NISTIR 5118

Development of a Fast-Response Variable-Amplitude Programmable Reaction Control System

Building and Fire Research Laboratory
Gaithersburg, Maryland 20899

NIST
United States Department of Commerce
Technology Administration
National Institute of Standards and Technology

NISTIR 5118

Development of a Fast-Response Variable-Amplitude Programmable Reaction Control System

William C. Stone

January 1993
Building and Fire Research Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

U.S. Department of Commerce
Ronald H. Brown, *Secretary*
National Institute of Standards and Technology
John W. Lyons, *Director*

Abstract

This report describes a high speed loading system comprised of: a piezoelectric stack and an associated microprocessor-based programmable DC power source; a low-loss mechanical displacement amplifier; a high pressure spring-loaded axial valve; an integral high pressure valve seat; an expansion nozzle; and a high pressure gas supply. A half square wave voltage signal of varying duration and amplitude was used to drive the piezoelectric stack in such a manner as to produce changing displacements which were subsequently amplified by a monolithically milled hinged anvil. The anvil is connected to a valve core which seats upon, and is normally forced closed upon a specially hardened throat of an expansion nozzle by means of a compression spring. The interior side of the thruster nozzle communicates with a high pressure gas supply. The flow of gas through the nozzle is normally prohibited by the spring loaded valve core. As the piezoelectric stack expands under computer control the valve core is lifted off the nozzle throat permitting gas to expand through the nozzle and create thrust that is directly proportional to the control signal from the microprocessor. The device is designed to operate as a stand-alone unit with a dedicated onboard microcontroller system and onboard energy storage system. Minimum pulse width resolution (base-to-base) was shown to be 0.98 ms with a lag time of 0.37 ms relative to the initiation of the drive pulse. Linear amplitude response was achieved beyond a threshold drive voltage of 80 vdc and was maintained through the limit of testing at 240 vdc. Thrust change rates of 12000 N/s were achieved. The demonstration unit developed for this study produced a peak thrust level of 15 N at operating pressures of 41 MPa.

Keywords: active control; amplitude control; cold gas thruster; microprocessor control; piezoelectric stack; pulse width resolution; reaction control system; system identification;

List of Symbols and Acronyms

a	=	Long Lever Arm of Displacement Amplifier (mm)
A_{anvil}	=	Cross Sectional Area = 780 mm^2
A_e	=	Nozzle Exit Area (m^2)
A_t	=	Nozzle Throat Area (m^2)
ACR	=	Amplitude Control Rate (N/s)
A/D	=	Analog to Digital Converter
b	=	Hinge to Center of Mass Length of Displacement Amplifier (mm)
c	=	Short Lever Arm of Displacement Amplifier (mm)
C	=	Capacitance of Piezoelectric Stack (F)
C_t	=	Thrust Coefficient
C^*	=	Generalized Damping
d_{th}	=	Nozzle Throat Diameter (mm)
DPW	=	Drive Pulse Width (s)
D/A	=	Digital to Analog Converter
E(t)	=	Instantaneous Piezoelectric Drive Voltage
E_{HSILS}	=	Active Energy Dissipation in HSILS Drive Pulse (J)
E_{RCS}	=	Active Energy Dissipation in Solenoid Drive Pulse (J)
EPROM	=	Erasable Programable Read Only Memory IC chip
F	=	Actual Thrust (N)
F_{imp}	=	Specified Peak Impulse Force (N) = F_{max}
F_{max}	=	Peak Gaussian Pulse Force (N)
$F_{preload}$	=	Return Spring Preload Compression Force (N)
F_{theo}	=	Theoretical Impulse Force at Time t (N)
F_{C1}	=	Damping Force (N)
F_{K1}	=	Force Due to Spring Compression/Extension (N)
F_{M1}	=	Inertial Force Contributed by Displacement Amplifier (N)
F_{M2}	=	Inertial Force Contributed by Valve Core Assembly (N)
HDYN	=	Program for Dynamic Simulation of HSILS Response
HSILS	=	High Speed Intelligent Loading System
I_{sp}	=	Specific Impulse (s)
I_x	=	X-Axis Moment of Inertia of Displacement Amplifier (mm^4)
I_{xy}	=	Polar Moment of Inertia of Displacement Amplifier (mm^4)
I_y	=	Y-Axis Moment of Inertia of Displacement Amplifier (mm^4)
I_0	=	Mass Moment of Inertia of Displacement Amplifier (g-mm^2)
K_1	=	Precompression Spring Stiffness (N/mm)
K^*	=	Generalized Stiffness
M	=	Molecular Weight of the Propulsive Gas
M_1	=	Valve Core Mass (g)
M_2	=	Displacement Amplifier Mass (g)
M_I	=	Inertial Moment caused by Displacement Amplifier Rotation (N-m)
M^*	=	Generalized Mass
MMH	=	Monomethyl Hydrazine

NIST	=	National Institute of Standards and Technology
n	=	Number of Data Scans
PC	=	Personal Computer
P_a	=	Ambient Pressure (Pa)
P_c	=	Chamber Pressure (Pa)
P_e	=	Nozzle Exit Pressure (Pa)
P_0	=	Initial Chamber Pressure (Pa)
P(t)	=	Piezoelectric Forcing Function (N)
R	=	Solenoid Resistance (ohms)
R_a	=	Required Displacement Amplification Ratio
R_{con}	=	Universal Gas Constant
RCS	=	Reaction Control System (in this report refers to a generic, solenoid operated reaction control system or gas thruster)
RL	=	Rise Lag (s)
RPW	=	Response Pulse Width (s)
t	=	Elapsed Time (s)
t(i)	=	Time at Data Scan i (s)
t_{anvil}	=	Cross section thickness of amplifier = 10mm
T_{abs}	=	Absolute Temperature
vdc	=	Volts, Direct Current
V(i)	=	Drive Pulse Voltage at Scan i
V_{drive}	=	Required Drive Voltage from Pulse Power Supply to Achieve δ_c
V_{peak}	=	Maximum Drive Pulse Voltage
V_{tank}	=	Pressure Vessel Volume, given by $(4/3)\pi(D/2)^3$
$w^{\cdot}$	=	Mass of Gas in Accumulator System (kg)
Y^*	=	Valve Core Displacement at Time, t (mm)
γ	=	Specific Heat Ratio of the Compressed Gas
δ_c	=	Valve Core Displacement (mm)
δ_{stack}	=	Piezoelectric Stack Displacement (mm)
δ_y	=	Piezoelectric Stack Displacement (mm)
δW	=	Virtual Work
σ	=	Gaussian Pulse Shape Function
θ	=	Valve Core Half Cone Angle (degrees)
$\tilde{\omega}$	=	Natural Frequency of HSILS System (Hz)

Table of Contents

1.0 Introduction

A series of mathematical investigations were carried out at NIST between 1986 and 1989 [Carasso & Simiu (1988); Carasso and Simiu (1989); Simiu and Cook (1989); and Carasso (1991)] with the objective of developing robust system identification techniques for the response of flexible structures. These techniques are likely to be particularly useful for structural systems with non-collocated active control, where noise is present in the sensor data used as input to the active control algorithm. Examples of such structural systems include flexible robotic manipulators, tension-guyed deep water offshore oil platforms, active hook control for construction cranes, and large orbiting spacecraft.

The research reported by Carasso and Simiu was based on the finding that it is advantageous to recover the dynamic characteristics of a structure by deconvolution of an output due to an excitation by a pulse defined by an inverse Gaussian function or an approximation thereof. Where identification of higher order mode shapes and frequencies is sought the drive pulse must be short. In the case of certain spacecraft, for example, it is apparent that active control systems should accomodate structural response at frequencies of vibration of several hundred Hertz (Belvin, 1986). This would, in turn, imply the need to impart to the structure a smooth, uni-directional, mathematically defined test pulse whose total duration is less than, e.g., 0.001 s.

The need for such a device motivated research, started in 1988, with the goal of developing a working prototype High Speed Intelligent Loading System, or HSILS (HSILS, 1990), and to ascertain: its pulse width resolution; its ability to interactively vary amplitude (force applied to the test structure); and the feasibility of generating mathematically definable force-time histories using either closed or semi-closed loop control. Implicit in the latter objective was the necessity of employing an onboard, programmable microprocessor control system which could be used to drive the device as a stand-alone thruster (e.g. as might be used on a space craft). Note also that a required characteristic of such a control/identification loading system is that it be capable of operating without the presence of a reaction surface, frame, or other interacting structure. The reason for this requirement is obvious for the case of the control of spacecraft and orbiting structural systems.

In the present paper we discuss the background of the design that was ultimately developed, the rationale behind the selected technology, and the experimentally observed pulse width resolution and amplitude control behavior of the system. The capability of the system to impart random shape, pre-defined impulses to a test structure will be discussed in a subsequent report.

1.1 Background:

The present report relates to the problem of providing a means for introducing a force with a predetermined variation in time (harmonic or non-harmonic) into a mechanical or structural system. The system is assumed to consist of one or more rigid bodies or flexible elements, and the force is used for active control or structural dynamics identification.

The first phase of the work reported here consisted of an extensive search on existing valves and on independent actuator technologies that might be integrated to form an amplitude controllable, high speed valve. The results of this search are presented below.

1.1.1 Servo-hydraulics

Closed loop servo-hydraulic and servo-pneumatic loading equipment is now in common use for structural testing and industrial fabrication. Such equipment works on the principle of directing a pressurized fluid (either hydraulic oil, or compressed air) from a pressurized reservoir to a bi-directional actuator which may be made to extend, contract, or remain stationary depending upon the state of an electronically controlled servovalve. While it is possible to program these actuators to impart an "intelligent" forcing function into a test structure, the response time to an instantaneous change in the command signal is generally about 1/30 second or longer, although special high speed versions have been developed.

Fig. 1.1 shows an exploded schematic of one such valve (a Moog Series 760 electrohydraulic servovalve). A typical limiting response under ideal conditions for such an actuator is shown in Fig. 1.2 in which it can be seen that there is approximately a 10ms rise time to an impulsive loading with approximately the same time (10 ms) lag time between the initiation of the electronic drive pulse to the initiation of actuator movement. Generally there will also be an associated termination or fall lag which is similar in duration to the rise lag. From this it may be inferred that the minimum pulse width resolution is about three times the rise time, or 30 ms.

An important additional drawback of servo-hydraulic valves is evident from Fig. 1.2, where it can be seen that there is no direct correlation between the drive signal and the actuator response. This makes this type of system particularly unsuitable for active control of short period structures. It may be further noted that such actuators require a reaction system and are therefore unsuitable for active control of free bodies.

1.1.2 Pneumatic Solenoids

Several loading systems presently exist for spacecraft which require no reaction surface. These fall into a broad classification known as Reaction Control Systems (RCS). Of particular interest to the present discussion are those reaction control systems which comprise small rocket thrusters that may be used for translational as well as rotational control of spacecraft and orbiting structural systems. Classical RCS thrusters employ a pressurized supply of liquid fuel, an electronically controlled solenoid-type valve, a combustion chamber (if the thruster is of a bi-propellant variety), and an expansion nozzle. Alternatively, the propellant may simply be an elemental inert gas, such as nitrogen, in which case the RCS is known as a "cold gas thruster."

These control solenoids (e.g. Fig. 1.3) are normally closed and only the duration of the open time is subject to variation. There is typically a threshold voltage which is required to activate motion of the solenoid core but beyond that further increases in voltage generally yield no further increase in the amplitude of the flow nor do they influence the minimum pulse width of the system (experimental data showing this behavior is presented later in this paper).

While solenoid controlled RCS thrusters can impart differing levels of thrust duration, there is no means of varying the thrust level (which is controlled by the fixed diameter of the thruster nozzle) and therefore no "intelligence" can be imparted to the load-time history. Furthermore, such systems suffer from significant lag times in response to changes in a control signal. An example response from a typical RCS solenoid, used in a cold gas thruster configuration, is given in Fig. 1.4 (**Hallauer and Smith, 1990**).

Hallauer and Smith made use of a Mac Valves, Inc. Model 113B-551BA solenoid valve which, in their specific application was referred to as an AJT (Air Jet Thruster). The response shown in Fig. 1.4 was achieved using this particular valve although others (as will be described below) could have been used that would have yielded similar responses. Regarding lag times (between the leading edge of the drive pulse and the initiation of valve opening) for solenoids, Hallauer and Smith report that:

> "[there are] time delays between the control signal stimuli at the zero-crossing instants and the realization of either full-on or full-off thrust. The full-on and full-off delays are about 10ms and 5ms, respectively. Time delays are very important relative to application of AJTs for active vibration control, because they tend to destabilize closed-loop control systems. These delays are due primarily to the fluid and mechanical dynamics of the solenoid valve itself."

The issue of binary amplitude response is also commented upon:

> "The thrust of AJTs built from standard solenoid valves cannot easily be modulated, because these valves have only two stable states, fully closed and fully opened."

It has been observed (**Kattchee,1967**) that RCS solenoid operated thrusters

> "lack of precise impulse bit repeatability due to the opening and closing characteristics of the valves, manifold fill times, and chemical ignition delay times."

The term "impulse bit" is defined as the integral of the force (N) applied by the thruster over the domain of its minimum pulse width (s).

1.1.3 Piezo-Pumped Reaction Control System

A second type of RCS thruster (see Figs. 1.5 and 1.6) employs a piezoelectric pump for the two fuel components of a bi-propellant thruster (**Kattchee, 1967**). This is known as the pulse-pumped vernier engine concept and operates on the principle of pulsing a piezoelectric stack with a specified voltage waveform which then causes a metered amount of propellants to be injected into a combustion chamber, thereby producing thrust. The threshold "impulse bit" is determined when measurable thrust is just produced at the shortest possible drive pulses at the lowest possible amplitude (voltage).

There are strong reasons for using piezoelectrics in fast valve technology. A typical modern piezoelectric element will reach maximum extension in response to a step change in applied voltage in approximately the time required to propagate sound through the piezoelectric material in the direction of expansion. Thus, for a Lead Zirconate Titanate (PZT) stack measuring approximately 75 mm in height, the response time to a change in drive voltage is approximately 10 microseconds. Under ideal conditions this would translate to a minimum pulse width (triangular pulse) of 20 microseconds. As will be discussed below, mechanical, acoustic, viscous damping, and chemical reactant delays will degrade the performance. The degree to which such speed can be approached depends on the skill of the mechanism designer, on the required thrust amplitude, and on other performance characteristics.

The concept of a piezoelectric stack has changed somewhat during the past 25 years. Originally, due to material processing considerations, the size of a "high performance" piezoelectric element was severely limited and high axial strains were only achievable in relatively thin elements, usually in the form of either disks, and rectangular or annular wafers. For the particular proof-of-concept program reported by Kattchee, PZT-5H (Lead Zirconate Titanate) ceramic disks were stacked one on top of the other to assemble a cylindrical piezoelectric "stack". This particular material has a maximum axial strain of approximately 0.02% (**Piezoelectric Ceramics, 1974**), which, for example, would result in a maximum 0.004 mm total extension for a 20 mm high stack.

The piezoelectric stack used in Katchee's thruster is connected to a series of inlet and outlet valves which permit the piezoelectrically operated pulse-pump to draw propellant into a holding chamber during a constriction of the stack and to expel the propellant through the outlet valve during an expansion of the stack. Each such cycle delivers a finite, measurable quantity of fuel to a combustion chamber and subsequently to an expansion nozzle. The frequency and duration of the arrival of the electrical pulses which drive the piezoelectric stack thus determine the total impulse delivered by the thruster during a specified length of time. This type of operation is analogous to the manner in which pulse-width modulation can be used to turn a light on for specified periods of time at the same illumination level.

The approach described above appears to be a highly effective method for metering precise RCS impulses. The shortest pulse width measured by Katchee (see Fig. 1.7) was on the order of 3-5 ms with corresponding lag (response) times of approximately 3ms. Both of these performance indicators are considerably superior to the solenoid RCS systems previously described. In this particular situation, lag time is comprised of a number of components including a) the expansion/contraction rate of the piezoelectric stack; b) viscosity related flow resistance delays; and c) ignition delay.

