide a more instantaneous temperature reading than the methods provided here and should be explored for their potential benefits.

Bibliography

[1] F Massa. Ultrasonic transducers for use in air. *Proceedings of the IEEE*, 53(10):1363–1371, oct 1965.

[2] J. C. Livengood, T. P. Rona, and J. J. Baruch. Ultrasonic Temperature Measurement in Internal Combustion Engine Chamber. *Journal of the Acustical Society of America*, 26(5):824–830, 1954.

[3] W Bauer, C Tam, J Heywood, and C Ziegler. Fast Gas Temperature Measurement by Velocity of Sound for IC Engine Applications (972826 Technical Paper)- SAE Digital Library. *Society of Automotive Engineers, Inc.*, 1997.

[4] K. Allmendinger, M. Hart, R. Loisch, and M. Scherer. Method and device for determining gas pressure and temperature in an hollow space, May 28 2002. US Patent 6,394,647.

[5] E.A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 153–158. IEEE, 2000.

[6] P. R. N. Childs, J. R. Greenwood, and C. A. Long. Review of temperature measurement. *Review of Scientific Instruments*, 71(8):2959, aug 2000.

[7] J.W. Dally, W.F. Riley, and K.G. McConnell. *Instrumentation for Engineering Measurements*. Wiley, 2nd ed. edition, 1993.

[8] D P DeWitt and G D Nutter. *Theory and Practice of Radiation Thermometry*. Wiley, 1988.

[9] Volker Sick. High speed imaging in fundamental and applied combustion research. *Proceedings of the Combustion Institute*, 34(2):3509–3530, jan 2013.

[10] Alfred M. Mayer. On an acoustic pyrometer. *Philosophical Magazine Series 4*, 45(297):18–22, 1873.

[11] G. G. Sherratt and Ezer Griffiths. The Determination of the Specific Heat of Gases at High Temperatures by the Sound Velocity Method. I. Carbon Monoxide. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 147(861):292–308, 1934.

[12] Chauncey Guy Suits. The Determination of Arc Temperature from Sound Velocity Measurements. I. *Physics*, 6(6):190, dec 1935.

[13] Lawrence C. Lynnworth. *Ultrasonic Measurements for Process Control*. Academic Press Inc., San Diego, 1989.

[14] George Gabriel Stokes. *On the Friction of Fluids in Motion, and the Equilibrium and Motion of Elastic Solids*. 1846.

[15] J. David N. Cheeke. *Fundamentals and Applications of Ultrasonic Waves*. CRAC Press LLC, 2002.

[16] Vern O Knudsen. The absorption of sound in air, in oxygen, and in nitrogen - effects of humidity and temperature. *The Journal of the Acoustical Society of America*, 5(2):112–121, 1933.

[17] L J Sivian. High frequency absorption in air and other gases. *The Journal of the Acoustical Society of America*, 19(5):914–916, 1947.

[18] Cyril M. Harris. Absorption of Sound in Air versus Humidity and Temperature. *The Journal of the Acoustical Society of America*, 40(1):148, 1966.

[19] Leo L. Beranek. *Acoustics*. American Institute of Physics, New York, revised edition, 1986.

[20] John B. Heywood. *Internal Combustion Engine Fundamentals*. McGraw-Hill, Inc., New York, 1988.

[21] William F. Northrop, Wei Fang, and Bin Huang. Combustion Phasing Effect on Cycle Efficiency of a Diesel Engine Using Advanced Gasoline Fumigation. *Journal of Engineering for Gas Turbines and Power*, 135(3):032801 (6 pages), feb 2013.

[22] John E Dec. Advanced compression-ignition engines - understanding the in-cylinder processes. *Proceedings of the Combustion Institute*, 32(2):2727–2742, 2009.

[23] Stephen R Turns et al. *An introduction to combustion*, volume 287. McGraw-hill New York, 1996.

[24] William F. Northrop, Stanislav V. Bohac, and Dennis N. Assanis. Premixed low temperature combustion of biodiesel and blends in a high speed compression ignition engine. *SAE International Journal of Fuels and Lubricants*, 2(1):28–40, 2009.

[25] K. Amano and Y. Matsubara. Temperature and pressure detecting type spark plug, January 15 1991. US Patent 4,984,905.

[26] Y. Matsubara and S. Matsumura. Temperature sensor for use in a spark plug of an internal combustion engine, June 18 1991. US Patent 5,024,534.

[27] H. Takeuchi. Temperature sensor for internal combustion engine, March 26 2009. US Patent App. 12/232,479.

[28] A. Christ and T. HEIDINGER. Spark plug having a thermosensor, August 14 2012. US Patent 8,244,447.

[29] W. Mauermann. Ultraschall-Messung schnell veranderlicher Gastemperaturen. *Elektronik*, 5:69–72, 1975.

[30] G. Hohenberg. GASTEMPERATUR-MESSVERFAHREN DURCH LAUFZEITMESSUNG VON ULTRASCHALLIMPULSEN - EIN NEUER WEG ZUR ERFASSUNG INNERMOTORISCHER VORGANGE. *AUTOMOBIL-INDUSTRIE*, 20(2):25–38, 1975.

[31] P A Lakshminarayanan, P A Janakiraman, M K Gajendra Babu, and B S Murthy. Measurement of pulsating temperature and velocity in an internal combustion engine using an ultrasonic flowmeter. *Journal of Physics E: Scientific Instruments*, 12(11):1053–1058, nov 1979.

[32] Ichiro Higashino, Muneo Hitomi, and Jun Yoshizawa. A method to measure an exhaust gas temperature with ultrasound. *Memoirs of the Faculty of Engineering, Osaka City University*, 8:37–43, 1966.

[33] MW Dadd. Acoustic thermometry in gases using pulse techniques. *High Temperature Technology*, 1(6):333, 1983.

[34] Stefan L Hahn. *Hilbert transforms in signal processing*. Artech House on Demand, 1996.

[35] Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.

[36] Chaw-Bing Chang, Robert H. Whiting, and Michael Athans. On the State and Parameter Estimation for Maneuvering Reentry Vehicles. *IEEE Transactions on Automatic Control*, pages 99–105, 1977.

[37] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *Intelligent Transportation Systems Magazine, IEEE*, 2(4):31–43, 2010.

[38] Edward J Krakiwsky, Clyde B Harris, and Richard VC Wong. A kalman filter for integrating dead reckoning, map matching and gps positioning. In *Position Location and Navigation Symposium, 1988. Record. Navigation into the 21st Century. IEEE PLANS'88., IEEE*, pages 39–46. IEEE, 1988.

[39] Christopher Hide, Terry Moore, and Martin Smith. Adaptive kalman filtering for low-cost ins/gps. *The Journal of Navigation*, 56(01):143–152, 2003.

[40] J. Ruess, A. Milias-Argeitis, S. Summers, and J. Lygeros. Moment estimation for chemically reacting systems by extended Kalman filtering. *Journal of Chemical Physics*, 135(2011):165102, oct 2011.

[41] Charles K. Chui and Guanrong Chen. *Kalman Filtering with Real-Time Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 4 edition, 2009.

[42] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[43] Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. *TR 95-041*, pages 1–16, jul 2006.

