ADVANCES IN DEVELOPING A SIGNAL PROCESSING TOOL FOR ROCKET MOTOR MEASUREMENTS

Ye Peter Chen
Michelle Gaudiose
Ryan Murphy
Laura Schrader
Reza Seirafi
K. Preston White, Jr.

Systems and Information Engineering
University of Virginia
Charlottesville, VA 22903 USA
kpwhite@virginia.edu

P. K. Wu
Aerojet Corporation
5945 Wellington Road
Gainesville, VA 20155 USA
PK.Wu@aerojet.com

ABSTRACT

Solid rocket propellants are critical components in weapons, automobile airbag, and satellite technologies. The instantaneous rate at which a propellant burns is an essential design parameter in all of these applications. While existing formulas accurately predict burn rates for most solid engine propellants, these standard equations assume that the burn is homogeneous along an axis parallel to the burning surface. This assumption is not valid for solid fuel ramjets (SFRJ) and certain advanced solid rocket motors (SRM). For this reason, Aerojet Corporation is developing a novel ultrasound technology to measure burn rate directly during live-fire tests. This paper describes recent progress on the design of algorithms to identify the location of the burning propellant surface from noisy ultrasound time-series data. The key challenges are to identify the ultrasound echo corresponding to the burning surface for each measurement pulse and to isolate a single point within this echo which is consistent across pulses. We describe a new algorithm for this purpose. Using data sets from both live-fire and laboratory experiments, we compare predictions based on this algorithm to those based on alternate approaches. Also described is the redesign of a prototype human-computer interface that is used to record and display burn rate data.

1 INTRODUCTION

Aerojet Corporation—a major aerospace/defense contractor specializing in missile and space propulsion—is developing a non-intrusive measurement system to calculate the non-homogeneous burn rate of solid fuel ramjets (SFRJ) and solid fuel rocket motors (SRM). Solid rocket propellants are critical components in the weapons, airbag, and satellite technologies. The rate and pattern of propellant burn is crucial to ballistic motor performance and, therefore, to motor design. Predicting burn rates in SRM and SFRJ is more difficult than in many other applications. This is because the burn rate not homogenous, and is influence by factors such as the flow-field, edge effects, and density.

Figure 1 depicts the architecture of a SRM. Upon ignition of the exposed propellant surface, the chamber pressure rises and gasses are emitted through the nozzle, resulting in decreasing pressure and increasing velocity of the rocket. The rocket thrust produced is proportional to the mass flow rate and exit velocity of chamber gases. Thrust varies in proportion to the propellant combustion rate and, if thrust can be accurately determined, the efficiency of the SFRJs and SRMs can be predicted more precisely. Aerojet is experimenting with an ultrasonic measurement technique to measure directly the burn rate of these motors.
Prior related work on ultrasound measurement of solid-rocket propellant burn rate is reported by DiSalvo, et al. (1999), Hafenrichter, et al. (2002), Hori, et al. (2000), Ivey, et al. (2002), Ivey, et al. (2003), and Wu, et al. (2003). The UT has two signal components: (1) a transducer that emits ultrasonic signals, and (2) a receiver that captures and records the reflected acoustic signals from the layered surfaces within the rocket. An example of a reflective layer is the regressing surface of the burning solid propellant. Using this technique, ultrasonic pulses are emitted at a constant rate into the rocket motor over the course of its burn. Each pulse gives a snapshot of the solid fuel’s web thickness (the amount of propellant remaining) at that point in time.

Figure 2 depicts the results of UT recording for a single measurement pulse. As can be seen, part of the ultrasonic signal is reflected at each of the rocket motor’s layers, returning to the receiver. The remainder of the signal continues through the medium, reflecting off of other rocket layers (Ivey et al., 2003).