Two points are worth discussing with respect to this type of control system. First, the amplitude variance (peak force) of the thrust which can be achieved in this manner is severely limited by the maximum stroke of the piezoelectric stack and thus relatively large stacks must be used to achieve significant valve displacements. Second, because the fuel charge must be replenished after each firing, and because the volume is essentially predetermined by the geometry of the pump system, there is a further limitation on the ability to control amplitude: once the stack has achieved maximum stroke it can displace no further fuel. Thus amplitude modulation, over a wide range of both pulse durations and amplitudes -- necessary to achieve an "intelligent" force-time pulse -- is not practicable with this approach.

1.1.4 Piezoelectric Metering Valves and Fuel Injectors

There are a number of similar piezoelectric stack-driven valve systems in use for purposes other than aerospace reaction control systems. The two most striking examples are injection valves for atomic spectrophotometers and the more commonly known application in fuel injectors for engines. For spectrophotometers, the metering of usually minute amounts of trace or reactant elements into a carrier stream is of interest. Injection of large quantitities is not of great importance and hence axial stacks, similar to that described by Katchee (**Katchee, 1967**) have been used and improved upon. Examples of such valves are shown in Figs. 1.8, 1.9, and 1.10 (from **Weber & Blanckenhagen, 1987; Kippenhan, 1962; and Yamauchi, Nogi, & Ohyama, 1991**, respectively). All of these have the common elements of a piezoelectric stack, an expansion nozzle, and a spring loaded valve means which normally prevents pressurized fluid or gas from escaping through the nozzle. Webber and Blanckenhagen employ a separate valve core (Fig. 1.8) which is unseated from the nozzle throat when the piezoelectric stack expands. Kippenhan (Fig. 1.9) attaches the valve seat directly to the end of the piezoelectric stack. In a minor variation, Yamauchi, Nogi, and Ohyama (Fig. 1.10) allow a spring loaded metal sphere to lift off the valve seat when the piezoelectric stack expands.

All of these valves have minimum pulse widths on the order of 5-7ms. However, unlike the valve utilized by Katchee, each of the valves shown in Figs. 1.8, 1.9, and 1.10 is pressure-fed and therefore can maintain, for as long as fuel and pressure remain, a fixed mass flow rate (and therefore, in the RCS sense, a thrust amplitude). All of these valves exhibit the limitation on amplitude range common to all axial stack valves described above. Similar designs, with minor variations to suit a particular set of needs, are described by **Kushida, 1986; Hiratsuka, Kawasaki & Yashuhiko, 1991; and Janner & Gregorius, 1984.**

1.1.5 Mechanically Amplified Piezoelectric Valves, Actuators, Electrical Relays and Dot Matrix Printer Drivers

As mentioned above, direct axial valve control using a piezoelectric stack imposes a severe restriction on the range of motion of the valve core. One means of circumventing this problem is to use a mechanical displacement amplifier, such as shown in Fig. 1.11 and as described by (**Moloney, 1986**). The device depicted in Fig. 1.11 is an exhaust valve for an internal combustion engine. Unlike the previous injectors, in which fairly high pressure is maintained on the fuel resevoir side to offset the displacement limitations of the axial stack, the valve shown in Fig. 1.11 must operate under reverse pressure and also deliver relatively large opening displacements in order for the vent to operate properly.

The displacement amplifier used by Moloney is in essence a lever in which the centerline of the force generated by the piezoelectric stack is located a slight distance from the lever hinge and, in the particular configuration shown in Fig. 1.11, on the same side of the hinge as the actuator element which is to be the recipient of the enhanced displacement. Moloney provides no performance data on this actuator design in his patent, but specific reference is made to the piezoelectric stack as being comprised of a series of piezoelectric disks. From this it might be inferred that the device will exhibit similar response characteristics to axial stack designs; that is, in the range of 5 ms pulse widths. The presence of sliding valve guides will introduce damping into the dynamic system which, in general, increase the minimum achievable pulse width. The effects of damping will be discussed in detail later in this report.

Other, less intuitive, displacement amplifiers have been developed by **(Knappe, 1972; Mitsuhashi, Wakamatsu, Aihara, & Okihara, 1985; and Takahashi, 1985)**. Fig. 1.12 shows an "operating device for high speed reciprocating machine parts," described by Knappe in which two opposing piezolelectric stacks are used to move a hinged actuating plate. In this particular design a second piezoelectric actuator is used to create a moment couple that in turn causes pure flexural loading of an actuator beam. The amplification factor is thus determined by the ratio of the half-length of the flexural beam divided by the centerline to centerline distance between the two stacks.

Mitsuhashi et al. describe a piezoelectrically operated electronic relay, shown in Fig. 1.13. Its heart is a piezoelectric stack and an associated displacement amplifier. As part of this study Mitsuhashi et al. investigated the performance of various piezoelectric devices, including both bimorphs and the more modern version of an axial stack known as a "monolithic" or "cofired" stack. Fig. 1.14 shows two types of flexural bimorph actuators and a modern monolithic stack.

A bimorph, in its most simple form, can be envisioned as a bilaminar cantilevered beam. It consists of two thin piezoelectric strips bonded together in a fashion analogous to that of a bimetalic strip thermocouple. As shown in Fig. 1.14 the strips can be poled either in parallel or in series. When a voltage V is applied between the electrodes on their outer surfaces the upper strip contracts lengthwise while the lower one expands lengthwise. This differential strain causes the cantilever to bend. The cofired stack shown in Fig. 1.14 uses internal and external electrodes to join hundreds of very thin, but high strain rate, planar elements which are then fused to form a monolithic stack. This technique results in an improved performance, as measured by maximum axial strains of 0.1% compared with 0.02% from a series of stacked individual disks.

Table 1.1 presents typical laboratory results obtained by Mitsuhashi et.al. These clearly indicate that bimorph type actuators are capable of achieving substantially greater absolute deflections than a stack. However, the maximum force that can be achieved using a bimorph type element is quite limited and this factor essentially disqualifies it from use as a high pressure valve component. The monolithic stack, on the other hand, while having superior force and energy capacity characteristics, clearly suffers from the drawback of limited deflections, in spite of the improvements in axial strain capacity.

Mitsuhashi et.al. determined that the strain limitations of the monolithic stack could be overcome (as others concluded above) by means of mechanical amplifiers:

> "multilayer piezoelelectric actuators themselves displace very little, 7.8 um, at a rated drive voltage (E=1000 V/mm), and the displacement has to be magnified at least more than 60 times to obtain a displacement of 0.5 mm or more, as the driving source of relays."

A few of the concepts they investigated are shown in Fig. 1.15. Most of these are numerically machined single metal systems containing low resistance flexural hinges. They go on to point out that:

> "the most important point in designing a high efficiency magnification mechanism is the lengths among the point of force, fulcrum, and point of application of each arm, namely, the lever ratio, and spring stiffness of each hinge spring."

Using these concepts and parametric finite element analysis, Mitsuhashi et al. developed a working prototype electronic relay as shown in Fig. 1.13 with a magnification factor of 70. The minimum opening time, including bounce, was shown to be 2.5 ms and the authors claim a peak switching speed of 300 Hz (implying 3.3 ms minimum pulsewith).

A final example of an amplified piezoelectric stack system is shown in Fig. 1.16, as described by Takahashi. The operation of this amplifier, as used in a dot matrix printer, is as follows:

> "the actuator stroke is amplified by a two stage hinge-lever mechanical linkage. When the actuator expands upwards, levers I and II simultaneously rotate in clockwise and counter-clockwise directions, respectively. Lever III then rotates in a counter-clockwise direction and the wire attached to the lever III end moves. An impact matrix printer head can be constructed with these elements, where the ends of wires are arranged in a line through wire guides. The wire ends strike a paper through an inked ribbon. The printer head thus-obtained has higher printing speed, lower power consumption, and less heat generation, compared with a conventional electromagnetic drive printer head."

The last statement above is particularly important since it underscores the superior speed of piezoelectric actuators over electromagnetically operated systems (as, for example, the solenoid operated pneumatic RCS thrusters previously described). The project undertaken by Takahashi was more fundamental than the end-product development of an impact matrix printer. The primary scope of the research was to develop an integrated monolithic stack with performance superior to a discrete stack assembly of the type used by Katchee. The new manufacturing technique is shown graphically in Figs. 1.17 and 1.18. The specific process was as follows:

> "piezoelectric ceramic calcined powder with some organic binder, plasticizer and solvent is milled into a slurry. The slurry is cast into a "green" sheet several tens of micrometers thick on an organic film using a spatula. After drying, the green sheet is stripped from the organic film and is cut into appropriate sizes. Metal paste is printed on it as the internal electrode. A number of green sheets with metal paste are laminated and pressed together. The pressed body is heated to about 500°C for

> more than 100 hours to break down the organic materials included as binder and plasticizer. The ceramic body is then sintered in place at 1000-1300°C. The sintered body is cut into appropriate sizes, according to the various projected uses, and an individual internal electrode is electrically connected in parallel with two external electrodes."

Takahashi was able to achieve a 15 um displacement from a 9 mm long stack at 300V, roughly a 0.2% strain. Further properties are presented in Table 1.2 in which comparisons are drawn between bimorph and cofired stack actuators developed from similar materials. The chief advantages of the stack, as previously noted, are high force, and high response speed.

1.1.8 Microprocessor Controlled Harmonic Force Actuator

Zimmerman et.al. (**Zimmerman, Horner, & Inman, 1986**) developed an autonomously controlled proof-mass actuator for active vibration control of lightweight, relatively flexible, lightly damped spacecraft as shown in Fig. 1.19. The proof-mass actuator applies to the structure (spacecraft) a cyclic force in a manner analogous to that used for tuned-mass dampers now commonly used to control the motions of office buildings in seismic regions. Zimmerman et.al. explain:

> "The force applied to the structure by the actuator is equal and opposite to the force exerted on the proof-mass. The finite motion of the proof-mass is controlled by an electromagnetic field [generated by coils located within the apparatus; see Fig. 1.19]. The proof-mass actuator is used only for vibration control since the finite motion of the proof-mass does not allow it to be used for rigid body control. The low frequency force output of the actuator is limited by the displacement of the proof-mass, whereas the high frequency force output is limited by the maximum allowable coil current. The allowable coil current is such that the coil does not overheat.
>
> The force output of the proof-mass actuator is equal and opposite to the force applied by the electromagnetic field on the proof-mass. The actuator is comprised of a movable proof-mass, a fixed coil, two collocated sensors, a digital microcontroller, and a power amplifier. All of the components are mounted as a single unit which minimizes the interconnecting cable lengths. The use of external cables required in implementing control would greatly affect the dynamics of the structural system. Additionally, the effects themselves may change on a daily basis. Therefore, the actuator was designed such that power lines are the only required external lines for the actuator to operate."

The actuator developed by Zimmerman et al was capable of generating steady state force output of 2.75 N from 2 Hz through 1000 Hz. Although this device cannot produce monotonic (non-cyclic) impulses of the type required for rigid body motion control, several of its concepts, namely the use of onboard sensors, power conversion, and microcontrollers for the implementation of adaptive control algorithms (Figs. 1.18 and 1.19), are particularly useful in considering the design of a more robust thruster.

1.2 Summary

Of all the potential actuator technologies reviewed here the piezoelectric, cofired stack has the highest performance characteristics in terms of response speed and deliverable axial force. It does, however, have the drawback of providing these at the expense of axial extension. The use of a piezoelectric stack for the purposes of valve operation (in which finite displacements are required in order to achieve effective mass flow) requires the presence of a mechanical displacement amplifier. As will be described in Chapter 2 there are tradeoffs in performance that result from the mechanical design details of the displacement amplifier (its natural frequency, damping, and hinge stiffnesses) and from the selected amplification ratio.

For purposes of practical engineering utility a reaction control system must be capable of being mounted at the control point of interest on the target structure (for example, at the end of a flexible beam which contains an effector at that location) without significantly affecting the dynamic characteristics of the structure. This demands the use of compact, lightweight hardware in combination with an onboard active control system and sensors, as well as primary power storage for short term operation of the actuator. The means by and degree to which these attributes were achieved in the NIST prototype HSILS unit are discussed in Chapters 2, 3 and 4.

There are extensive appendices at the end of this report which discuss simulation software used to model the performance of candidate HSILS designs; electronic schematics for the onboard microcontroller and associated energy storage and power conversion electronics for driving a cofired stack; low level software for the microcontroller subsystem; and complete shop drawings for the HSILS prototype unit constructed at NIST.

1.3 Scope and Limitations of NIST Research

An extensive review of the available literature indicated that precision identification and control of flexible structures is predicated upon the ability to provide an "intelligent" forcing function with as short a pulse width resolution as possible and in which amplitude control is feasible. Several sources indicated that active control may be desirable to frequencies of 1000 Hz. This implies the generation of a minimum pulse width of less than .001 seconds and yet of sufficient amplitude so as to be useful for control of real life engineering structures.

Accordingly, the following objectives were identified at the inception of the NIST project to develop a prototype HSILS:

a) to provide a thruster-type loading system which is capable of achieving "intelligence" in a force-time pulse of short duration (i.e. a high speed intelligent loading system).

b) to provide a high speed intelligent loading system for which the amplitude of the force-time pulse is capable of being varied over a wide spectrum of force levels in a continuous manner.

c) to provide a high speed intelligent loading system which has the capability of applying an arbitrary force-time pulse to a structure without the need of a reaction surface nor any support infrastructure.

d) to provide a highly responsive, programmable high speed loading system which is controlled locally by an embedded microcontroller with onboard memory and is capable of being operated in either open or closed-loop control, without and with force and/or acceleration feedback, respectively.

e) to provide a highly responsive, programmable cold gas reaction control thruster which may be used for precision control of spacecraft and orbiting structures.

The degree to which each of the above objectives was met is discussed in Chapter 4. Although the electronics control system described in this report is capable of synthesizing any voltage-time history (and thereby causing the HSILS output thrust to track accordingly), the initial laboratory data reflect experimental behavior of the HSILS device when driven by half square wave voltage-time histories of varying pulse width and amplitude.