[44] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

[45] Simon Julier and Jeffrey Uhlmann. Unscented Filtering and Non Linear Estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[46] Leopoldo Angrisani, Aldo Baccigalupi, and Rosario Schiano Lo Moriello. A measurement method based on kalman filtering for ultrasonic time-of-flight estimation. In *Instrumentation and Measurement Technology Conference, 2004. IMTC 04. Proceedings of the 21st IEEE*, volume 1, pages 210–215. IEEE, 2004.

[47] L. Angrisani, A. Baccigalupi, and R. Schiano Lo Moriello. A Measurement Method Based on Kalman Filtering for Ultrasonic Time-of-Flight Estimation. *IEEE Transactions on Instrumentation and Measurement*, 55(2):442–448, apr 2006.

[48] L. Angrisani, A. Baccigalupi, and R. Schiano Lo Moriello. Ultrasonic Time-of-Flight Estimation Through Unscented Kalman Filter. *IEEE Transactions on Instrumentation and Measurement*, 55(4):1077–1084, aug 2006.

[49] K. N. Huang, C. F. Huang, Y. C. Li, and M. S. Young. High precision, fast ultrasonic thermometer based on measurement of the speed of sound in air. *Review of Scientific Instruments*, 73(11):4022, oct 2002.

[50] Wen-Yuan Tsai, Hsin-Chieh Chen, and Teh-Lu Liao. High accuracy ultrasonic air temperature measurement using multi-frequency continuous wave. *Sensors and Actuators A: Physical*, 132(2):526–532, nov 2006.

A Tabulated results

Table A.1: *Tabulated results for the room temperature 300 rpm test condition. Blank entries indicate there were not enough quality signals received to process.*

Trigger	Sample	Signals Processed	$\bar{x}_{measured}$ [K]	Relative $SE_{\bar{x}_{meas.}}$	$\bar{x}_{predicted}$ [K]	Relative $SE_{\bar{x}_{pred.}}$	$\epsilon_{\%}$
0 CAD	**1**	49	303.03	0.0017%	333.21	0.043%	10%
	2	50	303.25	0.0020%	332.82	0.047%	10%
	3	0	-	-	-	-	-
	4	0	-	-	-	-	-
45 CAD	**1**	50	303.72	0.0012%	290.14	0.12%	4.5%
	2	50	303.87	0.00022%	288.33	0.10%	5.1%
	3	1	301.87	-	300.87	-	0.33%
	4	11	300.77	0.0032%	298.21	0.29%	0.85%
90 CAD	**1**	49	304.16	0.0016%	296.71	0.082%	2.5%
	2	48	304.34	0.0024%	297.05	0.078%	2.4%
	3	15	300.37	0.0023%	303.56	0.25%	1.1%
	4	0	-	-	-	-	-
135 CAD	**1**	45	304.54	0.0019%	300.13	0.090%	1.4%
	2	46	304.70	0.0020%	300.27	0.092%	1.5%
	3	2	302.18	0	297.25	0.72%	1.6%
	4	38	300.93	0.0029%	296.92	0.21%	1.3%
180 CAD	**1**	49	304.93	0.0019%	302.74	0.079%	0.72%
	2	50	305.86	0.0029%	305.46	0.091%	0.13%
	3	49	300.47	0.0041%	295.47	0.076%	1.7%
	4	0	-	-	-	-	-
	5	50	300.60	0.0014%	301.20	0.10%	0.20%
225 CAD	**1**	49	306.04	0.0017%	307.16	0.065%	0.37%
	2	46	306.11	0.0022%	308.84	0.091%	0.89%
	3	2	302.24	0.0017%	303.60	0.12%	0.45%
	4	21	301.13	0.0039%	308.93	0.26%	2.6%
270 CAD	**1**	50	306.37	0.0018%	318.27	0.073%	3.9%
	2	47	306.30	0.0019%	318.27	0.081%	3.9%
	3	23	300.53	0.0033%	308.26	0.089%	2.6%
	4	1	301.49	-	313.18	-	3.9%
315 CAD	**1**	32	306.45	0.0023%	312.07	0.11%	1.8%
	2	27	306.55	0.0019%	312.58	0.11%	2.0%
	3	0	-	-	-	-	-
	4	1	300.77	-	308.72	-	2.6%
360 CAD	**1**	15	306.75	0.0020%	350.37	0.19%	14%
	2	8	306.86	0.0053%	350.95	0.21%	14%
	3	0	-	-	-	-	-
	4	0	-	-	-	-	-

Table A.2: *Tabulated results for the room temperature 1200 rpm test condition. Blank entries indicate there were not enough quality signals received to process.*

Trigger	Sample	Signals Processed	$\bar{x}_{measured}$ [K]	Relative $SE_{\bar{x}_{meas.}}$	$\bar{x}_{predicted}$ [K]	Relative $SE_{\bar{x}_{pred.}}$	$\epsilon_{\%}$
0 CAD	**1**	4	307.59	0.0082%	323.16	0.84%	5.1%
	2	44	312.01	0.0045%	309.84	0.19%	0.70%
45 CAD	**1**	2	307.32	0.013%	274.73	0.62%	11%
	2	0	-	-	-	-	-
90 CAD	**1**	0	-	-	-	-	-
	2	0	-	-	-	-	-
135 CAD	**1**	18	308.30	0.0064%	304.53	0.17%	1.2%
	2	22	313.60	0.0097%	320.96	0.20%	2.3%
180 CAD	**1**	30	307.80	0.0036%	315.40	0.21%	2.5%
	2	14	314.17	0.0086%	338.18	0.36%	7.6%
	3	0	-	-	-	-	-
225 CAD	**1**	47	308.94	0.0023%	322.05	0.17%	4.2%
	2	38	314.80	0.0050%	339.48	0.098%	7.8%
270 CAD	**1**	24	309.31	0.0068%	345.24	0.11%	12%
	2	2	315.02	0.051%	354.29	0.36%	12%
315 CAD	**1**	43	309.82	0.0017%	339.91	0.085%	9.7%
360 CAD	**1**	33	310.31	0.0068%	313.78	0.25%	1.1%

Table A.3: *Tabulated results for the elevated temperature 300 rpm test condition. Blank entries indicate there were not enough quality signals received to process.*

Trigger	Sample	Signals Processed	$\bar{x}_{measured}$ [K]	Relative $SE_{\bar{x}_{meas.}}$	$\bar{x}_{predicted}$ [K]	Relative $SE_{\bar{x}_{pred.}}$	$\epsilon_{\%}$
0 CAD	**1**	11	324.62	0.0087%	385.00	1.9%	19%
	2	31	323.89	0.0052%	381.85	0.12%	18%
45 CAD	**1**	45	325.23	0.0053%	312.78	0.084%	3.8%
	2	50	325.19	0.0036%	313.43	0.080%	3.6%
90 CAD	**1**	46	325.20	0.032%	310.78	0.095%	4.4%
	2	42	324.87	0.0029%	311.92	0.095%	4.0%
135 CAD	**1**	25	324.24	0.010%	316.14	0.26%	2.5%
	2	15	325.16	0.003%	317.43	0.25%	2.4%
180 CAD	**1**	27	324.86	0.021%	317.61	0.22%	2.2%
	2	48	324.75	0.0030%	323.62	0.13%	0.35%
225 CAD	**1**	49	324.50	0.0034%	325.08	0.13%	0.18%
	2	48	325.23	0.024%	325.61	0.14%	0.12%
270 CAD	**1**	30	324.07	0.017%	325.93	0.072%	0.57%
	2	14	324.99	0.0040%	327.96	0.11%	0.91%
315 CAD	**1**	8	324.56	0.0090%	328.32	0.11%	1.2%
	2	7	324.01	0.0045%	328.37	0.14%	1.3%
360 CAD	**1**	17	324.93	0.0065%	393.21	0.32%	21%
	2	38	323.72	0.0026%	381.91	0.092%	18%