Figure 2. Signal reflections off of acoustic impedances correlated to an ultrasonic signal (Wu et al., 2003)

By determining the web thickness from each of these recorded signals, the burn rate is then determined as the slope of the web thickness over time. The primary technical challenge is to distinguish and track the reflection corresponding to the regressing propellant surface, amidst noise and the other reflections, in order to perform this calculation. This UT was used by Aerojet and a University of Virginia Capstone team during the 2002-2003 academic year in Phase I of this project, in an attempt to overcome this technical challenge.


The goal of Phase I of this project was to calculate the average burn rate of a simulated rocket burn. The project team devised a method employing Reverse High Dynamic Range Compression (RHDRC) for this purpose. This section summarizes RHDRC. A more detailed description of the algorithm can be found in Ivey et al. (2003). In general the algorithm comprises two main parts:

1. A semi-wave integration is applied which transforms the original signal, amplifying the echoes in the reflected pulse by exploiting both frequency and amplitude components. The integration results in each semi-wave being represented by single amplitude value at a single time.
2. A thresholding function is applied to the semi-waves resulting from step (1). The threshold is set just below the value of the highest amplitude of the smallest echo that should remain in the transformed signal. Amplitudes below that threshold value are set equal to zero, and are characterized as noise.

Applying these two steps reduces the discrete data signal from (approximately) 2,500 to 250 data points, where the points are clustered into distinct groups, representing the ultrasonic echoes or wave fronts. Using this algorithm, each wave front is characterized by its median value. By differencing the time occurrence of wave fronts across the files, the distance that each wave moves with respect to time can be calculated. The burn rate is determine directly from this calculation.

4 LIMITATIONS OF THE PRIOR PROTOTYPE

While the results of Phase I were promising, limitations experienced in using the RDHRC prototype exposed the need for further research and potential enhancement. Therefore, the first major step in completing the current project was to gain an understanding of why the RHDRC algorithm did not include all necessary functionality. The performance of the primary components of the prototype, the GUI and RHDRC algorithm, was evaluated in order to determine the major weaknesses. During preliminary tests, several major shortcomings became apparent. These areas were addressed during the development of the Maximum Algorithm and corresponding GUI. The following two subsections outline the major shortcomings in both Phase I prototype components.

4.1 GUI Shortcomings

The GUI was tested on data collected by the 2002-2003 Capstone team from a water experiment, which is discussed in Section 5 of this paper. The following bullet list details the GUI shortcomings.

- **Plot axes.** All of the plots in the GUI—the main window plot, threshold calculation plot, clustering function plot, and wave front burn rate plot lack labeled axes. This makes it difficult to understand what is being displayed and how the plots should be interpreted. For example, the plot of individual wave front burn rates in the burn rate calculation window has an x-axis...
Since the RHDRC algorithm was created using the simple water experiment data as its basis, testing the algorithm on the complex live rocket burn data allowed for the major shortcomings of the RHDRC algorithm to be identified. The following list details the shortcomings of the RHDRC algorithm.

- **Noise filtering.** The algorithm’s first step is to eliminate noise from each of the ultrasonic pulse files. This step requires the user to input a threshold below which all points in the burn will be set to zero. On the live burn data, the appropriate value to select for the threshold is not obvious a priori. In contrast to the water experiment data, the wave reflection that must be tracked for burn rate determination is significantly smaller than the other reflections, and thus will be filtered out as noise unless a very small threshold is chosen. Choosing a small threshold, however, will result in the inclusion of unwanted noise. Using trial and error, a threshold of 50 mV seems to include the target wave reflection corresponding to the solid propellant. Unfortunately, as hypothesized, this threshold also passes much unwanted noise at the beginning of the reflected pulse.

- **Clustering.** The next step in the RHDRC algorithm is to cluster the results of the threshold calculation according to the median of each separate cluster. The use of a low threshold should cause the number of clusters to be large. Again, it is not entirely intuitive as to what input value should be used as input for the zero buffer length. After trial and error, the minimum zero buffer length that can be used is determined to be 46 mV—any value less than 46 mV causes the program to crash. With the live rocket data, the lowest threshold only creates clusters for the original pulse and the first two reflections. The rest of the reflections, most importantly, the echo corresponding to the regressing solid fuel, are left out of the algorithm. In effect, the failure to include the solid fuel reflection cluster(s) renders the algorithm and GUI useless on the live rocket burn data.