Item No.	Nomenclature or Description	No. Req'd
1	SCREW, SCH CAP	4
2	CONNECTOR ELECTRICAL	1
3	O-RING, CONNECTOR	1
4	SCREW, SCH CAP	4
5	LOCKWASHER	4
6	CAP MOTOR	1
7	O-RING, MOTOR CAP	1
8	SCREW, SCH CAP	4
9	LOCKWASHER	4
10	HYDRAULIC AMPLIFIER ASSY	1
11	RIVETS	2
12	O-RING	1
13	UNION	1
14	O-RING	2
15	O-RING	1
16	O-RING	1
17	FITTING SELF-LOCKING	1
18	PIN, BUSHING	1
19	O-RING BUSHING PIN	1
20	SCREW SCH CAP	8
21	LOCKWASHER	8
22	CAP, END	1
23	CAP, END	1
24	O-RING	2
25	O-RING	2
26	BUSHING AND SPOOL ASSY	1
27	O-RING	6
28	O-RING	2
29	FILTER PLUG	2
30	O-RING	2
31	INLET ORIFICE ASSY	2
32	FILTER TUBE	1
33	NAMEPLATE	1
34	BODY	1
35	PIN, ROLL	1
36	O-RING	4

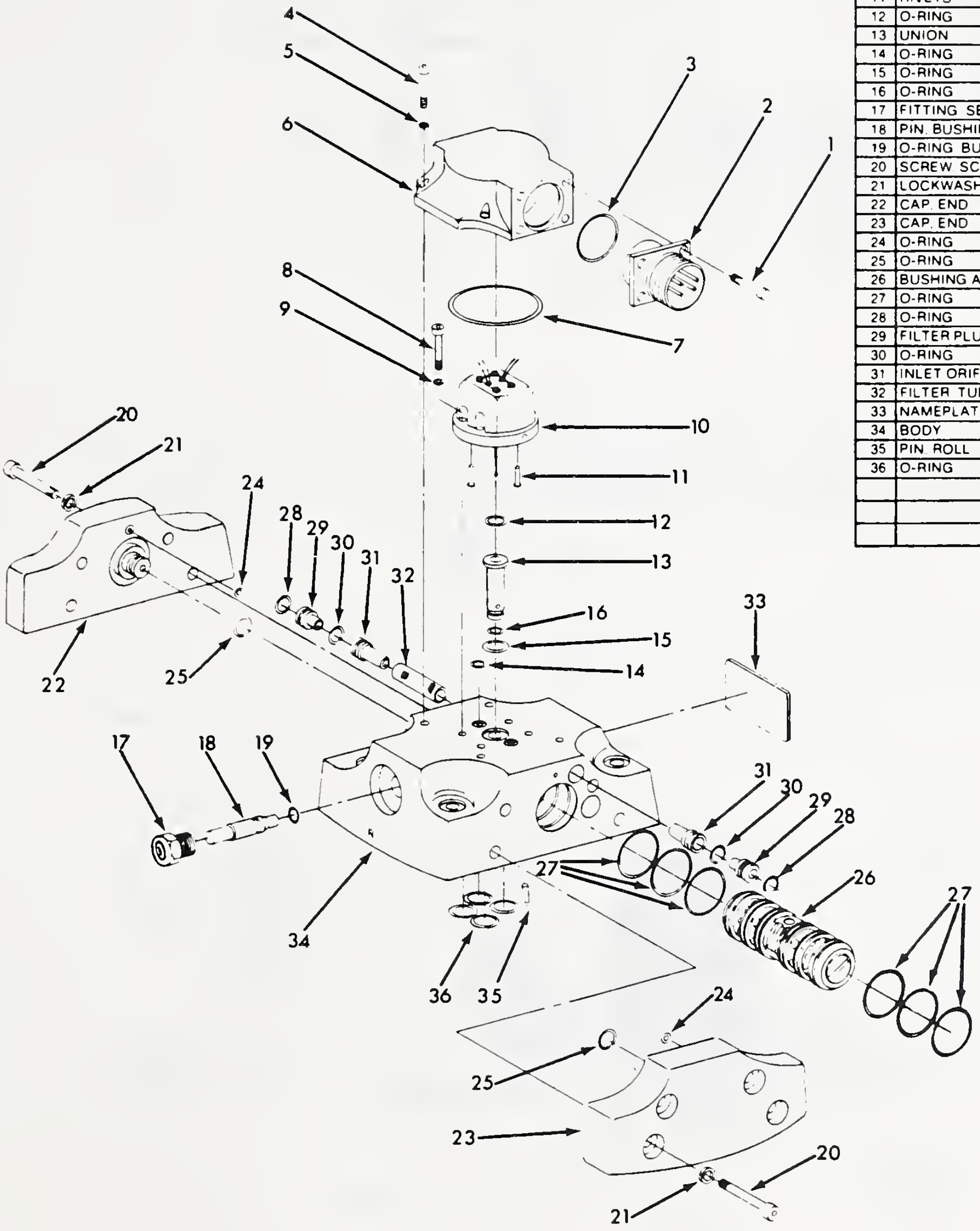


Fig. 1.1: Typical electrohydraulic servovalve and its components.

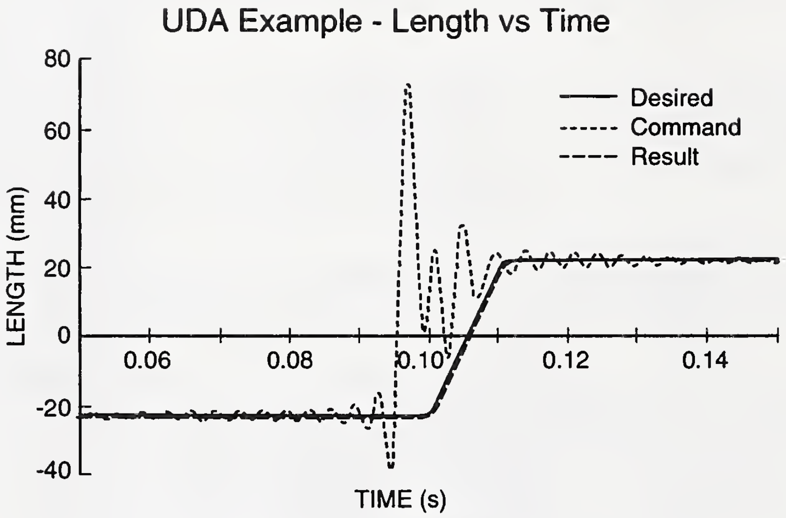


Fig. 1.2: Limiting electrohydraulic (servohydraulic) performance characteristics.

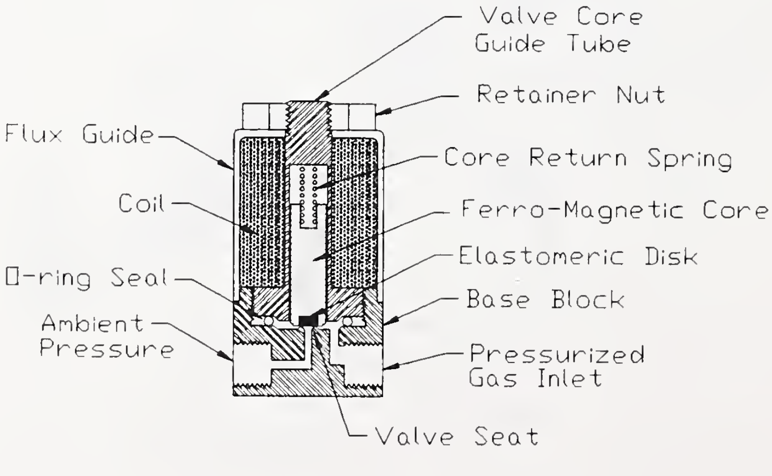


Fig. 1.3: Cross sectional view of typical pneumatic solenoid valve.

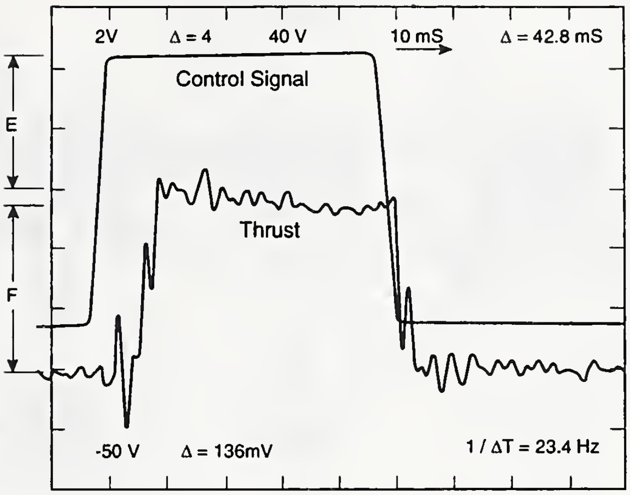


Fig. 1.4: Typical calibration data for air-jet thruster magnitude and time delays reported by Hallauer and Smith. The input voltage was a 10-Hz, 20-mV peak sine wave.

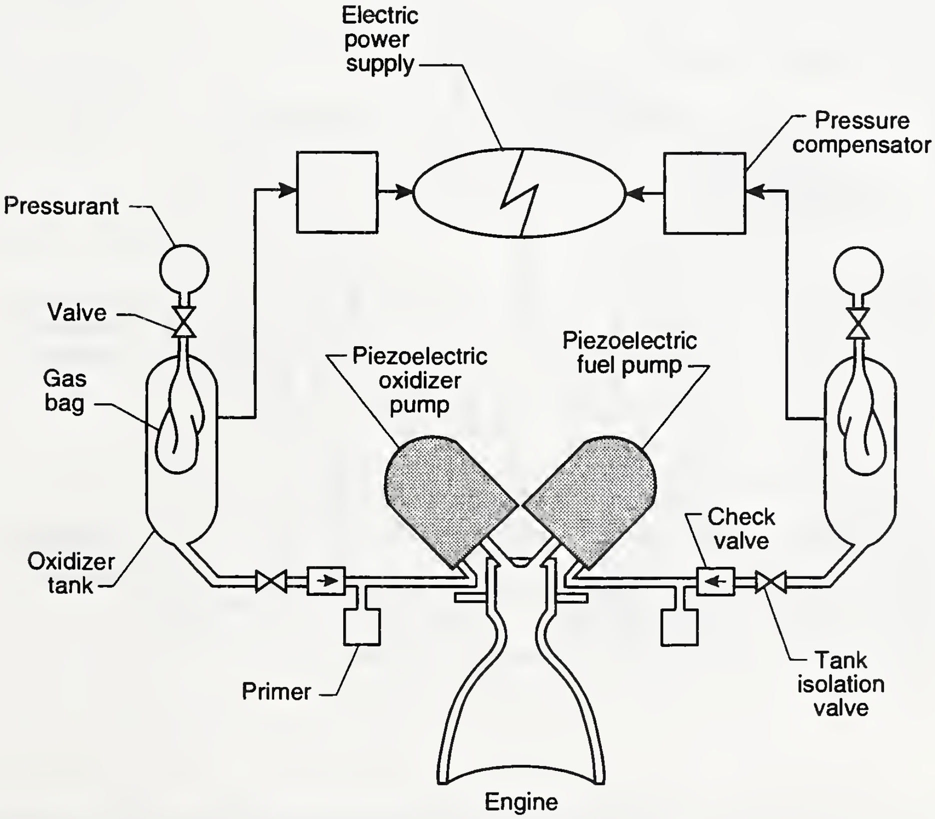


Fig. 1.5: Piezoelectric vernier thruster system developed by Katchee.

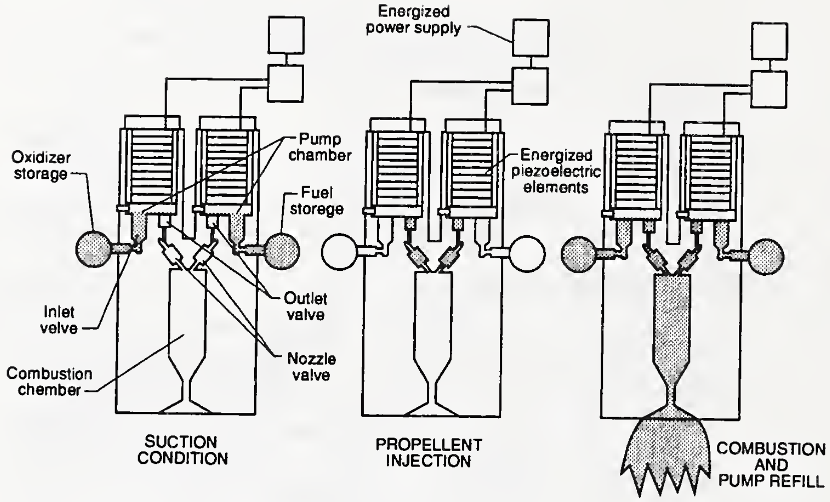


Fig. 1.6: Injector operating sequence used for piezoelectric vernier thruster system.

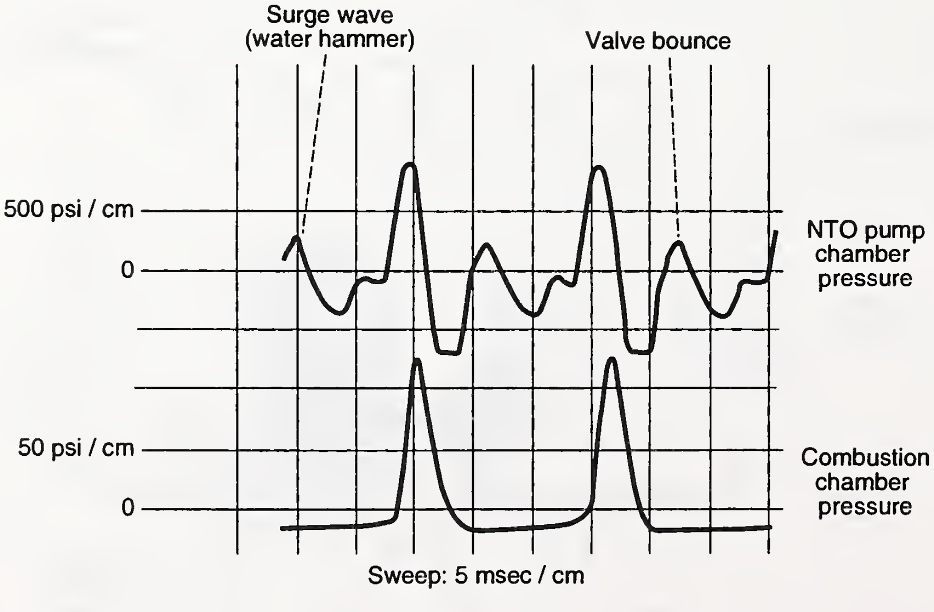


Fig. 1.7: Sample pressure traces from tests performed by Katchee on a piezoelectric vernier thruster system.

Component Designations:

1) Valve body
2) Nozzle body
3) Nozzle member
4) Storage space
5) Valve seat
6) Valve stem
7) Closing piece
8) Connecting pipe section
9) Membrane
10) Drive space
11) Drive body
12) First thermally and electrically insulating body.
13) Second thermally and electrically insulating body.
14) Elastic element
15) Adjustment device
16) Casing
17) Flange connection
18) Pump nozzle
19) Bore
19a) Bore
20) Piezoelectric drive

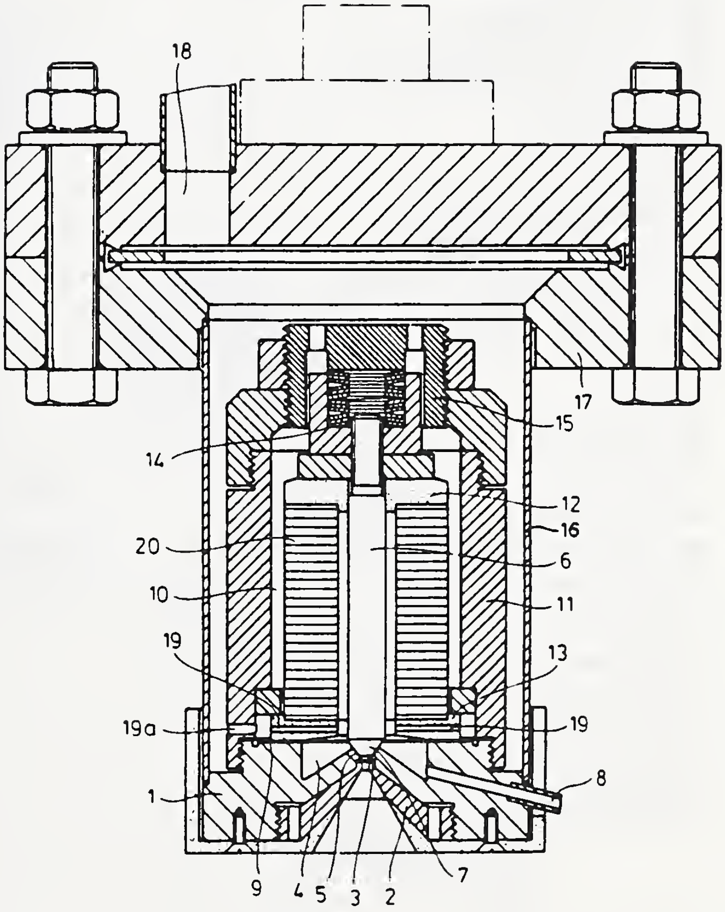


Fig. 1.8: Piezoelectric pulse valve developed by Weber et.al. for injection of trace elements into atomic beam spectrophotometers.