Table A.4: *Tabulated results for the elevated temperature 1200 rpm test condition. Blank entries indicate there were not enough quality signals received to process.*

Trigger	Sample	Signals Processed	$\bar{x}_{measured}$ [K]	Relative $SE_{\bar{x}_{meas.}}$	$\bar{x}_{predicted}$ [K]	Relative $SE_{\bar{x}_{pred.}}$	$\epsilon_{\%}$
0 CAD	**1**	17	323.57	0.0098%	341.39	0.34%	5.5%
	2	20	321.36	0.015%	342.62	0.20%	6.6%
45 CAD	**1**	48	324.09	0.0052%	283.46	0.12%	13%
	2	48	323.26	0.011%	283.10	0.12%	12%
90 CAD	**1**	6	324.86	0.024%	296.45	0.25%	8.7%
	2	3	324.07	0.013%	294.40	0.14%	9.2%
135 CAD	**1**	8	323.99	0.0087%	316.93	0.42%	2.2%
	2	14	324.59	0.0072%	313.44	0.39%	3.4%
	3	3	325.04	0.0081%	317.24	0.86%	2.4%
180 CAD	**1**	4	324.54	0.017%	324.04	0.67%	0.15%
	2	36	324.59	0.026%	329.80	0.36%	1.6%
	3	21	325.50	0.0089%	340.22	0.14%	4.5%
225 CAD	**1**	48	324.85	0.0061%	347.44	0.15%	7.0%
	2	42	324.69	0.0021%	348.67	0.14%	7.4%
	3	23	325.69	0.0056%	339.67	0.21%	4.3%
270 CAD	**1**	47	325.21	0.0056%	360.40	0.058%	11%
	2	41	325.13	0.0040%	361.01	0.058%	11%
315 CAD	**1**	36	325.54	0.0047%	371.81	0.16%	14%
	2	39	325.30	0.0081%	371.12	0.13%	14%
360 CAD	**1**	33	324.45	0.0099%	342.67	0.15%	5.6%
	2	6	325.46	0.021%	346.85	0.74%	6.6%

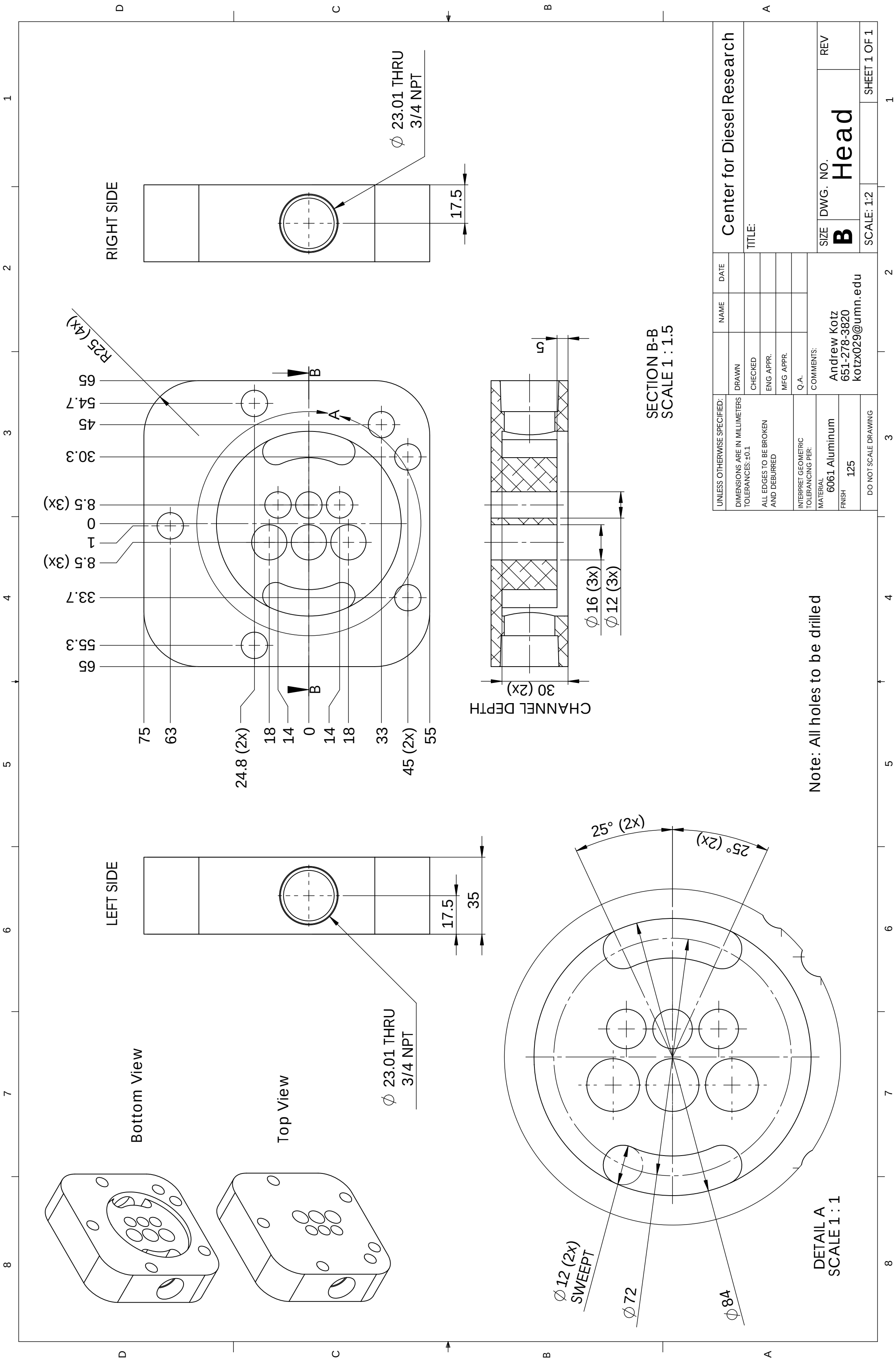
Table A.5: *Tabulated results for the high temperature 300 rpm test condition. Blank entries indicate there were not enough quality signals received to process.*

Trigger	Sample	Signals Processed	$\bar{x}_{measured}$ [K]	Relative $SE_{\bar{x}_{meas.}}$	$\bar{x}_{predicted}$ [K]	Relative $SE_{\bar{x}_{pred.}}$	$\epsilon_{\%}$
0 CAD	**1**	0	-	-	-	-	-
	2	0	-	-	-	-	-
45 CAD	**1**	30	353.05	0.044%	336.35	0.29%	4.7%
	2	1	352.48	-	333.51	-	5.4%
90 CAD	**1**	12	356.68	0.039%	335.96	0.31%	5.8%
	2	41	352.50	0.0081%	336.21	0.11%	4.6%
135 CAD	**1**	0	-	-	-	-	-
	2	0	-	-	-	-	-
180 CAD	**1**	6	355.12	0.031%	331.87	0.29%	6.5%
	2	50	353.39	0.027%	335.50	0.12%	5.1%
	3	34	354.07	0.085%	338.59	0.14%	4.4%
	4	39	352.63	0.013%	326.97	0.23%	7.3%
225 CAD	**1**	0	-	-	-	-	-
	2	0	-	-	-	-	-
270 CAD	**1**	39	354.46	0.039%	342.78	0.056%	3.3%
	2	20	352.85	0.0082%	345.77	0.081%	2.0%
315 CAD	**1**	7	352.25	0.011%	335.53	0.16%	4.7%
	2	6	354.04	0.034%	341.80	0.29%	3.5%
360 CAD	**1**	1	353.46	-	445.98	-	26%
	2	4	354.05	0.015%	407.37	0.41%	15%