Thus, the analysis of the RHDRC algorithm on live rocket motor data evinced several key issues. First, there is an overall lack of clarity associated with each step of the algorithm; even though the GUI helps to make some sense of each step, it is still unclear as to exactly what is happening while the algorithm proceeds in the background. Furthermore, it has shown that the RHDRC algorithm is unable to handle the complexity of the live rocket burn data set. For these reasons, the 2003-2004 team developed a new algorithm and improved the GUI.

### 5 DEVELOPMENT OF THE MAXIMUM ALGORITHM

Before creating the algorithm, an investigation of the live rocket burn data was performed. This investigation allowed for a better understanding of what was actually happening to the regressing solid fuel with each timed ultrasonic pulse. Several interesting behaviors of the data were found, including the inconsistency of the ultrasonic trans-
ducer’s power over time. Algorithm possibilities were explored in each of these areas, but ultimately, the use of a simple new GUI feature led to the development of the Maximum Algorithm. This GUI feature involved the animation of the 549 ultrasonic pulse reflection files that recorded the live rocket burn. Since each ultrasonic pulse was emitted at a set time interval after its predecessor, graphing the files chronologically on a single graph with a slight delay between each update allowed for a visual representation of the live rocket burn. The graphical animation of the ultrasonic pulse reflections illustrated a very important facet of the data: the regressing fuel’s reflection wave, even though shifting with time, maintained its maximum amplitude at the same point within the reflection. This discovery proved to be consistent with all other sets of data that were tested, including the water experiment data from the previous year.

The primary technical challenge for this project was to determine the burn rate of a solid fuel rocket motor by tracking the wave reflection associated with the regressing solid fuel. With the aforementioned finding, a relatively simple way of tracking the reflection wave was discovered; specifically, tracking the maximum amplitude point within the target echo over the course of a burn. This simple, one-step approach contrasts the RHDRC algorithm that uses several more complicated steps. While the concept of tracking the maximum is straightforward, the actual algorithm is a little more complex, as explained below.

5.1.1 Explanation of the Maximum Algorithm

The Maximum Algorithm is described through the following series of steps:

1. **Range specification.** Before the algorithm can execute, the user must specify two separate time value ranges. This step is relatively straightforward, and with the use of the GUI, the user can actually visualize the ranges on a graph, with the ability to update until the desired values are defined.
   a. **Regressing solid fuel reflection wave range.** This range should be the time at which the first part of the reflection of the targeted echo returns to the transducer from the very first ultrasonic pulse and the time at which the last part of the reflection of the targeted echo returns to the transducer from the very first ultrasonic pulse.
   b. **Stationary wave range.** This range should be the time at which the first part of the closest stationary echo to the echo specified in part a returns to the transducer from the very first ultrasonic pulse, and the time at which the last part of its reflection returns to the transducer from the very first ultrasonic pulse.

2. **Determination of the Maximum amplitude.** Once the ranges have been specified for the first file, the time that the maximum amplitude of the target echo occurs in that range is determined and stored in a matrix.

3. **Determination of the Maximum amplitude within the next file.** Inspecting the next file (in chronological order), the amplitude at the stored time value from the previous file is then compared to the amplitude of the new file at the same time value.
   a. If the values are the same, the targeted wave has not moved since the last ultrasonic pulse, and the stored value remains the same. The algorithm then returns to step 2.
   b. If the values are different, the targeted wave has moved. The original time range is then modified to reflect the movement of the wave. The range is shifted by the amount that the maximum amplitude has moved. The stored value for the file then becomes the time at which the maximum amplitude occurs within the new time range. The algorithm then returns to step 2.