Component Designations:

11) electrostrictive valve
12) hollow cylindrical body
13) external threads
14) orifice
16) outer cavity
17) metal seat
18) apertured adjusting nut
19) axial threaded nut
21) cylindrical stem
22) convex surface
23) internal screw threads
24) seal nut
26) radial tap
27) radially extending guide screw
28) spacer
29) longitudinal slot
31) circumferential groove
32) o-ring
33) enclosure
34) port
36) piezoelectric disks
37) thin copper disks
38) circular metal top plate
39) circular metal bottom plate
41) connecting tab
42) wire
43) insulated wire
44) passage
45) ground wire
46) registering passage
47) step washer
48) washer
49) washer
51) circular plug
52) metal tip
53) annular spring washer
54) flange

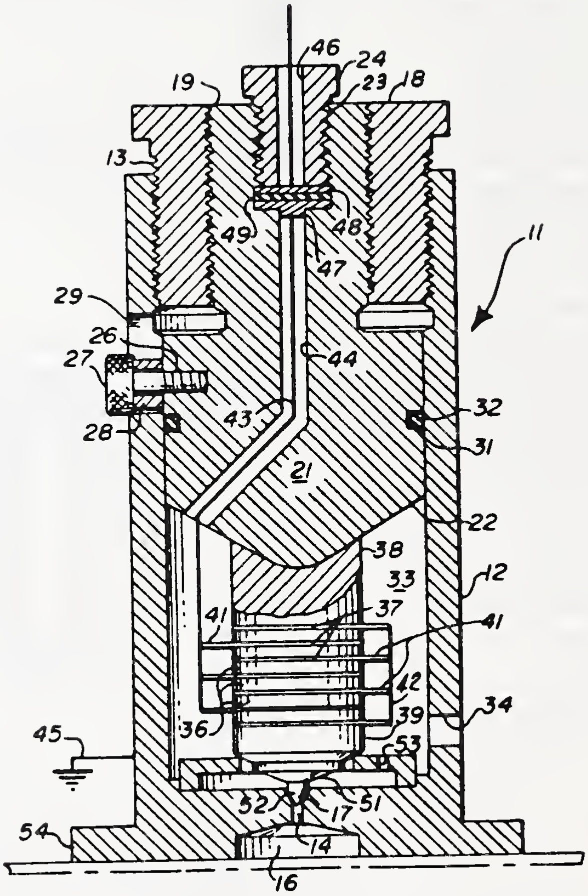


Fig. 1.9: Piezoelectric pulse valve from Kippenham for injection of deuterium gas into plasma field.

Component Designations:

104) transducer
104a) transducer flange
105) electrostrictive detection device
106) horn
106a) injection path
106b) ball valve seat
106c) dispersion cup
107) electrostrictive device
107') electrostrictive elements
108) padding
109) nut
110) positive electrode
111) negative electrode
112) monitor electrode
121) lead wire for 110)
122) lead wire for 111)
125) lead wire for 112)

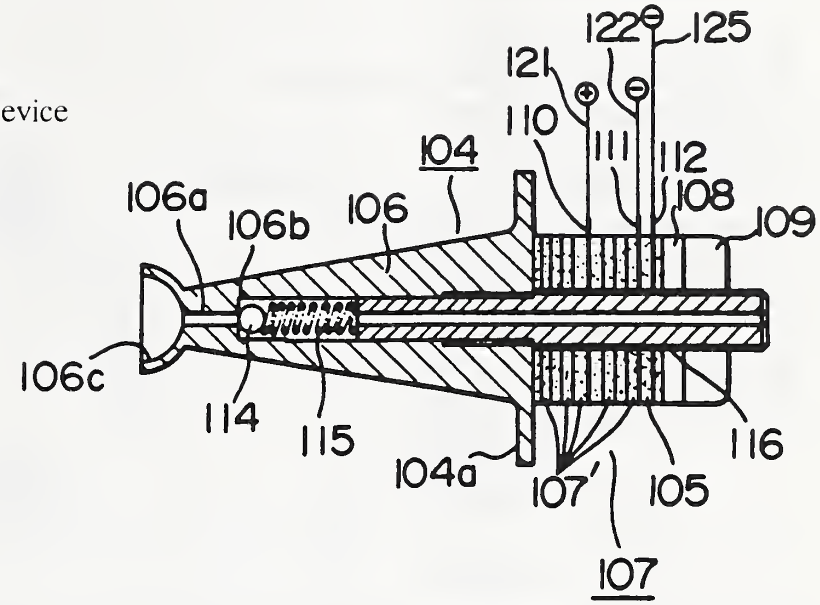


Fig. 1.10: Piezoelectric fuel injection valve from Yamauchi et.al.

Component Designations:

1) piezoelectric device
2) piezoelectric stack elements
3) mounting
4) housing
5) operating pin
6) operating lever
7) pivot
8) carrier
9) engine block
10) lever arm connection
11) actuating rod
12) valve

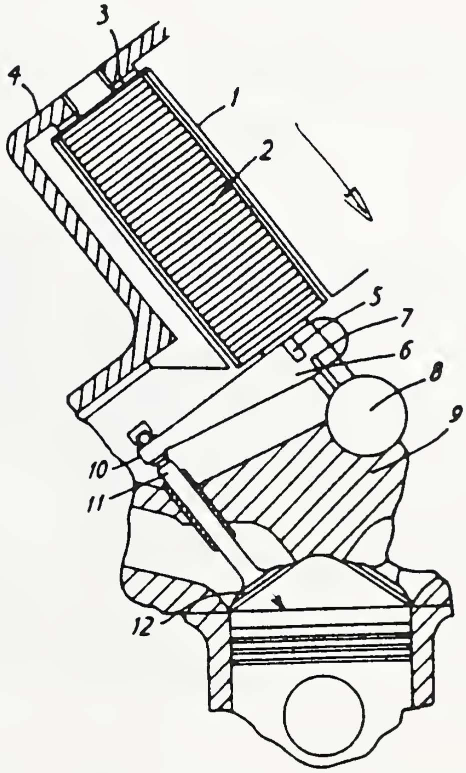


Fig. 1.11: Piezoelectric valve operating mechanism for internal combustion engines from Moloney.

Component Designations:

10) flexure arm/support
11) bottom surface of actuator support
12) piezoelectric stack #1
13) piezoelectric stack #2
14) hinge support
16) top surface of actuator support
17) milled hinge
18) pin
20) actuator element
22) support frame
23) milled hinges

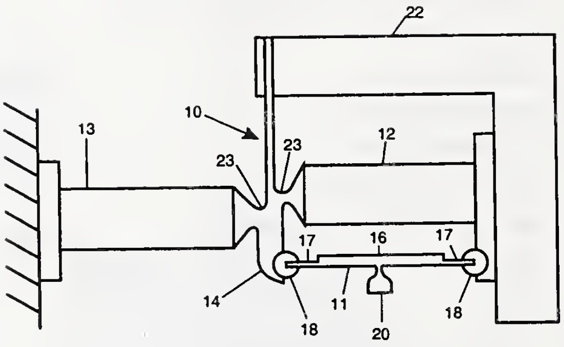


Fig. 1.12: Operating mechanism for high speed reciprocating machine parts (printers) from Knappe.

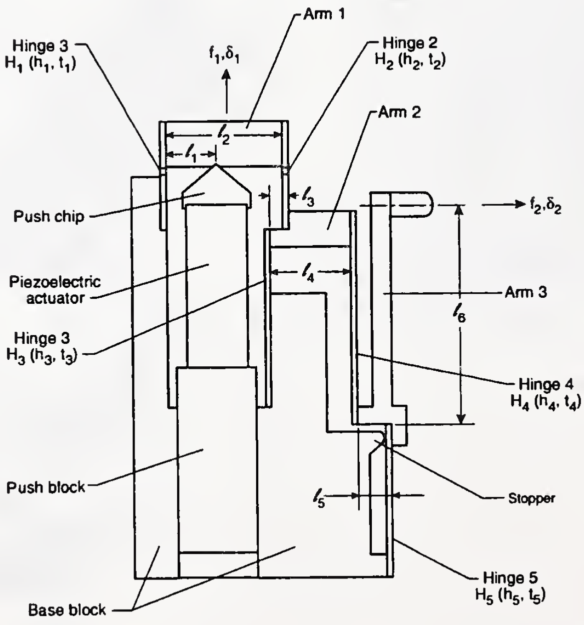
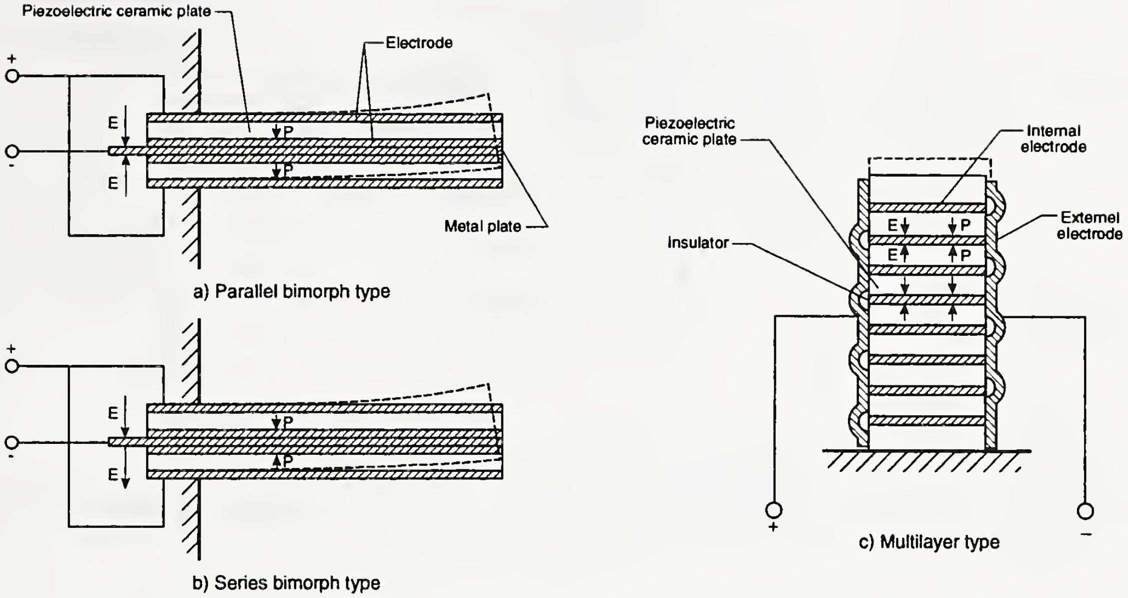


Fig. 1.13: Piezoelectric mechanical displacement amplifier used in the development of a piezoelectric relay from Mitsuhashi et.al.



a) Parallel bimorph type

b) Series bimorph type

c) Multilayer type

Fig. 1.14: Principle piezoelectric actuator architectures (from Mitsuhashi, et.al.)

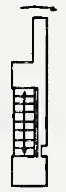
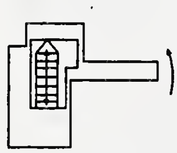
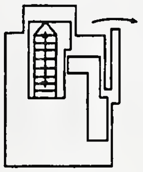
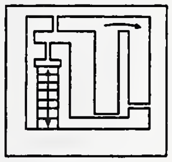
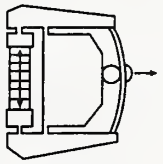
Item	Pushdown type	One-stege lever type	Two-stege lever type	Compound type	Spring deflection type
Shape					
Structure	Simple	Reletively simple	Complex	Complex	Reletively simple
Manufectur-ebility	Good	Good	Reletively good	Poor	Reletively good
Mafnifying factor (times)	20 - 30	30 - 40	50 - 70	50 - 70	30 - 50

Fig. 1.15: Characteristics of several piezoelectric displacement amplification mechanisms (from Mitsuhashi et.al.)

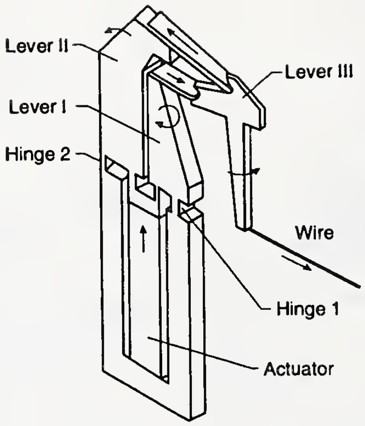


Fig. 1.16: Piezoelectric stack operated impact matric printer head element employing a differential two stage hinge-lever (from Takahashi).

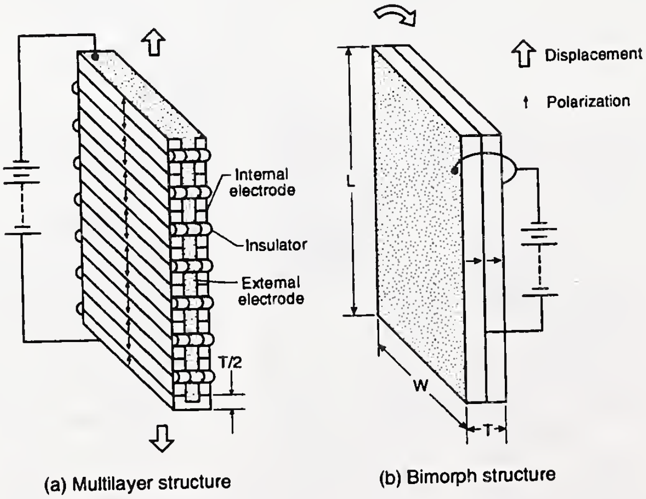


Fig. 1.17: Architecture of piezoelectric stack and bimorph elements (from Takiahashi).

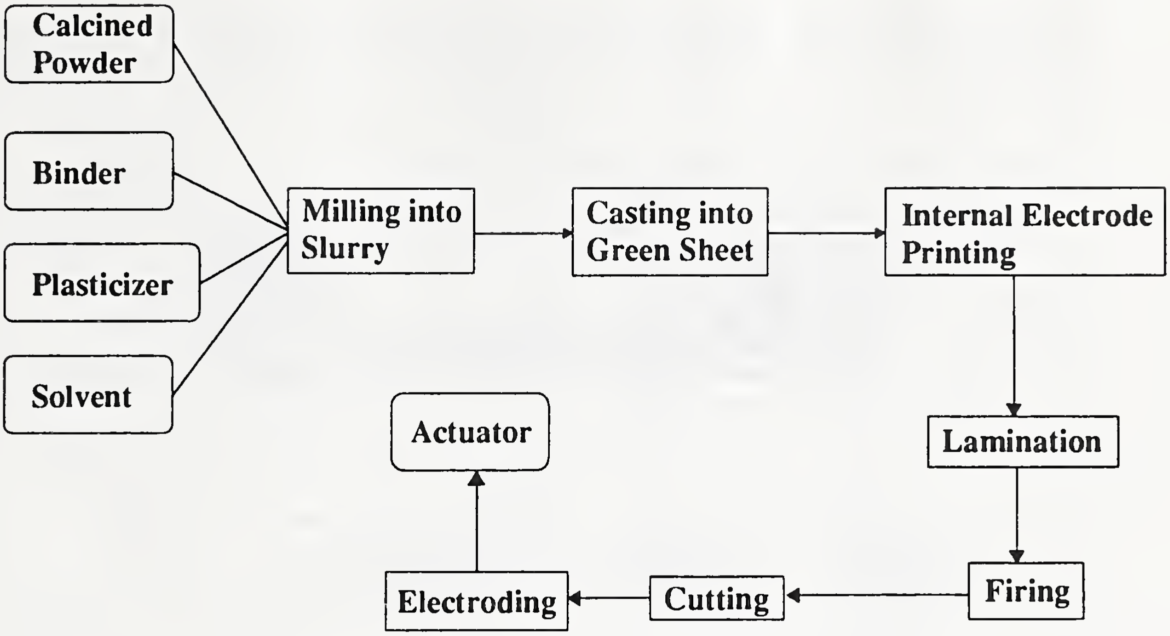


Fig. 1.18: Process flow chart for multilayer piezoelectric ceramic actuators made by the tape-casting method (from Takahashi).