Table A.6: *Tabulated results for the high temperature 1200 rpm test condition. Blank entries indicate there were not enough quality signals received to process.*

Trigger	Sample	Signals Processed	$\bar{x}_{measured}$ [K]	Relative $SE_{\bar{x}_{meas.}}$	$\bar{x}_{predicted}$ [K]	Relative $SE_{\bar{x}_{pred.}}$	$\epsilon_{\%}$
0 CAD	**1**	0	-	-	-	-	-
	2	7	351.52	0.023%	365.53	0.25%	4.0%
	3	7	354.47	0.10%	376.91	0.31%	6.3%
45 CAD	**1**	37	354.11	0.053%	308.75	0.15%	13%
	2	34	354.38	0.0045%	310.33	0.13%	12%
90 CAD	**1**	0	-	-	-	-	-
	2	1	354.94	-	302.38	-	15%
	3	1	352.38	-	313.70	-	11%
	4	0	-	-	-	-	-
135 CAD	**1**	0	-	-	-	-	-
	2	0	-	-	-	-	-
180 CAD	**1**	0	-	-	-	-	-
	2	0	-	-	-	-	-
	3	0	-	-	-	-	-
	4	8	353.52	0.040%	346.34	0.27%	2.0%
	5	0	-	-	-	-	-
	6	0	-	-	-	-	-
	7	39	354.08	0.012%	350.74	0.12%	0.94%
	8	9	351.79	0.0082%	351.71	0.24%	0.024%
	9	14	352.99	0.0087%	351.17	0.30%	0.52%
	10	21	352.06	0.0096%	354.45	0.22%	0.68%
	11	18	352.83	0.017%	350.55	0.19%	0.65%
	12	14	353.84	0.022%	361.16	0.39%	2.1%
225 CAD	**1**	0	-	-	-	-	-
	2	0	-	-	-	-	-
270 CAD	**1**	0	-	-	-	-	-
	2	3	355.82	0.024%	378.09	1.4%	6.3%
	3	33	352.79	0.014%	376.31	0.096%	6.7%
	4	47	353.34	0.0069%	383.74	0.054%	8.6%
315 CAD	**1**	33	353.68	0.0056%	401.03	0.22%	13%
	2	48	353.85	0.0072%	406.44	0.10%	15%
360 CAD	**1**	0	-	-	-	-	-
	2	2	353.68	0.037%	367.32	0.098%	3.9%
	3	8	354.70	0.023%	373.62	0.12%	5.3%

B Modified engine head


RIGHT SIDE
⌀ 23.01 THRU
3/4 NPT
17.5
R25 (4x)
65
54.7
45
30.3
8.5 (3x)
0
1
8.5 (3x)
33.7
55.3
65
75
63
24.8 (2x)
18
14
0
14
18
33
45 (2x)
55
CHANNEL DEPTH
30 (2x)
5
⌀ 16 (3x)
⌀ 12 (3x)
SECTION B-B
SCALE 1 : 1.5
Note: All holes to be drilled
LEFT SIDE
17.5
35
⌀ 23.01 THRU
3/4 NPT
Bottom View
Top View
25° (2x)
25° (2x)
⌀ 12 (2x)
SWEEPT
⌀ 72
⌀ 84
DETAIL A
SCALE 1 : 1
Center for Diesel Research
TITLE:
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
TOLERANCES: ±0.1
ALL EDGES TO BE BROKEN AND DEBURRED
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
6061 Aluminum
FINISH
125
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
Andrew Kotz
651-278-3820
kotzx029@umn.edu
SIZE
B
DWG. NO.
Head
REV
SCALE: 1:2
SHEET 1 OF 1

C Matlab code

Parse Signal

```
function [ Signal,Time,Trig,CAD,PistLoc,PistTime,Temp,NomTOF ]...
    = ParseSignal( RawData,signal )
%PARSE SIGNAL Parses the raw data file signal

j=signal*2-1;

Signal = RawData(j,5:end);
Signal = Signal(1:2000);

DataPts = length(Signal);

ts = RawData(j,4);

Time = 0:ts:ts*(DataPts-1);
Time = (Time+RawData(j,3))*1000; %ms

CAD = CADcalc(RawData(j+1,:));

[PistLoc,PistTime] = PistonLocation(CAD);

Temp = RawData(j,1);
Trig = CAD(1,1);

NomTOF = NominalTOF(Temp,Trig);

end
```

Published with MATLAB® R2015a

Piston Location

```
function [ PosOut,TIME ] = PistonLocation( CAD )
%PISTON LOCATION uses crank angle data to calculate piston position vs
 time
%information

timeStep = 5e-7;
[ConRod,CrankRad,Clearance,WPO] = EngineParameters();


xMax = ConRod+CrankRad; %[m] maximum extension of piston

%CADad = zeros(1,2000);

CADad(1,1) = CAD(1,1);
TIME(1,1) = 0;

i=2;
j=0;
%k=0;
w=1;
y=2;
while i<=2000;
    j=j+1;
    if CAD(1,i)~=CAD(1,i-1)
        step = timeStep*j;
        CADad(1,y) = CAD(1,i);
        TIME(1,y) = TIME(1,y-1)+step;
        y=y+1;
        j=0;
    end
    i=i+1;
end


x = CrankRad*cosd(CADad) + sqrt(ConRod^2-(CrankRad*sind(CADad)-
WPO).^2); %piston extension
PosOut = (xMax-x)+Clearance; %Path Length

if length(PosOut)>1;
    PosOut = PosOut(1,2:end);
    TIME = TIME(1,2:end)*1000;
else
    PosOut(1,1:2000) = PosOut;
    TIME = 0:timeStep:timeStep*(2000-1);
    TIME = TIME*1000;
end

end
```