4. **Recursion.** The recursive process of going between steps 2 and 3 continues until the last file has been reached.

5. **Calculation of Burn Rate.** The matrix of stored time values serves as a tracker of the targeted wave over the course of the burn. Using the speed of sound, the time representation of the stationary wave, and the time value of the maximum amplitude, the instantaneous thickness of the solid fuel can be determined using the following equation:

\[
wb = c \times \Delta t / 2
\]  

where \(wb\) is the instantaneous web thickness, \(c\) is the speed of sound, and \(t\) is the time difference between the progressing echo and the moving echo. The derivative of this thickness is the instantaneous combustion rate. Since the ultrasonic measurement is discreet instead of continuous, the instantaneous combustion rate can be calculated using following equation:

\[
r = \Delta wb / \Delta T
\]

where \(r\) is the combustion rate, \(\Delta wb\) is the difference in the web thickness between two consecutive signals, and \(\Delta T\) is the signal interval time.

6 EVALUATING THE PERFORMANCE OF THE MAXIMUM ALGORITHM

Once the algorithm was created and implemented in a modified GUI, it was tested on three extensive data sets from: (1) a live rocket motor burn, (2) a simple water experiment conducted by the 2002-2003 Capstone team, and (3) a similar, more complex water experiment conducted by Aerojet in 2004.

Figure 3 displays a graphical result of the Maximum Algorithm. The line represents the propellant web thickness with respect to time for the live rocket motor burn.
Traditional methods of measuring web thickness are not sufficiently accurate, which is why the actual web thickness is not known for the live motor burn. As a result, the accuracy of the live burn results cannot be validated. A comparison between the animation of the burn and the web thickness plot indicates that the Maximum Algorithm result parallels the nature of the burn.

Figure 3. Propellant Thickness vs. Time for Live Burn Data

Since the results of the live rocket burn cannot be verified, the two aforementioned experiments were performed in order to create data sets with known burn rates. The Maximum Algorithm was validated against both of these experimental data sets by comparing the resulting burn rate to the known burn rates. The following subsection details these experiments.

6.1 Description of Water Experiments

Both water experiments consisted of a non-metallic boot and water, in the place of solid fuel propellant. The more complex experiment conducted by Aerojet incorporated a non-metallic adapter in order to add additional stationary echoes. In both experiments, water was released from the boot at a constant rate and a digital scale was used to collect weight measurements every 0.2 seconds. At the same time, the UT was also utilized to record the flow, or simulated “burn,” of the water out of the boot. The water release rates produced by the scale were converted into a “burn rate”, a means of comparison amongst burn rate results computed from UT analyzing methods.

For the simple water setup, three experiments were performed with ten trials each, varying the frequency as well as the strength of the signal. The first experiment used a frequency of 0.6 MHz and a gain of 30 dB, the second experiment used a frequency of 0.6 MHz and a gain of 2 dB, and the third experiment used a frequency of 0.3 MHz and a gain of 30 dB. For the more complex water setup, three trials were performed with a constant frequency and gain. Using the Maximum Algorithm, average burn rates were computed for each trial and compared to the results of the scale data.

6.2 Results for the Maximum Algorithm

After the average burn rate was calculated for each trial, the percent difference between the Maximum Algorithm and the scale data was calculated for each of the trials in both experiments. Table 1 presents the results for the simple water experiment and Table 2 presents the results for the more complex water experiment.

Table 1. Percent difference between Maximum Algorithm and scale data for the simple water experiment

<table>
<thead>
<tr>
<th>Mean % Difference</th>
<th>Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal H₂O</td>
<td>2.074</td>
</tr>
<tr>
<td>Low Gain</td>
<td>2.524</td>
</tr>
<tr>
<td>Low Freq</td>
<td>2.293</td>
</tr>
</tbody>
</table>

The results are unintuitive. Despite the limited number of trials, the Maximum Algorithm was more accurate and precise on the complex water experiment data. This suggests that there may have been experimental error in the simple water experiment. In order to evaluate this possibility, as well as provide insight into whether the Maximum Algorithm was actually an improvement upon the RHDRC algorithm, the Maximum’s results were compared to the RHDRC’s results on the same two data sets.