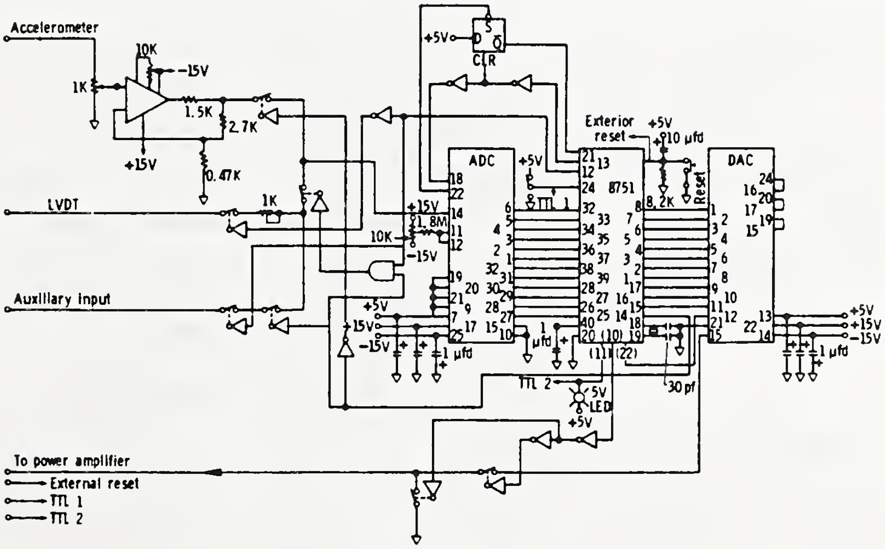
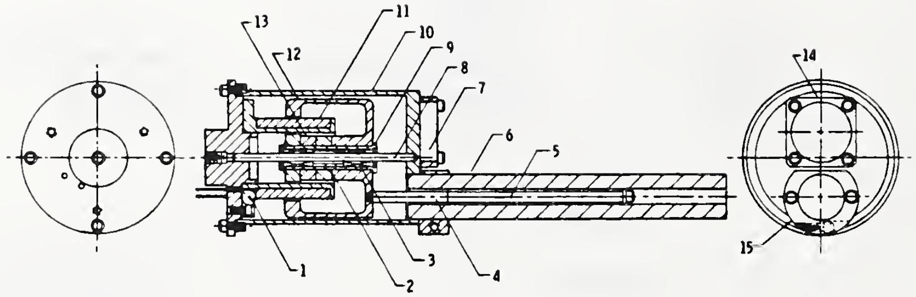


Fig. 1.19 Circuitry schematic of the digital onboard controller developed by Zimmerman et.al. for a harmonic proof-mass actuator.



1. Fixed Coil Base
2. Bearing Spacer
3. Ball Bushing
4. LVDT Connector
5. LVDT Core
6. LVDT Sleeve
7. Accelerometer Adapter
8. Steel Shaft
9. Bearing Sleeve
10. Outer Case
11. Coil Winding Assembly
12. Flux Loop
13. Rare Earth Magnet
14. Accelerometer Adapter
15. LVDT Clamp

Fig. 1.20: Schematic and major parts list for Zimmerman's proof-mass actuator.

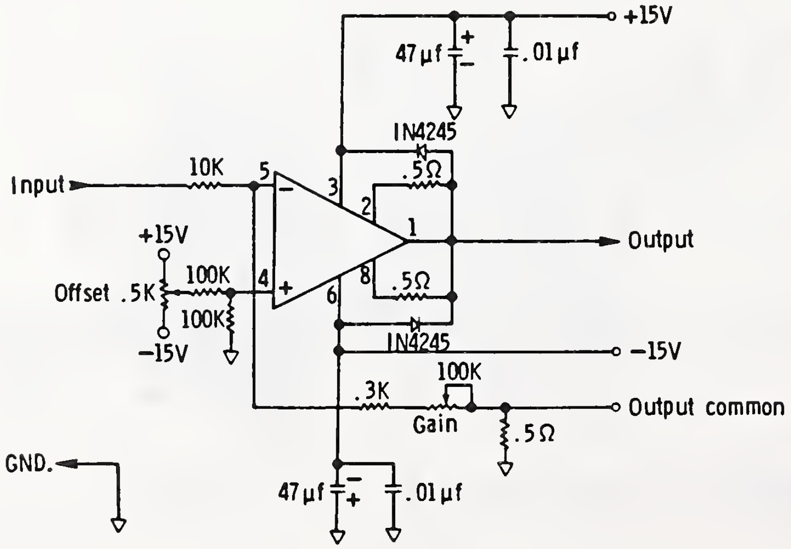


Fig. 1.21: Circuit schematic for Zimmerman's power amplifier used to drive the proof-mass actuator.

Table 1.1: Characteristics of Piezoelectric Ceramic Actuators (from Mitsuhashi et.al.)

Category		Bimorph Element	Multilayer (Stack) Element
Displacement Mode		Unstiffened strain	Stiffened strain
External Dimensions (LxWxH), mm.		25x10x0.17 (two layers)	3x2x9
Thickness of lamination		170 um	60 um
Number of laminations		2	128
Capacitance		140 nF	580 nF
Volume		85 mm^3	54 mm^3
Output (at 60 vdc)	Displacement: Force: Energy:	280 um 18 g 250 erg	7.8 um 21,000 g 8,000 erg
Allowable Electric Field		400 V/mm	>2000 V/mm
Polarity		yes	none
Creep (96H)		>30%	<5%
Shift Displacement (96H)		>30%	<2%

Table 1.2: Calculated actuator characteristics for a multilayer piezoelectric ceramic stack actuator and a bimorph actuator (from Takahashi).

Structure	Electr. Energy (UE) [x10^{-8}J]	Stored Mech. Energy (UM) [x10^{-8}J]	Electro-mech. coupling factor (UM/UE) %	Induced Displ. (um)	Clamping Force (newtons)	Resonant Freq. (kHz)
Stack	1927	891	46	5	35	89
Bimorph	1927	125	6.5	138	0.18	0.59

2.0 Design of a High Speed Intelligent Loading System (HSILS)

2.1 Concept Development

At the inception of the HSILS project the question was posed as to whether a forcing function (thrust-time history) could be achieved which approximated an inverse Gaussian pulse, particularly one of very short duration. An example of such a pulse is shown in Fig. 2.1 along with its governing equation. The rate at which the pulse decays from its maxima, F_{max}, is determined by the shape coefficient, σ. None of the devices described in Chapter 1, in their present state, is capable of reproducing the force-time history described in Fig. 2.1. If a reaction control system is to reproduce a pulse similar to that shown in Fig. 2.1, it must be capable of tracking rapid, pre-programmed or real-time generated command amplitude signals. Such a capability requires local computer control (or "intelligence") at the actuator in addition to a fast-reponse electro-mechanical thrust activation system capable of achieving amplitude control and making amplitude changes very rapidly.

At issue are several performance factors which are defined graphically in Fig. 2.2. These factors define the various command and response terms associated with a reaction control system in which a half square wave command pulse is issued. There is usually a lag time associated with both the rising and falling edges of the thrust pulse, relative to the rising and falling edges of the command signal. In a real reaction control system (as would be recorded by onboard accelerometers or force sensors) there will also generally be some "ringing" oscillations associated with the rising and falling edges of the thrust pulse as well as superimposed noise.

The rise lag (RL in Fig. 2.2) is defined as the elapsed time from the leading edge of the drive (or control) pulse to the initiation of measurable thrust. This is a measure of the electromechanical efficiency of the actuator. Rise lag time is associated with mechanical response times of, for example, electrostrictive elements (piezoelectric stacks); wave propagation through connecting elements of a displacement amplifier; the presence of various sources of friction and viscous damping; and reaction times for fuel constituents (if a bi-propellant is used instead of a compressed gas).

The response pulse width (RPW) is defined as the base-to-base time of the response thrust profile, which equals the elapsed time from thrust initiation to thrust termination for an actuator responding to the minimum amplitude half square wave command signal required to initiate measurable thrust. The RPW is a useful measure of the absolute shortest impulse that can be produced by a reaction control system.

If one views rise lag as electromechanical propagation delay, then one might expect the response pulse to terminate at a time interval equal to DPW (the drive pulse width) following initiation of measurable thrust from the actuator. In actual fact this proves not to be the case. The difference in elapsed time from initiation of measurable thrust to the point at which thrust commences a marked decrease minus the DPW constitutes the parameter known as fall lag. For inductive type systems (e.g. solenoid coils) this results from delays in the collapse of the magnetic field. It can also result from mechanical delays caused by having to overcome static friction in seals.

A final performance parameter is the amplitude control rate (ACR), which is a measure of the maximum thrust slew rate (or rate of thrust rise per unit time) to a unit change in command signal. Typically this would have units of N/s and could be estimated by dividing the peak thrust by the rise time (RT, in Fig. 2.2). In the case of a solenoid operated thruster, as described in Chapter 1, there are only two states that can be achieved: full-off or full-on, and thus the ACR rating is somewhat inappropriate. One could construct an array of axially aligned thrusters of this type and drive them in parallel through the output pins on a microcontroller such that each thruster could be discretely turned either on or off. In this manner, crude amplitude control could be achieved, and the ACR would simply be the unit thrust change divided by the rise time to reach that thrust level. This equivalent ACR will be used later in Chapter 4 to compare performance of solenoid and HSILS actuators.

2.1.1 HSILS Overview

Briefly, the NIST HSILS loading system incorporates a very high speed, electronically controllable valve system which is capable of continuously metering the flow of pressurized gas and/or pressurized propellants through an expansion nozzle to create thrust. The device may be envisioned in a simplified way as shown in Fig. 2.3, in which a valve core element seats directly against the throat of the expansion nozzle. Axial (vertical) displacement of the valve core, under the command of an embedded microcontroller, causes a gap to form which permits mass flow from the pressurized fuel container to the expansion nozzle. As long as the annular area determined by the gap between the valve core and the valve seat remains less than the nozzle throat area linear thrust control is possible. Maximum thrust is thus determined by the nozzle throat diameter.

Where the expansion of pressurized gas or of a monopropellant is the thrust-producing factor the high pressure valve seat is integral with the throat of the expansion nozzle (as shown in Fig. 2.4). If the combustion of hypergolic bipropellants is the thrust-producing factor the valve seat communicates with an injector tube. The injector tube communicates with a combustion chamber and is designed such that the separate flows of the two propellants impinge upon on another inside the combustion chamber in such a manner as to promote complete and uniform burning prior to expansion of the reactants through the exhaust nozzle.

The heart of the high speed valve is a monolithic (or "co-fired") piezoelectric (or electrostrictive) stack. A custom stack measuring 76 mm long by 9.5 mm square was produced specially for the project by Piezo Systems of Cambridge Massachusetts. This stack was capable of producing 0.2% strain when subjected to an electric potential of approximately 300 vdc (Volts, direct current). This strain is essentially fully developed within 10 microseconds of the application of a step change in voltage to the stack and therefore allows for extremely rapid displacement control, provided an appropriate voltage-time history can be applied to the stack. Electrostrictive characteristics, including blocked force response, of this piezoelectric stack are presented in Fig. 2.5. A blocked force response chart shows the relationship between the maximum force level capable of being developed by the piezoelectric element between a pair of rigid reaction surfaces at a specified level of total expansion when the element is driven at a constant voltage. There exists a family of blocked force curves for any piezoelectric element with one curve for each unique drive voltage. The relationship between force and displacement at any given drive voltage is typically linear. However, the force (y-axis) intercept may not be linearly related to the drive voltage, hence (e.g. Fig. 2.5) two separate equations of state are required to characterize the stack.

The performance characteristics described above and in Fig. 2.5 can be considered representative of the best available today for commercial grade piezoelectric materials. However, even with a relatively long, hybrid stack 0.2% strain is insufficient for operation of an effective rocket engine feed line valve. To be practicable, a strain of approximately 1-2% would be required for direct axial operation of a thruster capable of producing a force amplitude sufficient for active control purposes.

Accordingly, the device employs a specially designed mechanical displacement amplifier which produces as its output a displacement which is opposite in direction to the expansion (or contraction) of the piezoelectric stack and larger than the absolute magnitude of the piezoelectric stack displacement. The amplifier consists of a stiff eccentric anvil (see Fig. 2.6) which is monolithically milled from the same base block, and is integral with, a support means. The anvil is connected to the support means by a finely milled hinge (see Appendix E, Fig. E.10) that is integral to both the anvil and the support means and comprises a structural element that provides negligible resistance to rotation about an axis perpendicular to the plane of the anvil. The hinge is also designed to provide stiff resistance to displacements and rotations perpendicular to its rotational axis. The anvil is designed to provide, as nearly as practicable, a rigid lever, the fulcrum of which is provided by the hinge.

The piezoelectric stack is provided with special low friction, high strength bearings at both ends which insure axial transmission of force as the stack expands under increasing voltage. One such bearing forms an interface with an adjustable, threaded nut which serves as a moveable reaction surface within a housing. The bearing means at the output end of the piezoelectric stack forms an interface with the short side of the lever/anvil of the displacement amplifier.

The output of the displacement amplifier is connected to a valve core by a similar low friction, high strength bearing such that as the piezoelectric stack expands under increasing applied voltage the valve core lifts off its corresponding valve seat and permits high pressure gas and/or propellant to flow. In order to achieve continuous control, however, the valve core must be capable of being returned to the closed position under positive control. This is achieved by use of a suitable adjustable force compression spring means which maintains the valve in a normally closed position. The spring force is designed to insure that the closing acceleration of the valve core is capable of matching the un-amplified contraction acceleration of the piezoelectric stack in such a manner as to maintain all of the bearing in a state of compression.

The valve core is constrained to move in a linear fashion by various guides and seals which additionally serve to prevent buckling of the rod during the application of compressive loads and to damp out unwanted vibrations. The valve seal is achieved by means of a metal-to-metal contact between the end of the valve core, which is tapered at an angle (typically 59° inclusive), and which mates to the valve seat which is also tapered but with an angle (typically 60° inclusive) that is slightly larger than that for the valve core (see Fig. 2.4).

In one form of the device, in which a cold gas thruster is to be provided, said valve seat is integral with the throat of an expansion nozzle. A chamber on the interior side of the valve seat is in communication with a high pressure reservoir of gas (e.g. nitrogen). The expansion of this gas through the nozzle as the valve core is raised from and lowered to the closed position creates thrust which is directly proportional to the displacement of the valve core, and therefore directly proportional to the control signal (voltage-time history) used to drive the piezoelectric stack. The actual response "speed" (delay time) between a step change in control voltage to the piezoelectric stack and the corresponding change in thrust level at the nozzle is determined by the response time of the piezoelectric stack and the propagation speed of the resulting stress wave to the valve tip. This response speed can be shown through structural dynamics calculations to be substantially better than for any of the aforementioned alternative loading systems.

In a second form of the device, in which a bi-propellant, pressure fed thruster is to be provided, the valve seats are located at one end of propellent delivery tubes which are in communication with injection ports within a combustion chamber. A chamber on the interior side of one valve seat is in communication with a reservoir of propellent (e.g. monomethyl hydrazine), while a chamber on the interior side of the second valve seat is in communication with a reservoir of oxidizer (e.g. nitrogen tetroxide). Both propellent and oxidizer reservoirs are pressurized by means of expansive bladders which are in communication with respective high pressure inert gas supplies. These expand the bladders as fuel and oxidizer are used, and thus permit fuel to be delivered independently of attitude. The expansion of the combustion product gases through the nozzle, which occurs when the valve cores are raised from and lowered to the closed position creates thrust which is directly proportional to the displacement of the valve core, and therefore directly proportional to the control signal (voltage-time history) used to drive the piezoelectric stack.

In this report, a prototype cold gas thruster version of HSILS was built and tested using compressed nitrogen as the "fuel".

2.2 Hardware and Software Design

2.2.1 HSILS Mechanical Description (Operation)

Because of the mechanical complexity of the HSILS, and because of the density of components contained in the figures associated with the following text, numbered callouts have been used to designate particular parts rather than actual text labels.

Referring to Fig. 2.7, a housing, 24, is provided as a means for anchoring the device to the structure to be controlled or tested, and for providing internal reaction points for its moving parts. Housing 24 is machined from a solid block of a high strength, lightweight engineering material (such as aluminum alloy) or otherwise fabricated by such processes as centrifugal casting, injection molding or the like, as a monolithic unit containing the functional cavities which will be explained in detail below.