Published with MATLAB® R2015a

Signal Selection

```
function [ StrucOUT ] = SlopeSelectSignals( strucIN )
%SLOPE SELECT SIGNALS Selects the echo that will later be processed
%   Code selects the signal based on theoretical time of flight and
 test
%   conditions. The signal is kept if the echo meets certain slope
 criteria
%   on the rising edge of the echo.

manual = 0; % 1 for manual, 2 for auto

minTime = 0.35; %[ms] start time to look for signal


fields = fieldnames(strucIN);
[numFields,~] = size(fields);
j = 1;
k=1;
for i = 1:numFields;
    testNum = char(fields(i,1));
    message = ['^^^^^^Test Number ',testNum,'^^^^^^'];
    disp(message);
    origData = strucIN.(testNum).data;
    Data = strucIN.(testNum).data;
    [numSignals,~] = size(Data);
    numSignals = numSignals/2;
    SlopeDiscardData = zeros(size(Data));
    Header = strucIN.(testNum).textdata;
    nKeep = 0;
    Slopes = zeros(numSignals,1);
    TriggersOUT = zeros(numSignals,1);
    TempsOUT = zeros(numSignals,1);
    StartInd = zeros(numSignals,1);
    NomTOFout = zeros(numSignals,1);
    NomFDout = zeros(numSignals,1);
    AdjSignumTOFout = zeros(numSignals,1);
    AdjPistMoveTOFout = zeros(numSignals,1);
    AdjSignumFDout = zeros(numSignals,1);
    AdjPistMoveFDout = zeros(numSignals,1);
    Adj = zeros(numSignals,1);
    SignalOUT = zeros(numSignals,2000);
    TimeOUT = zeros(numSignals,2000);
    PistonTimeOUT = zeros(numSignals,2000);
    PistonLocationOUT = zeros(numSignals,2000);
    CADout = zeros(numSignals,2000);
    AllTOFout = zeros(numSignals,20);
    MinMaxTOFout = zeros(numSignals,2);

    while j <numSignals*2;
        signum = (j+1)/2;
        [signal,time,Trig,cad,PistLoc,PistTime,Temp,NomTOF] =...
```

```
    ParseSignal(Data,k);
DataPts = length(signal);
pistlength = length(PistLoc);


NomFD = NominalFD(Trig);

[AdjsignumTOF,adj] = AdjustTOF(NomTOF,minTime);
AdjsignumFD = NomFD*adj;
[Alltof,zers] = AllTOFs(NomTOF);
alltofLength = length(Alltof);
AdjPistMoveFD = ...
    AdjFDpistMove(PistTime,PistLoc,AdjsignumFD,AdjsignumTOF);

AdjPistMoveTOF = TimeOfFlight(Temp,AdjPistMoveFD);

[~,TOFind] = min(abs(time-AdjsignumTOF));

LPFsig = LP_Filter(signal,800); %low pass filter 800 kHz
hilbSig = (abs(hilbert(LPFsig))); %hilbert transform
sgHilb = sgolayfilt(hilbSig,4,101);
    %sovitsky-golay smoothing filter
nsgHilb = -sgHilb;

diff = Alltof-minTime;
ipos = find(diff>0);
[~,TOFstart] = min(ipos);
TOFstart = ipos(TOFstart);
TOFend = TOFstart+1;

[~,SearchStart] = min(abs(time-Alltof(TOFstart)));
[~,SearchEnd] = min(abs(time-Alltof(TOFend)));

slope = Slope(sgHilb(SearchStart:SearchEnd),10);
    %slope of signal

maxSlope = max(slope);
[minTOF,maxTOF] = ...
    extremesTOF(Temp+273.15,cad(1,1),cad(1,end));


%plot(Time,LPFsig,Time(SearchStart:SearchEnd), ...
%sgHilb(SearchStart:SearchEnd),Time,nsgHilb, ...
%Alltof,zers,'b.','MarkerSize',15);
%{
figure(1);
SigFig = subplot(2,1,2);
PistFig = subplot(2,1,1);
plot(SigFig,Time,LPFsig,Time(SearchStart:SearchEnd), ...
sgHilb(SearchStart:SearchEnd),Time,nsgHilb,Alltof, ...
zers,'b.','MarkerSize',15);
xlimit = xlim;
plot(PistFig,PistTime,PistLoc);
xlim(gca,xlimit);
```

```
figure, plot(Time(SearchStart:SearchEnd-10),slope);
%}
if manual==1
    signum = num2str(k);
    prompt = ['Keep signal ',signum,'[1=yes,2=no]?'];
    save = input(prompt);
    if maxSlope>0.004;
        disp('Auto Keep');
    else
        disp('No Keep');
    end
    slop = num2str(maxSlope);
        disp(['Slope=',slop]);
    if save==2
        Data(j,:)=0;
        Data(j+1,:)=0;
        %reject(k,1:3) = [len,mid,ht];
    elseif isempty(save)
        disp('You did not type anything');
        k=k-1;
        j=j-2;
    elseif save~=1
        disp('Bad Entry, Try Again');
        k=k-1;
        j=j-2;
    else
        %saved(k,1:3) = [len,mid,ht];
    end
else
    %if criteria met, keep data, else clear data
    Slopes(k,1) = maxSlope;
    if maxSlope>0.004 %slope criteria on echo
        nKeep = nKeep+1;
        [~,stOS] = max(sgHilb(SearchStart:SearchEnd));
        StartInd(k,1) = SearchStart-1+stOS;
        SlopeDiscardData(j,:)=0;
        SlopeDiscardData(j+1,:)=0;
        TriggersOUT(signum,1) = Trig;
        TempsOUT(signum,1) = Temp;
        NomTOFout(signum,1) = NomTOF;
        NomFDout(signum,1) = NomFD;
        AdjSignumTOFout(signum,1) = AdjsignumTOF;
        AdjSignumFDout(signum,1) = AdjsignumFD;
        AdjPistMoveTOFout(signum,1) = AdjPistMoveTOF;
        AdjPistMoveFDout(signum,1) = AdjPistMoveFD;
        Adj(signum,1) = adj;
        SignalOUT(signum,:) = signal;
        TimeOUT(signum,:) = time;
        PistonTimeOUT(signum,1:pistlength) = PistTime;
        PistonLocationOUT(signum,1:pistlength) = PistLoc;
        CADout(signum,:) = cad;
        AllTOFout(signum,1:alltofLength) = Alltof;
        MinMaxTOFout(signum,1) = minTOF;
        MinMaxTOFout(signum,2) = maxTOF;
```

```
                else
                    SlopeDiscardData(j,:)=Data(j,:);
                    SlopeDiscardData(j+1,:)=Data(j+1,:);
                    Data(j,:)=0;
                    Data(j+1,:)=0;
                end

            end
            j = j+2;
            k=k+1;
        end
        j = 1;
        k=1;

        PistonTimeOUT = PistonTimeOUT(:,1:pistlength);
        PistonLocationOUT = PistonLocationOUT(:,1:pistlength);
        AllTOFout = AllTOFout(:,1:alltofLength);

        %AvgSlopes = mean(Slopes);
        StrucOUT.(testNum).header = Header;
        StrucOUT.(testNum).OriginalData = origData;
        StrucOUT.(testNum).data = Data;
        StrucOUT.(testNum).NumberSlopeKeptToProcess = nKeep;
        StrucOUT.(testNum).DiscardedSlopeData = SlopeDiscardData;
        StrucOUT.(testNum).Signal = SignalOUT;
        StrucOUT.(testNum).Time = TimeOUT;
        StrucOUT.(testNum).CAD = CADout;
        StrucOUT.(testNum).PistonLocation = PistonLocationOUT;
        StrucOUT.(testNum).PistonTime = PistonTimeOUT;
        StrucOUT.(testNum).Trigger = TriggersOUT;
        StrucOUT.(testNum).Temperature = TempsOUT;
        StrucOUT.(testNum).slopes = Slopes;
        %StrucOUT.(testNum).AvgSlope = AvgSlopes;
        StrucOUT.(testNum).StartProcessIndice = StartInd;
        StrucOUT.(testNum).AllTOF = AllTOFout;
        StrucOUT.(testNum).MinMaxTOF = MinMaxTOFout;
        StrucOUT.(testNum).NominalTOF = NomTOFout;
        StrucOUT.(testNum).NominalFD = NomFDout;
        StrucOUT.(testNum).AdjustedSignalTOF = AdjSignumTOFout;
        StrucOUT.(testNum).AdjustedSignalFD = AdjSignumFDout;
        StrucOUT.(testNum).AdjustedPistonMovementTOF = ...
            AdjPistMoveTOFout;
        StrucOUT.(testNum).AdjustedPistonMovementFD = ...
            AdjPistMoveFDout;
        StrucOUT.(testNum).EchoNumberAnalyzed = Adj;
    end

    end
```