6.3 Comparison of Maximum to RHDRC

Before directly comparing the percent difference values, an investigation into the difference calculations from Phase I for the simple water experiment data was performed. The percent differences presented in Tables 1 and 2 were actually incorrect. Phase I averaged positive and negative difference values instead of averaging absolute difference values, thus lowering the overall difference. The corrected difference calculations for the RHDRC algorithm are seen below in Table 3:

Table 3. Percent difference between RHDRC algorithm and scale data for the simple water experiment

<table>
<thead>
<tr>
<th>Mean % Difference</th>
<th>Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal H₂O</td>
<td>1.955</td>
</tr>
<tr>
<td>Low Gain</td>
<td>1.921</td>
</tr>
<tr>
<td>Low Freq</td>
<td>1.960</td>
</tr>
</tbody>
</table>

A comparison of Tables 1 and 3 indicates that the Maximum Algorithm and RHDRC algorithm performed
very similarly on the simple water data. Figure 4 plots the
burn rates calculated for each trial by the scale, the Maxi-
mum Algorithm, and the RHDRC algorithm. As noted, the
two algorithms follow a similar trend, which is not fol-
lowed by the scale data.

![Figure 4. Comparing Water Release Rates for Maximum
Algorithm, RHDRC, and Scale Data](image)

The percent difference between the ultrasonic methods
(RHDRC and Maximum Algorithm) is less than the per-
cent difference between the ultrasonic methods and the
scale measurements. This suggests that the ultrasonic
method is more accurate than the scale data. Because the
testing method may be less accurate than the ultrasonic
method, it is hard to determine exactly how accurately the
Maximum Algorithm performed on the simple water data.

Since the complex water experiment was not per-
formed until 2004, it could not be tested during Phase I. In
order to further evaluate the Maximum Algorithm vs. the
RHDRC algorithm, the complex water data was subjected
to the RHDRC algorithm. After several attempts, it was
determined that the RHDRC algorithm was unable to han-
dle the more complex data, and results could not be col-
lected.

### 7 GUI

Because of the expanded scope of the RHDRC algorithm
and shortcomings of the Phase I prototype GUI, a new GUI
was designed for the Maximum Algorithm. There are sev-
eral underlying standards that designers should adhere to
when crafting an interface, including but not restricted to:
creating a simple and natural dialogue, implementing lay-
out consistency, speaking the user’s language, providing
feedback, and clearly marking exits. These heuristics were
taken into account during the Maximum Algorithm GUI
design process. The following details how these standards
were incorporated into the GUI.

- **Simple and Natural Dialogue** – The interface contains
all the needed functionality, and not much more,
minimizing room for error and confusion.

- **Consistency** – The various windows maintain a consis-
tent color scheme and layout. All main functions are
the same color and shape and are located in the same
area throughout the four screens.

- **User’s Language** – The interface shouldn’t confuse
the user and therefore employs familiar terminology.
The interface is intended for scientifically-based users,
however, can still be learned by common users.

- **Feedback** – Feedback is provided through the use of
progress bars, a function checklist, as well as guide-
lines for range definition while tracking the moving
echo. The checklist serves as a reminder to the user
concerning where he is in the program and what tasks
have been preformed thus far. In the “track moving
echo” window, the user must define ranges of two
echoes and graphical feedback aids the user in detect-
ing the most appropriate range.

- **Marked Exits** – Each window contains two exits, the
traditional “X” to close out a window, as well as an
additional “Close” button so the user does not feel
trapped by the program.

- **User Support** – The addition of “Help” buttons was im-
portant to guide the user through unknown areas of the
program. A user’s guide also was written to accom-
pany the interface.