Mechanical action within housing 24 is initiated by sending a direct current voltage signal via electrical connectors 80 and 81 to piezoelectric stack 1 which resides inside cavity 34 and which is axially oriented within and parallel to cavity 34. Integral bearing supports 2 and 82 are affixed permanently to both ends of the piezoelectric stack. These consist of machined blocks of a high strength material such as hardcoated aluminum alloy and contain a machined, tapered circular bearing hole to which are affixed spherical bearings 3 and 83. Bearings 3 and 83 should be made of a tough, lightweight material with low friction characteristics such as a ruby-sapphire sphere. These spherical bearings are used to insure that all load is transmitted axially through the piezoelectric stack to the anvil-lever 4 and to allow for rotation of the anvil-lever 4 about its hinge 5 without the consequent development of significant resistance to such motion at bearing 3.

The top bearing of the piezoelectric stack mates with a similar tapered circular bearing hole 90 in the base of a threaded nut 8 fabricated from a high strength material such as hardcoated aluminum alloy. The bottom bearing of the piezoelectric stack 3 mates with a similar tapered circular bearing hole 88 on the top surface of the short side of the anvil-lever 4 which comprises the moving part of displacement amplifier 6. Threaded nut 8 is used to adjust the vertical position of the piezoelectric stack with respect to anvil-lever 4 so as to insure that it is in compression contact with anvil-lever 4 at all times during the operation of the device.

Displacement amplifier 6 is comprised of a stiff, anvil-shaped lever 4, a hinge 5 and a support 6 which is also used to designate the overall amplifier. The amplifier 6, for reasons which will be subsequently described in detail, is fabricated from a very high strength material having a low modulus of elasticity. The displacement amplifier 6 is bolted to housing 24 and rests within cavity 36 such that the plane which is perpendicular to the axis of rotation of and lies on the centerline through the thickness of the anvil-lever, also lies on the centerline through the thickness of housing 24. Furthermore, the bottom bearing 3 of the piezoelectric stack 1, which contacts the short side of the anvil-lever 4 and the top bearing 83 of the piezoelectric stack are so positioned so as to orient the longitudinal axis of the piezoelectric stack in an initially perpendicular orientation with respect to the plane which comprises the top surface of anvil-lever 4.

A third spherical bearing 22 is provided at the end of the long side of anvil-lever 4 and is in compressive contact with a tapered circular bearing seat 87 which has been machined into the top face of an interior slot in core rod coupler 20, through which the long end of anvil-lever 4 passes. Core rod coupler 20 is fabricated from a high strength lightweight material such as hardcoated aluminum alloy and contains threaded holes 85 and 89 at its top and bottom to which are affixed guide rod 52 and valve core rod 19.

Valve core rod 19 is normally held in contact with valve seat 11 by means of compressive spring 23 which reacts against threaded rod 25 which is in turn screwed into a threaded borehole 35 on the centerline through the thickness of housing 24. Threaded nut 25 is used to adjust the level of precompression exerted by the valve core rod 19 on valve seat 11. Normally this level of precompression is that required to ensure that no leakage of high pressure gas and/or propellant occurs at the valve seat 11. Guide rod 52 connects to threaded hole 85 on the top of core rod coupler 20 and passes through the center of compression spring 23 and into guide hole 84, which is machined through the centerline of threaded nut 25. Guide hole 91 is machined from housing 24 and communicates with borehole 35 and pressure chamber 14. The purpose of guide holes 91 and 84 is to insure that only axial motion which is perpendicular to the plane of the top face of anvil-lever 4 is possible in the spring-loaded core rod assembly, which consists of guide rod 52, core rod coupler 20, and valve core rod 19.

Guide rod 52 and valve core rod 19 should be fabricated from a very high strength, high hardness material such as Vascomax 240 or similar high strength, hardened steel alloys.

In the embodiment of the device depicted in Fig. 2.7, where the latter is to be operated as a cold gas thruster, onboard high pressure gas reservoirs 10 are connected by high pressure fittings 29 and high pressure tubing 26 to form a single reservoir which is connected directly to pressure chamber 14 by means of fittings 33, 15, and high pressure tubing 13. It is presumed that gas reservoirs 10 are charged prior to using the device by means of an external gas supply which is subsequently disconnected prior to use of the system, so that the device can be operated autonomously from any umbilicals.

Pressure chamber 14 is in communication with the high pressure gas supply via fitting 15 which is a threaded fitting of common availability capable of being sealed in such a manner as to prevent any loss of gas either through the threaded connection into housing 24, or through the junction with supply tube 13. Since pressure chamber 14 is also in communication with borehole 35, which is unpressurized, via guide hole 91 a high pressure seal must be provided at this location which both permits axial motion of core rod 19 to take place while simultaneously preventing leakage of gas through guide hole 91. This is accomplished by means of radial o-ring 18 which seals valve core rod 19 against plate 16. Plate 16 is consequently sealed against housing 14 by means of face o-ring 18. Both o-rings, in combination, achieve the necessary seal for pressure chamber 14 while permitting axial motion of valve core rod 19. Pressure chamber 14 is furthermore in communication with expansion nozzle 12. The throat 11 of expansion nozzle 12, as was previously described, forms the high pressure valve seat for valve core rod 19. One possible method of fabricating the housing is to bore pressure chamber 14 from one side of housing 24. For this embodiment of the device, expansion nozzle 12 may be fabricated separately and subsequently bolted to housing 24 by means of fasteners 9. Sealing of the expansion nozzle assembly 12 to housing 24 may be achieved by means of a radial o-ring 39, as shown in Fig. 2.10.

Figs. 2.8 and 2.9 illustrate side and top views of the cold gas thruster embodiment of the device which show, in addition to those features described in Fig. 2.7, one possible means for mounting an onboard microcontroller subsystem 30 and an energy storage and amplification subsystem 32. These are affixed to opposing sides of gas reservoirs 10 by means of special brackets 31 which are connected to housing 24 by means of fasteners 40. It is further illustrated in Figs. 2.8 and 2.9 that boreholes 24 and 35 for the piezoelectric stack and for the valve core assembly, respectively, are machined in such a manner as to lie on the plane which passes through the centerline of the thickness of housing 24. Bracket 21 shown in Fig. 2.9 is provided with threaded holes which may be used to mount the device on the structure to be controlled and/or tested.

Figure 2.10 shows an extracted assembly of the internal working parts of the device, while Fig. 2.11 shows a detailed cutaway of pressure chamber 14 and its associated interfaces. During operation of the device a positive direct current voltage will cause piezoelectric stack 1 to expand. Since the stack 1 is connected to a rigid internal reaction surface at bearing seat 90, all expansion occurs at the lower end and is transferred via bearing 3 to tapered bearing seat 88 which is machined on the short side of anvil-lever 4. The vertical displacement of piezoelectric stack 1 therefore causes bearing seat 88 on anvil-lever 4 to move downward and for anvil-lever 4 to rotate about hinge 5. Consequently, and as will be explained in detail below, bearing 22 on the long side of anvil-lever 4 is caused to move upward, which is precisely the direction needed in order to unseat valve core tip 48 from valve seat 11 and initiate loading as gas is expanded through nozzle 12. This inversion of displacement direction is required, since the piezoelectric stack 1 is unable to effectively transmit loads in tension. Furthermore, a positive return mechanism must be available to close valve seat 11 for the same reason. This is accomplished by means of precompression spring 23 which is chosen so that the precompression force satisfies two criteria: First, it must be capable of sealing valve seat 11 so that no gas and/or propellant is permitted to escape through expansion nozzle 12 when the piezoelectric stack is in the uncharged state. Secondly, the precompression force must be capable of accelerating anvil-lever 4 and the valve core rod assembly (52,20, and 19) downward towards a closed valve position in such a manner as to always exceed the contracting acceleration of piezoelectric stack 1 as the voltage signal which drives the stack is diminished. This is required in order to maintain all bearings (83,3, and 22) in a state of continual compression; otherwise the possibility of separation occurs leading to subsequent shock loading and noise in the "intelligent" impulse.

Figure 2.11 shows a detailed view of pressure chamber 14. In the preferred embodiment of the device this chamber is milled from housing 24 as a circular borehole. Radial o-ring 39 and backup ring 47 (optional) provide for a high pressure static seal between nozzle 12 and the circular wall of chamber 14. Nozzle 12 is bolted to housing 24 by fasteners 9 which are sized to resist the resultant force acting to expel the nozzle from the housing due to the high internal pressure inside pressure chamber 14. Plate 16 contains two sets of o-ring seals which prevent gas from escaping via borehole 91, through which the valve core rod enters chamber 14. Face o-ring seal 18 is precompressed against housing 24 by plate 16 by means of fasteners 94 and prevents escape of gas around plate 16 and into borehole 91. O-ring seal 17 and optional backup rings 51 prevent escape of gas through borehole 95 in plate 16, through which the valve core rod passes. Seal 17 additionally serves as a viscous damper for valve core rod 19, which serves to minimize elastic rebound when the valve core rod tip 48 closes with valve seat 11.

During normal operation of the device, upward motion of valve core rod 19 will permit high pressure gas to expand through nozzle 12 and thus create thrust 50. The high pressure gas and/or propellant feed supply 49 to chamber 14 is delivered through fitting 15 and gas supply tubing 13, both of which must have significantly larger internal diameters than the throat diameter 103 of the expansion nozzle 12, as shown in Fig. 2.12. This ensures that fully developed flow is maintained at all times when the valve is open.

Referring now to Fig. 2.12, which is a general detail of the valve core tip and valve seat, valve core rod mating surface 48 is precision ground to have a slope 99 of 29.5 degrees, and thus an included angle of 59°. Core rod tip 100 is precision ground to a flatter angle 101 of approximately 30° from horizontal (120° included angle) so as not to come into contact with the valve seat. Valve seat 11 is machined with a tapered angle 98 of 30°, or a 60° included angle. Since angle 98 is slightly larger than angle 99, initial sealing of the valve occurs at the expansion nozzle throat 103.

Figure 2.13 presents a detailed side view of the displacement amplifier which is used to multiply the movement from the piezoelectric stack to a level useful for practical valve control. As previously described, this consists of a rigid anvil-lever 4, a hinge 5, and a support 6. During normal operation of the device, expansion of the piezoelectric stack in response to a controlled direct current signal will in turn exert force 88 on the tapered circular bearing support 88 on the short side of the anvil lever, whose fulcrum is approximately given by hinge 5. In turn anvil-lever 4 will rotate clockwise about hinge 5 and cause bearing support 86 on the long side of anvil-lever 4 to be forced upward, thereby generating force 105 as it reacts against precompression spring 23 (see Fig. 2.7). The upward displacement at bearing support 86 will be, to within four significant figures, equal to the expansion of the piezoelectric stack times the ratio of length 52 divided by length 54.

The tapered shape of anvil-hinge 4 is necessary in order to minimize both the mass of the lever (i.e., in order to reduce inertially induced forces and moments as the piezoelectric stack expands) as well as to minimize flexural deformations relative to the un-deformed top surface of the lever due to the action of applied forces 105 and 104. Hinge 5 is designed to be as thin as possible in order to minimize rotational resistance about an axis perpendicular to the principal plane of the amplifier. The height of the hinge is selected to be as tall as possible while avoiding the possibility of buckling under the combined compressive loads 104 and 105. The hinge width 56 is selected to prevent both bearing failure of the hinge and to inhibit the likelihood of the anvil-lever resonating in a torsional mode. These design criteria lead to the requirement for the amplifier to be fabricated from a very high strength, low modulus material with a relatively long linear elastic stress-strain curve such as titanium alloy. To minimize stress concentration and to alleviate the likelihood of fatigue cracking, hinge 5 is machined or otherwise fabricated with curved interfacing radii 58 between anvil-lever 4 and support 6. Note that hinge width 56 (and also the width of the anvil-lever) is less than the width of the support 55 so that when support 6 is affixed to housing 24 by means of fasteners 103 there will be no interference with the housing wall as the anvil-lever rotates. The amplification ratio of the anvil-lever is equal to length 52 divided by length 54.

Figure 2.14 refers to the second major embodiment of the device which is a pressure-fed bipropellent thruster of a type that can be used for very high precision attitude control of spacecraft. There are several fundamental differences between the operation of the device shown in Fig. 2.14 and that for the cold gas thruster described in Figs. 2.7, 2.8 and 2.9. The primary difference is that the propellent medium consists of two liquids (an oxidizer 111 and a fuel 110), which are stored inside non-reactive chambers 113 and 112, respectively. Chambers 113 and 112 are in turn pressurized by means of high pressure inert gas sources 64 and 61, which are in turn connected by means of delivery tubes 65 and 62 to the respective chambers. Delivery tubes 65 and 62 are further connected to expansive bladders 66 and 63, which expand and pressurize chambers 113 and 112 so as to occupy the volume of space vacant of oxidizer or fuel and to apply uniform pressure to the remaining oxidizer or fuel.

Acting under the influence of the high internal pressure from pressure vessels 64 and 61, oxidizer and fuel are forced to flow through delivery tubes 68 and 67, respectively, which communicate with valve chambers 114 and 115, respectively. Items 64, 65, 66, 111, 113, and 68 comprise the oxidizer pressure-fed delivery system and are defined as a functional unit by 76. Items 61, 62, 63, 110, 112, and 67 comprise the fuel pressure-fed delivery system and are defined as a functional unit by 75.

Functional unit 74 is comprised of all of the basic components described for the high speed intelligent valve system in Figs 2.7 through 2.13 less the cold gas supply system and the expansion nozzle. Unit 74 provides for highly responsive monitoring and metering of oxidizer from valve core chamber 114 through valve seat 116. Unlike the cold gas thruster embodiment of the device, valve seat 116 is in communication with delivery tube 70 which is in communication with combustion chamber 71. A supplementary nozzle 118 may be used to direct and disperse the flow of oxidizer into combustion chamber 71.

Functional unit 73 is comprised of all of the basic components described for the high speed intelligent valve system in Figs 2.7 through 2.13 less the cold gas supply system and the expansion nozzle. Unit 74 provides for highly responsive monitoring and metering of fuel from valve core chamber 115 through valve seat 117. Unlike the cold gas thruster embodiment of the device, valve seat 117 is in communication with delivery tube 69 which is in communication with combustion chamber 71. A supplementary nozzle 119 may be used to direct and disperse the flow of fuel into combustion chamber 71.

A typical oxidizer 111 for use in the device may be nitrogen tetroxide whereas a typical fuel 110 may be mono methyl hydrazine, although any oxidizer and fuel combination which is hypergolic in nature may be utilized. The impingement pattern of nozzles 118 and 119 may be optimized to produce uniform and complete burning of the constituents within combustion chamber 71. The combustion products are then expanded through nozzle 72 to produce thrust. It may be appreciated that, due to the ability to place valve system 74 and valve system 73 under separate microprocessor control, the mix ratio of fuel to oxidizer, as well as the flow rates to combustion chamber 72, can be varied at extremely rapid rates so as to be able to create, on demand, both very small, very rapid control forces, and large, rapid control forces, as determined to be necessary depending on the nature of the dynamic disturbing force which must be damped; an attitude control maneuver which must be initiated; or a pointing requirement which must be maintained with a high degree of precision. The reaction control system embodied in Fig. 2.14 makes possible extremely rapid control response, as well as very high precision in both small and large applied forces.

2.2.2 Onboard Microcontroller Subsystem Functional Overview

The overall problem is to provide a rocket motor with a programmable thrust vs. time profile. The thrust is varied by controlling the position of a fuel valve. The actual burn time is very short: less than one second and the valve must be able to fully open or close within 500 microseconds. To achieve this response the valve is actuated by a piezoelectric transducer which converts voltage (0-300v) into mechanical displacement. A position sensor can be used to provide position data logging or feedback for closed loop control. Fig. 2.15 provides a control system block diagram for the control system developed for the HSILS by Cytek, Inc., North Andover, MA.

The microcontroller will be based on an Intel 8751H-12 microprocessor. This device has 4 Kbytes of internal EPROM program memory, 128 bytes of internal data memory, an asynchronous serial port, two counter/timers, I/O and interrupt lines. Additional components include:

- 32 Kbytes of battery backed RAM to be used for profile and data logging
- A 10 bit Tracking Analog to Digital (A/D) Converter to acquire valve position data. Maximum conversion rate is 500 K samples/sec.
- A 10 bit Digital to Analog (D/A) Converter to generate the position signal for the Power Amplifier.
- Signal conditioning components for discrete input and output lines.