Published with MATLAB® R2015a

Kalman Filter

```
function [ DataStructureOUT ] = KFauto( DataStructureIN )
%KF AUTO Automated Kalman filtering of data sets
%   using the data structure input, the data is parsed to isolate the
 echo
%   envelope. The signal is then itteratively fitted using the Kalman
%   filter.

fields = fieldnames(DataStructureIN);
[numFields,~] = size(fields);

for n=1:numFields
    testNum = char(fields(n,1));
    message = ['^^^^^^Test Number ',testNum,'^^^^^^'];
    disp(message);

    %carry-over data structure
    DataStructureOUT.(testNum) = DataStructureIN.(testNum);

    Data_sig = DataStructureIN.(testNum).data;
    DataOut = DataStructureIN.(testNum).data;
    [numSignals,~] = size(Data_sig);

    skipped = DataStructureIN.(testNum).data;
    skipCount = 0;
    FitParam = zeros(numSignals/2,4);

    for j = 1:2:numSignals;
        if Data_sig(j,1)~=0;
            sigNum = (j+1)/2;
            disp(['Processing Signal ', num2str(sigNum)]);
            signal = DataStructureIN.(testNum).Signal(sigNum,:);
            sigrand = rand(1,2000,'double');
            time = DataStructureIN.(testNum).Time(sigNum,:);

            %find data point to cut ringing by finding
            %closest to desired cut-time
            [~,cutRing] = min(abs(time-0.225));

            %Low-pass filter signal
            LPFsig = LP_Filter(signal,800); %low pass filter 800 kHz
            %LPFsig = LPFsig(1,:);

            %Hilbert transform data to give analytic signal
            HTsig = abs(hilbert(LPFsig));
            HTsig = sgolayfilt(HTsig,4,41);
            HTsig(1:cutRing) = 0;


            %Selecting echo signal envelope to fit with Kalman filter
            %Changing windowSz variable changes the width
            %windowSz x 2) of the echo envelope from the maximum point
```

```
pwindowSz = 40;
nwindowSz = 35;
%[~,echoMax] = max(HTsig);
echoMax = ...
    DataStructureIN. ...
    (testNum).StartProcessIndice(sigNum,1);
echoStart = echoMax-pwindowSz;
echoEnd = echoMax+nwindowSz;
if echoEnd > size(HTsig,2);
    echoEnd = size(HTsig,2);
end
echoSig = HTsig(1,echoStart:echoEnd);
echoTime = time(1,echoStart:echoEnd);
Np = length(echoSig);

%-------Setting up initial conditions------------

%x0 = [A0,alpa,T,tau] is the state vector
        %A0=echo Amp,
        %alpha and T are transducer dependent,
        %tau=desired time of flight

x0 = ...
    [0.41, 2.7,11e-6,...
    (echoTime(1,1)-echoTime(1,1)*0.05)/1000]; %work ok
%x0 = [0.31, 3.0, 9e-6,...
%   (echoTime(1,1)-echoTime(1,1)*0.05)/1000];
        %experimental


N = length(x0);
%P0 = [1e-1, 1e-15, 1e-15, 1e-15];
%P = diag(P0);
P = 1e-14*eye(N);

%-----------Begining of Kalman Filter-------------


%-----------Using Example Code----------------------

q=1e-14;    %std of process
r=0.0000001;    %std of measurement
Q=q^2*eye(N); % covariance of process
R=r^2;        % covariance of measurement
%kts=echoTime(1,1);

%h=@(x) x(1)*((kts-x(4))/x(3))^x(2)*exp(-(kts-x(4))/x(3));
    % nonlinear state equations
    %(anonymous function, function handle)
f=@(x1) x1; %identity function
s=x0';
x=s;

sV = zeros(N,Np);          %actual
```

```
                zV = zeros(1,Np);
                pV = zeros(N,Np);
                change = 1;
                skipCheck=0;

                xEnd(1:N,:) = 0;
                i=0;
                while change>=1e-4;
                    i=i+1;
                    xV = zeros(N,Np);          %estimate % allocate memory
                    for k=1:Np;
                        kts=echoTime(1,k)/1000;
                        h=@(x) x(1)*...
                            ((kts-x(4))/x(3))^x(2)*exp(-(kts-x(4))/x(3));
                        z=echoSig(1,k);
                        %sV(:,k) = s;
                        zV(k) = z;
                        [x,P,X,X1] = ukf(f,x,P,h,z,Q,R);
                        [~,cP] = chol(P);
                        if cP~=0;
                            skipCount = skipCount+1;
                            change=0.9e-4;
                            skipCheck = 1;
                            Mesg = ['Skipped ', num2str(sigNum)];
                            disp(Mesg);
                            DataStructureOUT.(testNum) = ...
                                ClearExcluded...
                                (DataStructureOUT.(testNum),sigNum);
                            break;
                        end
                        xV(:,k) = x;
                        s=f(s);
                    end
                    if skipCheck==0;
                        xEnd(:,i) = xV(:,Np);
                        change = max(abs(xV(:,1)-xV(:,Np)));
                    end
                end
                if skipCheck==0;
                    skipped(j,:) = 0;
                    skipped(j+1,:) = 0;
                    FitParam(sigNum,:) = x';
                end
            end
        end
        DataStructureOUT.(testNum).NumKFProcessed = ...
            numSignals/2-skipCount;
        DataStructureOUT.(testNum).SkippedKalnamFilterData = skipped;
        DataStructureOUT.(testNum).FitParameters = FitParam;
    end
    end
```