### 8 CONCLUSIONS AND RECOMMENDATIONS

The Maximum Algorithm seems to have successfully
tracked the target echo reflecting from a regressing surface.
The calculated rate lies within 3% accuracy in both ex-
perimental water data sets. The lack of known burn rates
for the live rocket burn data set has limited the Maximum
Algorithm’s validation process, but a visual comparison
evined that the Maximum Algorithm closely paralleled
the actual burn rate. It is important to note that the Maxi-
mum Algorithm successfully overcame the deficiencies
which led to the failure of the RHDRC algorithm on the
two complex data sets. Considering the consistent per-
formance and overall accuracy of the Maximum Algorithm
for both the live rocket burn and the water data sets, it can
be concluded that it is more robust and therefore preferred
to RHDRC algorithm.

An improved GUI adds to the results of the project.
Specifically, the addition of appropriate labeling, feedback
mechanisms, and advanced functionalities facilitates the
interpretation of results. Overall, the new GUI enhances
the user’s capacity to interact with the Maximum Algo-
rithm and produce accurate results.

To enhance the statistical results, additional data must
be obtained from experiments whose environments are
more representative of the non-constant burn rate of solid-
propellant rockets. Furthermore, since the availability of
the actual burn rate is essential for verification of the re-
sults, simulation must be evaluated as an alternative. Fi-
nally, the percentage error associated with the experiment itself must be precisely accounted for in evaluation of the algorithm.

ACKNOWLEDGMENTS
This document was prepared based on the prior work of the Aerojet SIE Capstone team of (2002-2003). In particular, J. Y. Ivey, T. Rubenstein, A. Snyder, M. Stewart, E. Luebke, K. P. White, P. K. Wu, D. Whitney, and J. Sparks deserve special mention for their previous work and assistance.

REFERENCES


AUTHOR BIOGRAPHIES
YE PETER CHEN is a graduating fourth-year Systems and Information Engineering student at the University of Virginia. Peter will be working for American Management Systems in Herndon, VA, in the fall of 2004. He can be contacted by email at ypc5q@virginia.edu.

MICHÈLLE C. GAUDIOSE is a graduating fourth-year Systems and Information Engineering student from Chambersburg, PA. Michelle will be working for Lockheed Martin in Baltimore, MD, beginning in the summer of 2004. She can be contacted by email at mcg7w@virginia.edu.

RYAN P. MURPHY is a graduating fourth-year Systems and Information Engineering student from Oakton, VA. Ryan will be working for McKinsey & Company in Atlanta, GA, beginning in the fall of 2004. He can be contacted by email at rpm6x@virginia.edu.

LAURA M. SCHRADER is a graduating fourth-year Systems and Information Engineering student from Andover, MA. Laura will be working for Booz Allen Hamilton in McLean, VA, in the fall. She can be contacted by email at lms2h@virginia.edu.

REZA SEIRAFI is a graduating fourth-year Systems and Information Engineering student. He can be contacted by email at rs8fd@virginia.edu.

K. PRESTON WHITE, JR., is Professor of Systems and Information Engineering (SIE) at the University of Virginia. He received the B.S.E., M.S., and Ph.D. degrees from Duke University. He has held faculty appointments at Polytechnic University and Carnegie-Mellon University and served as Distinguished Visiting Professor at Newport News Shipbuilding and at SEMATECH. He sits on the VMASC Advisory Board and currently serves as Chairman of the Board of Directors of the Winter Simulation Conference. He is a member of INFORMS and INCOSE, a senior member of IEEE and IIE, and is listed in Who’s Who in America. His e-mail address is kpw8h@virginia.edu.

PEIKUAN WU, is a Principal Staff Engineer at Aerojet Corporation, Gainesville. He received his BS and MS in Mechanical Engineering at National Taiwan University and his Ph.D. in Aerospace Engineering at University of Michigan. He is a senior member of AIAA. He also serves in AIAA and JANNAF technical committee. His email address is PK.Wu@aerojet.com.