I/O Capabilities:

<u>Input:</u>

1 Analog strain gage input. +/- 5 volt excitation voltage, input range adjustable 0-50 millivolts to 0-200 millivolts. Measurement accuracy of 1%.

1 Asynchronous communications port (this is also an output) conforming to the RS232 electrical standard and operating at 9600 bits/sec.

4 Discrete inputs. An input of 0-.7 volts will be considered a logical 0. Inputs greater than 2 volts will be considered a logical 1. Voltages greater than 0.7 but less than 2 volts will be undefined. The input impedance will be 100K ohms. The maximum permissible input range is +/-35 volts.

2 Interrupt inputs: The electrical specifications are the same as the Discrete Inputs.

Output:

1 Analog output: The voltage range will be 0-5 volts with an output impedance of 1,000 ohms. The accuracy and settling time (to rated accuracy) will be 1% and 2 microseconds, respectively.

4 Discrete Outputs: These are of the open collector type and can sink a maximum of 20 milliamperes. The maximum applied voltage range is 0-35 volts.

The microcontroller employs printed circuit (PC) board construction with design emphasis on compactness and shock resistance. Communications between the microcontroller and the host PC take place over the serial link using a blocked asynchronous protocol as shown in Fig. 2.16. The parameters are 9600 bits/second; 8 bits; no parity; and 1 stop bit. The protocol requires that each packet be acknowledged before the next can be sent. The PC acts as master, initiating all transactions. ACK (06H) and NAK (15H) characters are used for positive and negative (error) acknowledgement, respectively.

2.2.3 Microcontroller Hardware Description

The microcontroller unit, shown in Fig. 2.17, uses an Intel 80C31 CMOS microcontroller as the computational unit. The clock rate is set to 11.059 MHz to facilitate the generation of standard asynchronous baud rates using an internal timer. The CPU design is conventional with program and main data memory residing outside the microcontroller. The data memory, a 32K x 8 CMOS static RAM device, is mounted on a Dallas Semiconductor DS1213C "SmartSocket" battery backup unit. Microcontroller ports 0 and 2 provide the data bus interfaces.

Port 1 of the microcontroller is used for discrete I/O. Each discrete output is buffered using an invertor and open collector transistor. The invertor ensures that the transistor will be in the "off" state at startup. Bits 0-3 are used for input and 4-7 for output. Port 3 is used for asynchronous serial I/O, external memory control, and miscellaneous I/O.

The analog input section has two inputs. The first, used to measure the output of the power amplifier, is single ended and unconditioned. The second input is for use with a bridge type strain gage sensor and uses a high speed instrumentation amplifier. A 5 volt excitation supply is also included. A multiplexer allows software selection of the input source. A TL020 half-flash, 8 bit, A/D converter converts a 0-2.5 volt input in 1.2 microseconds. A separate overflow bit indicates the validity of the reading. An AD 7525 8 bit D/A converter provides a 0-2.5 volt output to control the driver amplifier.

Complete schematics and component lists for the microcontroller subsystem are provided in Appendix C of this report.

2.2.4 Onboard Programmable Pulse Power

The power amplifier has the following characteristics:

- Type: single ended, Class B
- Input Voltage: 0-5 V
- Output Voltage: 0-300 V
- Voltage Gain: 50
- Power Handling Capability, 2 Second Interval: 15 mJ.
- Voltage Switching Rate: 1 volt/microsecond
- Frequency Response: 0-100 kHz @ +/-1dB
- Overshoot: 5% Maximum with a 20 microsecond settling time in response to a 50% full scale input step.

Like the microcontroller, the power amplifier was constructed using printed circuit technology except for the power transistors which are mounted on heat sinks. The power pack consists of a 400 V 25 mA power supply which charges a 150 uF capacitor. The power pack is disconnected from the power supply once the voltage falls below 300 V. Since the available energy stored in the 100 V discharge range of the capacitor is 15 mJ, the power amplifier cannot be overloaded. A safety interlock completely discharges the capacitor when the main power is disconnected of the power amplifier cover is opened. It should be noted that the energy stored in such a system supplies sufficient voltage and current to cause a **lethal shock**. Anyone attempting to re-create or expand upon the research described in this report should thus take appropriate safety precautions.

The driver amplifier must operate under some fairly severe constraints:

- The load is an essentially pure 25 uF capacitance.
- The load must be drive between 0-300 V at a switching rate of 1 volt/microsecond
- Frequency response is to be 0-100 kHz with a maximum overshoot of 5%.

As a result, the amplifier must be able to handle extremely high peak power levels ... as much as 8 kW. Also, the use of feedback to maintain DC and AC stability must allow for the very low frequency "pole" introduced by the load.

To meet these requirements, the driver amplifier, shown in Fig. 2.18, is divided into two stages. The first uses conventional op-amp topology to provide both voltage and current gain. This stage is switching rate limited at 1 volt/microsecond. This controls the maximum current into the stack (25 A). The primary feedback loop is within this stage. The second stage is a class AB voltage follower, i.e. it has a voltage gain of one and a current gain of at least 30. This stage also provides isolation between the feedback loop and the load.

The power supply section is comprised of a conventional low voltage supply driving the microcontroller and front end section of the driver amplifier and a high voltage supply for the load. Since a 300 V, 25 A supply would be impractical, the high voltage supply is sized to directly provide the bias current requirements of the driver amplifier and the time to charge a capacitor bank to 350 V. These capacitors supply the stack charging current. Complete schematics for the pulse power supply are provided in Appendix C of this report.

2.2.5 HSILS Driver Software

The software, both onboard the microcontroller as well as the PC interface code, was written in "C" using the Franklin 8051 compiler. A modular design to the code was adopted to permit future modification and flexibility. The program presently contains the following components:

Initialization

The initialization module is responsible for ensuring that all outputs are in a known state and initializing the internal clock and serial port.

Serial I/O drivers

The Serial I/O drivers provide interrupt driven buffering for incoming characters. This is necessary to guarantee proper operation at 9600 bits/second. The communication parameters are: 9600 bits/second; 8 bits; no parity; and one stop bit. The only entry points to the Serial I/O drivers are **getchar** and **putchar.**

Command Dispatcher

The command dispatcher simply reads a command from the communications processor and calls the appropriate execution module. Adding a new command is accomplished by creating a new reference in the command table described below.

Command Execution Modules

The microcontroller software (See Appendix D) uses standard ASCII codes and can be used with any standard communications program that supports direct (non-modem) connections. The initial version of this program contains the following executable commands.

Reset

Parameters: None
Description: Returns system to power on state. Resets any outputs but does not clear the profile memory.

Load Profile

Parameters: None
Description: Loads the stack actuation profile. When the load profile command is received, the controller waits for the user to enter time and voltage

pairs as follows:

time,voltage

where the time is in microseconds from start and voltage is a stack voltage from 0 to 300 V. Up to 1024 points may be entered. Load profile is ended when the line:

End Profile

is received. The last point must contain a zero voltage value.

Start Run

Parameters: None
Description: Begins execution of downloaded profile. After the last point, the output is returned to 0 and a prompt is returned to the host.

Set Discrete Outputs

Parameters: New value to output (0-15)
Description: Writes the lower 4 bits of the parameter to the discrete outputs. A "1" places the output in the "low" state.

Get Discrete Status

Parameters: None
Description: Returns a number from 0-15 reflecting the current state of the discrete input lines. A 1 bit value indicates the input is low.

2.3 RCS Control Thruster

A "control" experiment (cold gas thruster) was constructed using a traditional solenoid valve in order to provide first hand data for comparison with the performance of the HSILS actuator. This device, shown in Fig. 2.19, consisted of two interconnected stainless steel pressure vessels (service rated at 35 MPa; STP volume of 150 cc each); a recharge port; a 6mm stainless steel delivery pipe connecting the pressure vessels to the control valve; a Honeywell/Skinner B2LX94N solenoid valve; and an expansion nozzle.

The solenoid valve had the following specifications:

- Operating Voltage: 24vdc
- Operating Pressure 2.76 - 6.9 MPa
- Orifice size: 0.794 mm;
- Coil: 2440 +/-1% turns; #33 awg wire

86.6 +/-8% ohms at 25C
- Return spring: Part No. B04005; k = 1.384 n/mm
  force = 2 n preload; 3.3 n in full open state.
- Core Spec: Carpenter Steel #430FR
- Valve Seat Material: Teflon

Operation of this valve was accomplished using the same programmable pulse power unit developed for HSILS with the notable exception that the peak voltage was limited to 40 vdc to prevent failure of the 24vdc rated solenoid coil. All other aspects of the experimental program, including sensors, data collection and data reduction procedures were identical to those used for the HSILS tests.

2.4 Analytical Modeling

2.4.1 Force, Duration, and Gas Storage Requirements

In order to proceed from the conceptual HSILS unit shown in Figs. 2.7 through 2.13 to the construction of a practical experimental prototype a number of subsystem elements had to be sized and designed. This proved to be an iterative process since each element in the electro-mechanical linkage affected the design of all the other components (e.g. selection of a nozzle throat diameter affected the sizing of the return spring, which in turn affected the dynamic behavior of the unit). To remedy this a set of performance criteria were identified and an interactive computer graphics program, THRUSTER (see Appendix A for program listing), was developed which permitted parametric analysis of the system. The performance criteria selected for the bench test HSILS unit were as follows:

- Capable of generating inverse gaussian impulse.
- Peak impulse thrust = 44 N.
- Maximum impulse duration = 0.020 s.
- Onboard power and gas storage to meet, but not significantly exceed the above specifications in an effort to reduce mass to less than 5 kg for the entire HSILS unit.

The specific elements of the parametric analysis (under user control) were:

- Initial gas storage pressure (MPa)
- Pressure vessel inside diameter (i.e. STP volume, cc)
- Room temperature (oC)
- Nozzle throat diameter (mm)
- Displacement amplifier magnification factor
- Integration time step (seconds)
- Return spring stiffness (N/mm)
- Inverse gaussian pulse shape parameter, σ

Given these input data the ideal inverse gaussian impulse force as a function of time can be calculated as:

$$F_{theo} = \frac{\left(F_{imp}\sqrt{4\pi\left(\frac{\sigma^2}{6}\right)^3}\right)\sigma}{\sqrt{4\pi t^3}}\left(e^{\frac{-\sigma^2}{4t}}\right) \tag{1}$$

where:

F_{theo} = theoretical impulse force at time t (newtons)
F_{imp} = specified peak impulse force (newtons)
t = elapsed time (seconds)
σ = shape constant for inverse gaussian pulse

For the particular case of a cold gas thruster nozzle flow can be modeled by one-dimensional isentropic expansion relations. Sutton (**Sutton, 1986**) provides the following relationship for thrust:

$$F = P_c A_t C_f \tag{2}$$

where:

F = Actual thrust (newtons)
P_c = Chamber pressure (Pa)
A_t = Nozzle throat area (m^2)
C_f = Thrust coefficient, defined as:

$$C_f = \sqrt{\frac{2\gamma^2}{\gamma - 1}\left(\frac{2}{\gamma + 1}\right)^{\frac{(\gamma + 1)}{(\gamma - 1)}}\left[1 - \left(\frac{P_e}{P_c}\right)^{\left[\frac{\gamma - 1}{\gamma}\right]}\right]} + \frac{P_e - P_a}{P_c}\left(\frac{A_e}{A_t}\right) \tag{3}$$

where:

γ = Specific heat ratio of the compressed gas
P_e = Nozzle exit pressure (Pa)
P_a = Ambient pressure (Pa)
A_e = Nozzle exit area (m^2)

For laboratory tests at STP $P_e = P_a = 1$ bar; in space both terms would be effectively zero. Useful constants for various gases that might be used in a cold gas thruster are as follows:

Table 2.1: Gas Constants

Gas	Specific Heat Ratio, γ	Molecular Weight	C_f $[P_a=1$ bar]	C_f $[P_a=0]$	I_{sp} (seconds) $[P_a=1$ bar]	I_{sp} (seconds) $[P_a=0]$
Hydrogen (H_2)	1.410	2.015	1.626	1.800	267	296
Nitrogen (N_2)	1.404	28.016	1.629	1.807	72	80
Helium (He)	1.659	4.003	1.544	1.627	170	179

As the valve core lifts away from the nozzle throat, as shown in Fig. 2.20, pressurized gas escapes producing thrust as defined by eq (2). The effective throat area is given by:

$$A_t = 2\pi \left(\frac{d_{th}}{2} - \frac{\delta_c}{4} \cos\theta \right) \delta_c \sin\theta \tag{4}$$

where:

γ = Valve core displacement (mm)
d_{th} = Nozzle throat diameter (mm)
θ = Valve core half cone angle (degrees)

For a 60-degree valve core cone angle, typical of high pressure valve seats, the effective throat area reduces to:

$$A_t = \frac{\pi\, d_{th}\, \delta_c}{2} \tag{5}$$

For a low pressure elastomeric pad seat (180-degree cone angle = flat interface) the equation for effective throat area is given by:

$$A_t = \pi \, d_{th} \, \delta_c \tag{6}$$

In any event, the maximum effective throat area is controlled by the nozzle throat area given by:

$$A_t = \frac{\pi \, d_{th}^2}{4} \tag{7}$$

We may now substitute the expression for effective throat area into eq (2) to determine the required core displacement at time t (seconds) as:

$$\delta_c = \frac{F}{P_c \, \pi \, d_{th} \, C_f} \tag{8}$$

Since there is a limiting state for δ_c the real time value for F (actual thrust) must be recalculated as:

$$F = P_c \, \pi \, d_{th} \, C_f \, \delta_c \tag{9}$$

As the analysis progresses through time it is possible to determine the maximum core displacement. The NIST piezoelectric stack was proven to be capable of achieving a maximum axial displacement of 0.0686 mm at 225 vdc drive voltage. Knowing this mechanical limitation, the required displacement amplification ratio is given by:

$$R_a = \frac{\delta_{c\text{-max}}}{0.0686} \tag{10}$$

If the required amplification factor is greater than that specified program THRUSTER advises the user that the amplification factor must be adjusted and the analysis to this point is rerun. The required piezoelectric stack displacement at any given time is then defined as:

$$\delta_{stak} = \frac{\delta_c}{R_a} \tag{11}$$

Given a spring preload force, amplification factor, and required stack displacement, the required drive voltage from the pulse power system can be calculated using the block force piezoelectric equations from Fig. 2.5 as:

$$V_{drive} = \left(\frac{F_{preload} R_a - 65669 \frac{\delta_c}{R_a}}{159.86} \right)^{1.517} \qquad (12)$$

This completes one differential time step in the parametric HSILS analysis. At this point a finite amount of gas molecules have been expelled through the nozzle at a **weight** flow rate (expressed in units of newtons/second) given by Sutton (**Sutton, 1986**) as:

$$\dot{w} = \frac{F_{thrust}}{I_{sp}} \qquad (13)$$

Given that the initial weight of gas contained in the pressure vessel can be calculated from the ideal gas equation as:

$$w_0 = \frac{P_0 \, V_{tank} \, M}{R_{const} \, T_{abs}} \qquad (14)$$

where:

V_{tank} = Pressure vessel volume, given by $(4/3)\pi(D/2)^3$
M = Molecular weight of the propulsive gas
R_{con} = Universal gas constant
T_{abs} = Absolute temperature

it is then possible to determine the internal pressure, P_c, at any time t as:

$$P_c = P_0 - \frac{1}{w_0} \int \dot{w} dt \qquad (15)$$

where:

P_0 = initial internal tank pressure (MPa) and the bounds on the integral are from zero to time t. This then provides the initial values for the next numerical time step calculations. For small time steps (on the order of 0.0001 second) it is possible to accurately determine both the instantaneous velocity and acceleration of the valve core. This permits a determination of the required return spring force needed to insure that, as the piezoelectric stack contracts as voltage is reduced in a voltage-time history, there is no "lift off" at any of the spherical sapphire bearings which form the hinge contact points in the mechanism.