Published with MATLAB® R2015a

Unscented Kalman Filter

```
function [x,P,X,X1]=ukf(fstate,x,P,hmeas,z,Q,R)
% UKF   Unscented Kalman Filter for nonlinear dynamic systems
% [x, P] = ukf(f,x,P,h,z,Q,R) returns state estimate, x and state
 covariance, P
% for nonlinear dynamic system (for simplicity, noises are assumed as
 additive):
%           x_k+1 = f(x_k) + w_k
%           z_k   = h(x_k) + v_k
% where w ~ N(0,Q) meaning w is gaussian noise with covariance Q
%       v ~ N(0,R) meaning v is gaussian noise with covariance R
% Inputs:   f: function handle for f(x)
%           x: "a priori" state estimate
%           P: "a priori" estimated state covariance
%           h: fanction handle for h(x)
%           z: current measurement
%           Q: process noise covariance
%           R: measurement noise covariance
% Output:   x: "a posteriori" state estimate
%           P: "a posteriori" state covariance
%
% Example:
%{
n=3;      %number of state
q=0.1;    %std of process
r=0.1;    %std of measurement
Q=q^2*eye(n); % covariance of process
R=r^2;        % covariance of measurement
f=@(x)[x(2);x(3);0.05*x(1)*(x(2)+x(3))];  % nonlinear state equations
h=@(x)x(1);                               % measurement equation
s=[0;0;1];                                % initial state
x=s+q*randn(3,1); %initial state          % initial state with noise
P = eye(n);                               % initial state covraiance
N=20;                                     % total dynamic steps
xV = zeros(n,N);          %estmate        % allocate memory
sV = zeros(n,N);          %actual
zV = zeros(1,N);
for k=1:N
  z = h(s) + r*randn;                     % measurments
  sV(:,k)= s;                             % save actual state
  zV(k)  = z;                             % save measurment
  [x, P] = ukf(f,x,P,h,z,Q,R);            % ekf
  xV(:,k) = x;                            % save estimate
  s = f(s) + q*randn(3,1);                % update process
end
for k=1:3                                 % plot results
  subplot(3,1,k)
  plot(1:N, sV(k,:), '-', 1:N, xV(k,:), '--')
end
%}
% Reference: Julier, SJ. and Uhlmann, J.K., Unscented Filtering and
```

```
% Nonlinear Estimation, Proceedings of the IEEE, Vol. 92, No. 3,
% pp.401-422, 2004.
%
% By Yi Cao at Cranfield University, 04/01/2008
%
L=numel(x);                                  %numer of states
m=numel(z);                                  %numer of measurements
alpha=0.55;                          %default, tunable 0.9<alpha>0.5???
ki=0;                                        %default, tunable
beta=2;                                      %default, tunable
lambda=alpha^2*(L+ki)-L;                     %scaling factor
c=L+lambda;                                  %scaling factor
Wm=[lambda/c 0.5/c+zeros(1,2*L)];            %weights for means
Wc=Wm;
Wc(1)=Wc(1)+(1-alpha^2+beta);                %weights for covariance
c=sqrt(c);

[X]=sigmas(x,P,c);                             %sigma points around x
X;
[x1,X1,P1,X2]=ut(fstate,X,Wm,Wc,L,Q);
                            %unscented transformation of process
P1;
X2;
x1;
X1;
% X1=sigmas(x1,P1,c);                          %sigma points around x1
% X2=X1-x1(:,ones(1,size(X1,2)));              %deviation of X1
[z1,Z1,P2,Z2]=ut(hmeas,X1,Wm,Wc,m,R);
                            %unscented transformation of measurments
Z2;
P12=X2*diag(Wc)*Z2';                 %transformed cross-covariance
K=P12*inv(P2);
x=x1+K*(z-z1);                              %state update
diff=K*(z-z1);

P=P1-K*P12';                                 %covariance update


function [y,Y,P,Y1]=ut(f,X,Wm,Wc,n,R)
%Unscented Transformation
%Input:
%        f: nonlinear map
%        X: sigma points
%       Wm: weights for mean
%       Wc: weights for covraiance
%        n: numer of outputs of f
%        R: additive covariance
%Output:
%        y: transformed mean
%        Y: transformed smapling points
%        P: transformed covariance
%       Y1: transformed deviations

L=size(X,2);
```

```
y=zeros(n,1);
Y=zeros(n,L);
for k=1:L
    Y(:,k)=f(X(:,k));
    y=y+Wm(k)*Y(:,k);
end
Y1=Y-y(:,ones(1,L));
P=Y1*diag(Wc)*Y1'+R;

function [X]=sigmas(x,P,c)
%Sigma points around reference point
%Inputs:
%       x: reference point
%       P: covariance
%       c: coefficient
%Output:
%       X: Sigma points
%       chkP: is P positive definate (=0 implies yes)
[a,chkP]=chol(P);
if chkP~=0;
    disp('Uh-oh, trouble coming');
end
A = c*a';
Y = x(:,ones(1,numel(x)));
X = [x Y+A Y-A];
```

Published with MATLAB® R2015a

Calculate Results

```
function [ strucOUT ] = CalcResults( strucIN )
%CALC RESULTS calculates the results for the echo signals that were
 kept
%through the signal processing procedures.


fields = fieldnames(strucIN);
[numFields,~] = size(fields);

%PistLoc = zeros(50,2000);
%PistTime = zeros(50,2000);
k=0;
numSigs=0;
for j = 1:numFields;
    testNum = char(fields(j,1));
    message = ['^^^^^^Test Number ',testNum,'^^^^^^'];
    disp(message);
    strucOUT.(testNum) = strucIN.(testNum);
    [numSignals,~] = size(strucIN.(testNum).Signal);
    [DateStringOUT,DateNumOUT] = ...
        DateInfo(strucIN.(testNum).header(1,1));
    for i = 1:numSignals;

        if strucOUT.(testNum).FitParameters(i,:)~=0;
            numSigs = numSigs+1;
            strucOUT.(testNum).PredictedAdjSignalTemp(i,1) =...
                PredictTemp(strucOUT.(testNum). ...
                FitParameters(i,4)*1000,strucOUT. ...
                (testNum).AdjustedSignalFD(i,1));
            strucOUT.(testNum).PredictedAdjPistMoveTemp(i,1) =...
                PredictTemp(strucOUT.(testNum). ...
                FitParameters(i,4)*1000,strucOUT. ...
                (testNum).AdjustedPistonMovementFD(i,1));
            cad = strucOUT.(testNum).CAD(i,:);
            Temp = strucOUT.(testNum).Temperature(i,:);
            [minTOF,maxTOF] = ...
                extremesTOF(Temp+273.15,cad(1,1),cad(1,end));
            strucOUT.(testNum).MinMaxTOF(i,1) = minTOF;
            strucOUT.(testNum).MinMaxTOF(i,2) = maxTOF;

        end

    end
    if numSigs>0;
        strucOUT.(testNum).Results.numberSignals = numSigs;
        strucOUT.(testNum).Results.DateString = DateStringOUT;
        strucOUT.(testNum).Results.DateNumber = DateNumOUT;
        [Mean,stdev,SEM,relSEM] = StandardErrorOfMean...
            (nonzeros(strucOUT.(testNum).PredictedAdjSignalTemp));
        strucOUT.(testNum).Results.AdjSignalPredTemp_Avg = Mean;
```

```
Predicted = Mean;
strucOUT.(testNum).Results.AdjSignalPredTemp_Stdev = stdev;
strucOUT.(testNum).Results.AdjSignalPredTemp_SEM = SEM;
strucOUT.(testNum).Results.AdjSignalPredTemp_relSEM = relSEM;
[Uplim,Lowlim] = ConfidenceInterval95p...
    (nonzeros(strucOUT.(testNum).PredictedAdjSignalTemp));
strucOUT.(testNum).Results. ...
    AdjSignalPredTemp_CI95p(1,1) = Uplim;
strucOUT.(testNum).Results. ...
    AdjSignalPredTemp_CI95p(1,2) = Lowlim;

[Mean,stdev,SEM,relSEM] = StandardErrorOfMean...
    (nonzeros(strucOUT.(testNum).PredictedAdjPistMoveTemp));
strucOUT.(testNum).Results.AdjPistMovePredTemp_Avg = Mean;
strucOUT.(testNum).Results.AdjPistMovePredTemp_Stdev = stdev;
strucOUT.(testNum).Results.AdjPistMovePredTemp_SEM = SEM;
strucOUT.(testNum).Results. ...
    AdjPistMovePredTemp_relSEM = relSEM;
[Uplim,Lowlim] = ConfidenceInterval95p...
    (nonzeros(strucOUT.(testNum).PredictedAdjPistMoveTemp));
strucOUT.(testNum).Results. ...
    AdjPistMovePredTemp_CI95p(1,1) = Uplim;
strucOUT.(testNum).Results. ...
    AdjPistMovePredTemp_CI95p(1,2) = Lowlim;

[Mean,stdev,SEM,~] = StandardErrorOfMean...
    (nonzeros(strucOUT.(testNum).Temperature));
relSEM = (SEM/(Mean+273.15))*100;
strucOUT.(testNum).Results.MeasuredTemp_Avg = Mean;
strucOUT.(testNum).Results.MeasuredTempABS_Avg = Mean+273.15;
Measured = Mean+273.15;
strucOUT.(testNum).Results.MeasuredTemp_Stdev = stdev;
strucOUT.(testNum).Results.MeasuredTemp_SEM = SEM;
strucOUT.(testNum).Results.MeasuredTemp_relSEM = relSEM;
[Uplim,Lowlim] = ConfidenceInterval95p...
    (nonzeros(strucOUT.(testNum).Temperature));
strucOUT.(testNum).Results.MeasuredTemp_CI95p(1,1) = Uplim;
strucOUT.(testNum).Results.MeasuredTemp_CI95p(1,2) = Lowlim;

error = abs((Predicted-Measured)/Measured)*100;

strucOUT.(testNum).Results.PercentError = error;

[Mean,stdev,SEM,~] = StandardErrorOfMean...
    (nonzeros(strucOUT.(testNum).FitParameters(:,4))*1000);
strucOUT.(testNum).Results.ExpTOF_Avg = Mean;
strucOUT.(testNum).Results.ExpTOF_Stdev = stdev;
strucOUT.(testNum).Results.ExpTOF_SEM = SEM;
[Uplim,Lowlim] = ConfidenceInterval95p...
    (nonzeros(strucOUT.(testNum).FitParameters(:,4))*1000);
strucOUT.(testNum).Results.ExpTOF_CI95p(1,1) = Uplim;
strucOUT.(testNum).Results.ExpTOF_CI95p(1,2) = Lowlim;

[Mean,stdev,SEM,~] = StandardErrorOfMean...
```

```
    (nonzeros(strucOUT.(testNum).AdjustedSignalTOF));
strucOUT.(testNum).Results.AdjSignalTOF_Avg = Mean;
strucOUT.(testNum).Results.AdjSignalTOF_Stdev = stdev;
strucOUT.(testNum).Results.AdjSignalTOF_SEM = SEM;
[Uplim,Lowlim] = ConfidenceInterval95p...
    (nonzeros(strucOUT.(testNum).AdjustedSignalTOF));
strucOUT.(testNum).Results.AdjSignalTOF_CI95p(1,1) = Uplim;
strucOUT.(testNum).Results.AdjSignalTOF_CI95p(1,2) = Lowlim;

[Mean,stdev,SEM,~] = StandardErrorOfMean...
    (nonzeros(strucOUT.(testNum).AdjustedSignalFD));
strucOUT.(testNum).Results.AdjSignalFD_Avg = Mean;
strucOUT.(testNum).Results.AdjSignalFD_Stdev = stdev;
strucOUT.(testNum).Results.AdjSignalFD_SEM = SEM;
[Uplim,Lowlim] = ConfidenceInterval95p...
    (nonzeros(strucOUT.(testNum).AdjustedSignalFD));
strucOUT.(testNum).Results.AdjSignalFD_CI95p(1,1) = Uplim;
strucOUT.(testNum).Results.AdjSignalFD_CI95p(1,2) = Lowlim;

[Mean,stdev,SEM,~] = StandardErrorOfMean...
    (nonzeros(strucOUT.(testNum).AdjustedPistonMovementTOF));
strucOUT.(testNum).Results.AdjPistMoveTOF_Avg = Mean;
strucOUT.(testNum).Results.AdjPistMoveTOF_Stdev = stdev;
strucOUT.(testNum).Results.AdjPistMoveTOF_SEM = SEM;
[Uplim,Lowlim] = ConfidenceInterval95p...
    (nonzeros(strucOUT.(testNum).AdjustedPistonMovementTOF));
strucOUT.(testNum).Results.AdjPistMoveTOF_CI95p(1,1) = Uplim;
strucOUT.(testNum).Results.AdjPistMoveTOF_CI95p(1,2) = Lowlim;

[Mean,stdev,SEM,~] = StandardErrorOfMean...
    (nonzeros(strucOUT.(testNum).AdjustedPistonMovementFD));
strucOUT.(testNum).Results.AdjPistMoveFD_Avg = Mean;
strucOUT.(testNum).Results.AdjPistMoveFD_Stdev = stdev;
strucOUT.(testNum).Results.AdjPistMoveFD_SEM = SEM;
[Uplim,Lowlim] = ConfidenceInterval95p...
    (nonzeros(strucOUT.(testNum).AdjustedPistonMovementFD));
strucOUT.(testNum).Results.AdjPistMoveFD_CI95p(1,1) = Uplim;
strucOUT.(testNum).Results.AdjPistMoveFD_CI95p(1,2) = Lowlim;

[Mean,stdev,SEM,~] = StandardErrorOfMean...
    (nonzeros(strucOUT.(testNum).NominalTOF));
strucOUT.(testNum).Results.NomTOF_avg = Mean;
strucOUT.(testNum).Results.NomTOF_Stdev = stdev;
strucOUT.(testNum).Results.NomTOF_SEM = SEM;
[Uplim,Lowlim] = ConfidenceInterval95p...
    (nonzeros(strucOUT.(testNum).NominalTOF));
strucOUT.(testNum).Results.NomTOF_CI95p(1,1) = Uplim;
strucOUT.(testNum).Results.NomTOF_CI95p(1,2) = Lowlim;

[Mean,stdev,SEM,~] = StandardErrorOfMean...
    (nonzeros(strucOUT.(testNum).NominalFD));
strucOUT.(testNum).Results.NomFD_Avg = Mean;
strucOUT.(testNum).Results.NomFD_Stdev = stdev;
strucOUT.(testNum).Results.NomFD_SEM = SEM;
```

```
        [Uplim,Lowlim] = ConfidenceInterval95p...
            (nonzeros(strucOUT.(testNum).NominalFD));
        strucOUT.(testNum).Results.NomFD_CI95p(1,1) = Uplim;
        strucOUT.(testNum).Results.NomFD_CI95p(1,2) = Lowlim;

        [Mean,stdev,~,~] = StandardErrorOfMean...
            (nonzeros(strucOUT.(testNum).MinMaxTOF(:,1)));
        strucOUT.(testNum).Results.MinMaxTOF_Avg(1,1) = Mean;
        strucOUT.(testNum).Results.MinMaxTOF_Stdev(1,1) = stdev;

        [Mean,stdev,~,~] = StandardErrorOfMean...
            (nonzeros(strucOUT.(testNum).MinMaxTOF(:,2)));
        strucOUT.(testNum).Results.MinMaxTOF_Avg(1,2) = Mean;
        strucOUT.(testNum).Results.MinMaxTOF_Stdev(1,2) = stdev;

        [Mean,stdev,~,~] = StandardErrorOfMean...
            (nonzeros(strucOUT.(testNum).Trigger));
        strucOUT.(testNum).Results.Trigger_Avg = Mean;
        strucOUT.(testNum).Results.Trigger_Stdev = stdev;


    else
        strucOUT.(testNum).Results.numberSignals = numSigs;
    end
k = 0;
numSigs = 0;
end

end
```

Published with MATLAB® R2015a