The above equations do not model the transient dynamic effects associated with the various stiffnesses, masses, and damping properties of the HSILS components. Rather it is assumed for the purposes of initial component sizing that motion transmission is instantaneous and that all components are perfectly rigid and massless (except for the calculation of return spring stiffness). Transient structural response is discussed in Section 2.4.3 below.

2.4.2 Typical Limiting Design Conditions

Figs. 2.21 and 2.22 show the effect of the nozzle throat diameter on the ability of the HSILS to track an inverse gaussian pulse whose characteristics are given in the inset boxes on these figures. In viewing these figures the reader should keep in mind that as time increases from initiation of the pulse the chamber pressure, P_c, in the onboard gas storage vessel is decreasing in a non-linear fashion. The HSILS unit compensates for this by increasing the valve core displacement so as to achieve the desired thrust (force) level at any given point along the pre-defined pulse (which in the actual unit is downloaded from memory to the digital to analog converter and then amplified via the pulse power unit). Peak thrust is determined, as defined above in eq (2), by the internal chamber pressure, the thrust coefficient (which is a function of the gas used, and the nozzle throat area, which is effectively defined by the throat diameter, d_{th}.

In Fig. 2.21 it can be seen that for nozzle diameters less than approximately 1 mm the unit cannot achieve the desired peak thrust, as evidenced by a "flat top" profile. Actually, the maximum thrust plateaus in this case are sloped with gradually decreasing maximum thrust level with increasing time. This is a result of decreasing chamber pressure with time in the face of a full-open valve. An alternative means of viewing the effects of nozzle throat diameter is shown in Fig. 2.22 which plots the HSILS valve core displacement with time for the same problem shown in Fig. 2.21. Here it can be seen that beyond a throat diameter of 1 mm, the peak valve core displacement needed to achieve the desired peak thrust is decreased with increasing throat diameter. Conversely, for throat diameters less than 1 mm, the maximum nozzle throat area is achieved at lower valve core displacements, as evidenced by the curves with truncated tops. For design purposes the throat diameter should be selected to insure that the peak desired thrust is capable of being attained without truncation and that the required valve core displacement to achieve this can be delivered by the piezoelectric stack displacement amplifier. For this example, the amplification ratio was set at 12.

Figures 2.23 and 2.24 show the effects of the volume of the onboard gas storage vessel (as reflected by the inside diameter, D_{tank}, of a spherical tank) on the thrust tracking ability of the HSILS unit. The philosophy here is to design the minimum size onboard gas storage system so as to achieve a desired maximum pulse (or pulse series) capability between periods of recharge. Although the HSILS unit is intended for remote operation, it will require a relatively small umbilical to supply recharge power to the pulse power unit and to provide high pressure gas to the local "accumulator" storage vessel(s).

With each firing of the unit thrust is created by the expulsion of a finite number of gas molecules stored in the accumulator tank. If the volume of this tank is insufficient the HSILS unit will be unable to track the desired impulse which is downloaded from the onboard microprocessor. Fig. 2.23 shows the gradual degradation of the ability of the HSILS unit to track the desired pulse (shown for D_{tank}= 89 mm) with decreasing storage vessel diameter. During this time it can be seen in Fig. 2.24 that the onboard microcontroller will try and compensate for reduced internal pressure by increasing the valve core displacement. For tank diameters of less than 89 mm in this example the processor eventually commands a full-open condition (given by a valve core displacement of 0.3 mm in this case), after which no further compensation can be made and the tracking ability is lost, as shown in Fig. 2.23.

On the basis of analyses using THRUSTER the following design parameters were set for the laboratory demonstration HSILS unit:

- Onboard gas storage tank volume (@STP) = 0.6 liters, which corresponds to D_{tank} = 105 mm. In the actual unit this was provided by four 150 cc high pressure stainless sample cylinders.
- Service pressure = 34.5 MPa
- Nozzle throat diameter, d_{th} = 1.194 mm
- Amplification ratio, R_a = 12
- Return spring stiffness = 11.74 N/mm

Further specific mechanical details on the HSILS prototype are presented in Appendix E.

2.4.3 Dynamic Simulation of HSILS

The simulations described above, while being useful for initial design purposes, do not include the effects of transient structural response on the internal mechanism. This internal dynamic response ultimately will manifest itself at the valve core tip and subsequently will have an effect on thrust. In addition to assessing the impact of these vibrations a dynamic structural model is also useful for later identification of actuator characteristics. Of these characteristics, damping ratio is the most difficult to quantify a priori.

In this section a one-degree-of-freedom equivalent structural model for the HSILS is presented. This model was implemented in the interactive computer graphics program HDYN described in Appendix B. HDYN permits real time display of analytically predicted and experimentally observed impulse response of the HSILS and allows for interactive manipulation of the model parameters, including response amplitude, time shift, mass, stiffness, and damping.

Fig. 2.25 shows a simplified model of the internal mechanism of the HSILS (see Figs. 2.7 and 2.10 for reference to the actual hardware). The displacement amplifier (shown previously in Fig. 2.3) was determined (via finite element analyses) to have the dynamic characteristics presented in Table 2.2 for two different cross-sectional widths. An amplifier width of 10 mm was selected for the final design.

Table 2.2: Displacement Amplifier Modes and Frequencies

Mode Number	Frequency (hertz) thickness = 5 mm	Frequency (hertz) thickness = 10 mm
1	598	598
2	1160	1855
3	5220	7673
4	7670	9215
5	19085	19085
6	20493	29379
7	33789	37303
8	34185	52565
9	51828	72016
10	54843	73428

In the above data the first mode of vibration is the desired rocking motion about the milled hinge z-axis; modes 2 through 4 are sway and torsional rotations about the x and y axes. Mode 5 is the first true flexural response of the amplifier. The anvil shape of the amplifier was specifically selected to achieve high bending stiffness (e.g. 598 Hz in rocking versus 19085 Hz in flexure). For this reason the amplifier may be assumed for analytical modeling purposes to be essentially rigid (but **not** massless). Energy losses (damping) as a result of flexural strains at the milled hinge can be shown to be approximately 1% for the design described in Appendix E and are therefore omitted from the model. Calculated properties for the displacement amplifier are as follows:

t_{anvil} = Cross section thickness of amplifier = 10 mm
A_{anvil} = Cross Sectional Area = 780 mm^2
I_x = 86073 mm^2
I_y = 738519 mm^2

I_{xy} = 196162 mm^2
I_0 = $I_z = \rho(I_x+I_y)t_{anvil}$ = 37189 g-mm^2 (0.00033425 lb-sec^2)

The anvil was fabricated from 6AL-4V (6% aluminum; 4% vanadium; balance titanium) titanium alloy with a mass density of 4.51 g/cc; yield strength (0.2% offset) 828 MPa; and tensile strength 897 MPa. The following values apply to the anvil mass, M_2; the valve core mass, M_1; the return spring stiffness, K_1; and anvil lever arms a,b, and c.

M_1 = 3.8g (0.00837 lb)
M_2 = 35.8g (0.078854 lb)
K_1 = 11.738 N/mm (67 lb/inch)
a = long lever side = 61 mm (2.4 inch)
b = hinge to center of mass of anvil = 23.243 mm (0.91508 inch)
c = short lever side = 5.08 mm (0.2 inch)

From Fig. 2.25 equations can be written for the resisting forces to the piezoelectric stack driving force, P(t). These resisting forces are given as:

$$\begin{aligned}
F_{K1} &= K_1\left(\frac{a}{c}\right)y(t)\\
F_{M1} &= M_1\left(\frac{a}{c}\right)\ddot{y}(t)\\
F_{C1} &= C_1\left(\frac{a}{c}\right)\dot{y}(t)\\
F_{M2} &= M_2\left(\frac{b}{c}\right)\ddot{y}(t)\\
M_I &= I_0\ddot{\theta}\cong I_0\left(\frac{\ddot{y}}{c}\right)\\
P(t) &= piezoelectricforcingfunction
\end{aligned} \qquad (16)$$

Following the virtual displacements shown in Fig. 2.25 the virtual work equation can be written as:

$$\delta W = -F_{KI}\left(\frac{a}{c}\right)\delta y - F_{MI}\left(\frac{a}{c}\right)\delta y - F_{CI}\left(\frac{a}{c}\right)\delta y - F_{M2}\left(\frac{b}{c}\right)\delta y - M_I\left(\frac{\delta y}{c}\right) + P(t)\delta y \qquad (17)$$

Since the virtual work must be zero, Eq. 17 can be solved and re-arranged in generalized coordinates for the equivalent single degree-of-freedom problem as follows:

$$M^* \ddot{y} + C^* \dot{y} + K^* y = P(t)$$

where:

$$M^* = M_1 \left(\frac{a}{c}\right)^2 + M_2 \left(\frac{b}{c}\right)^2 + I_0 \left(\frac{1}{c}\right)^2 \tag{18}$$

$$C^* = C_1 \left(\frac{a}{c}\right)^2$$

$$K^* = K_1 \left(\frac{a}{c}\right)^2$$

The forcing function, P(t) in Eq. 18 is simply the piezoelectric stack equilibrium equation in the face of an applied voltage and existing instantaneous displacement, given as:

$$P(t) = -37500\ y(t) + 35.94\ E(t)^{0.6597} \tag{19}$$

where:

$$E(t) = InstantaneousDrivePulseVoltage$$

The instantaneous valve core displacement (which will be directly proportional to output thrust provided the gas accumulator pressure remains constant) is then given by:

$$Y^* = \left(\frac{a}{c}\right) y(t) \tag{20}$$

This value can then be substituted into Eq. 9 to obtain an estimate of the actual HSILS thrust. For short pulse widths (as typically achieved using the HSILS prototype unit during laboratory tests) the assumption of nearly constant accumulator pressure is valid.

Sample analysis runs using HDYN are shown in Figs. 2.26 through 2.30 to illustrate the type of behavior that were anticipated of the device. The predicted thrust response shown in these figures is based on the material and geometric properties described above for the prototype laboratory demonstration unit. Fig. 2.28 shows the response of the HSILS unit to a 0.001 s x 200 V half square wave pulse for various values of the damping coefficient C^* as a percentage of critical damping, defined by:

Laboratory tests on the prototype HSILS unit showed that at a core rod velocity of approximately 0.0254 meters/second a viscous damping force of 2.94 N was developed. This leads to an experimental damping coefficient for the system of 115.9 N-s/m. This value

$$C^*_{critical} = \frac{2M^*\omega}{\left(\frac{c}{a}\right)^2} = \frac{2\sqrt{M^*K^*}}{\left(\frac{c}{a}\right)^2} = 30\ N\text{-}s/m$$

where: (21)

ω = *naturalfrequency (radians/s)*

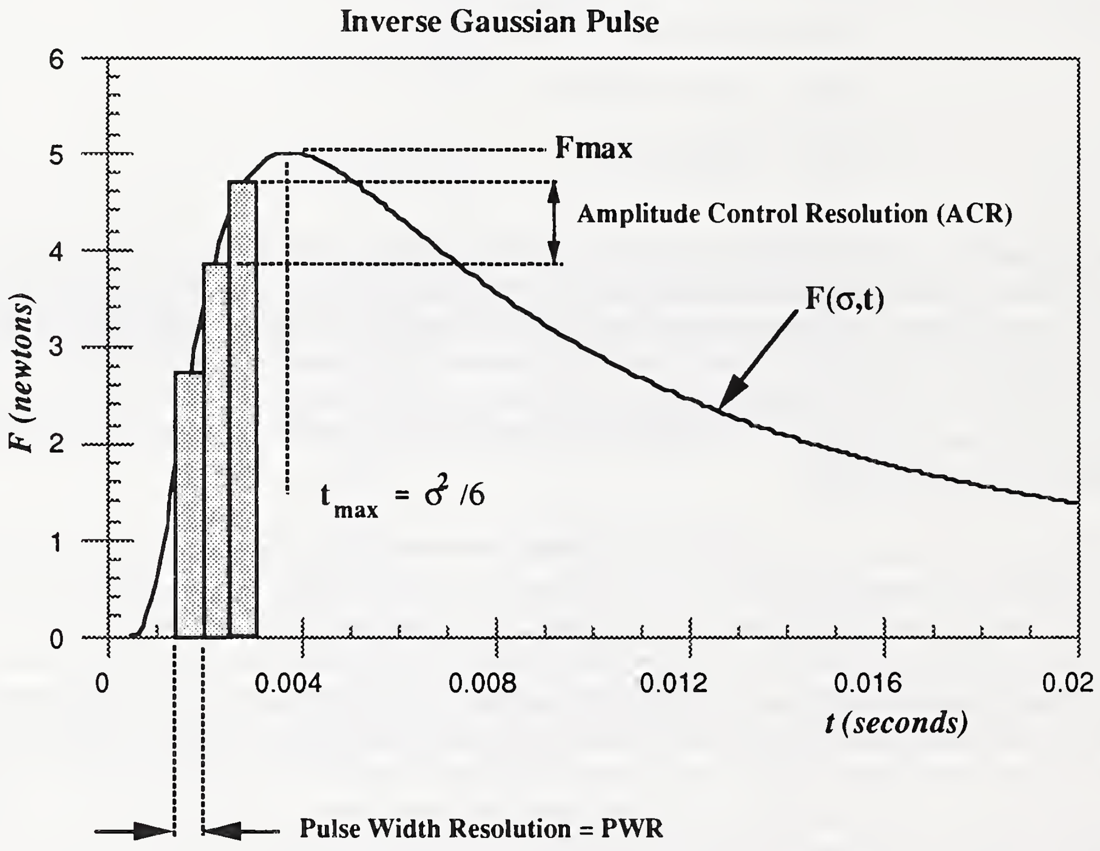
M^* = 2.759 $N\text{-}s^2/meter$

K^* = 1690296 N/m

indicates a heavily overdamped system with a damping ratio of 115.9/30 = 3.88, or 388% of critical. Viscous damping in the HSILS prototype was developed via friction between the valve core rod and the o-ring seals (both primary and backup rings) and guide slots through which it passes (see Fig. 2.11) to reach the high pressure gas chamber where it subsequently serves as the seal to the expansion nozzle. Since the experimentally measured damping coefficient was only able to be determined approximately, a sensitivity study was performed using HDYN (Appendix B).

Figs. 2.26 and 2.27 show the predicted response of the HSILS prototype to various drive voltage pulses of increasing duration (but identical peak voltage) with damping ratios of 388% and 50% of critical, respectively. Figure 2.28 shows the effect of damping, for a 200 V amplitude drive pulse with a nominal pulse-width of 0.001s, for a larger range of damping ratios. These figures indicate that for the highly overdamped case the peak response amplitudes are smaller than for the 50% damping case and the response tends to better reproduce the drive pulse shape. For shorter drive pulse widths the effect of overdamping can be seen as a rounding of the edges in the response pulse. In general, the "ringing" associated with the rising and falling edges of the pulse (as exhibited in Figs. 2.27 and 2.28) is effectively eliminated for the overdamped system. It is particularly important that this type of behavior, especially on the falling edge, be exhibited by the HSILS unit. The reason is that underdamped behavior would lead to multiple impacts (and subsequent rebound impulses) at the expansion nozzle throat as the valve core rod overshoots the full closure displacement. Aside from detracting from the ability to accurately track a voltage-time history, this also leads to impact induced fatigue, and eventual spalling, of the polished nozzle seat metal and subsequent leakage of high pressure gas.

Figures 2.29 and 2.30 show predicted HSILS response to a series of 0.001s drive pulses at varying drive voltage amplitudes. These show good tracking of the drive pulse for the overdamped system and deteriorated performance at reduced damping ratios. The frequency of oscillation for the underdamped cases is approximately 124 Hz, as calculated from the generalized mass and stiffness for the system defined in Eq. 18.



General Equation for Inverse Gaussian Pulse is Given by::

$$F(\sigma,t) = \frac{K\sigma \; e^{(-\sigma^2/4t)}}{\sqrt{4\pi t^3}} \qquad \text{where:} \quad K = \frac{\text{Fmax}\sqrt{4\pi\,(\sigma^2/6)^3}}{0.223\,\sigma}$$

In this example: Fmax = 5 N ; tmax = 3.75 ms ; σ = 0.15

Fig. 2.1: Inverse gaussian pulse and associated variable definitions.

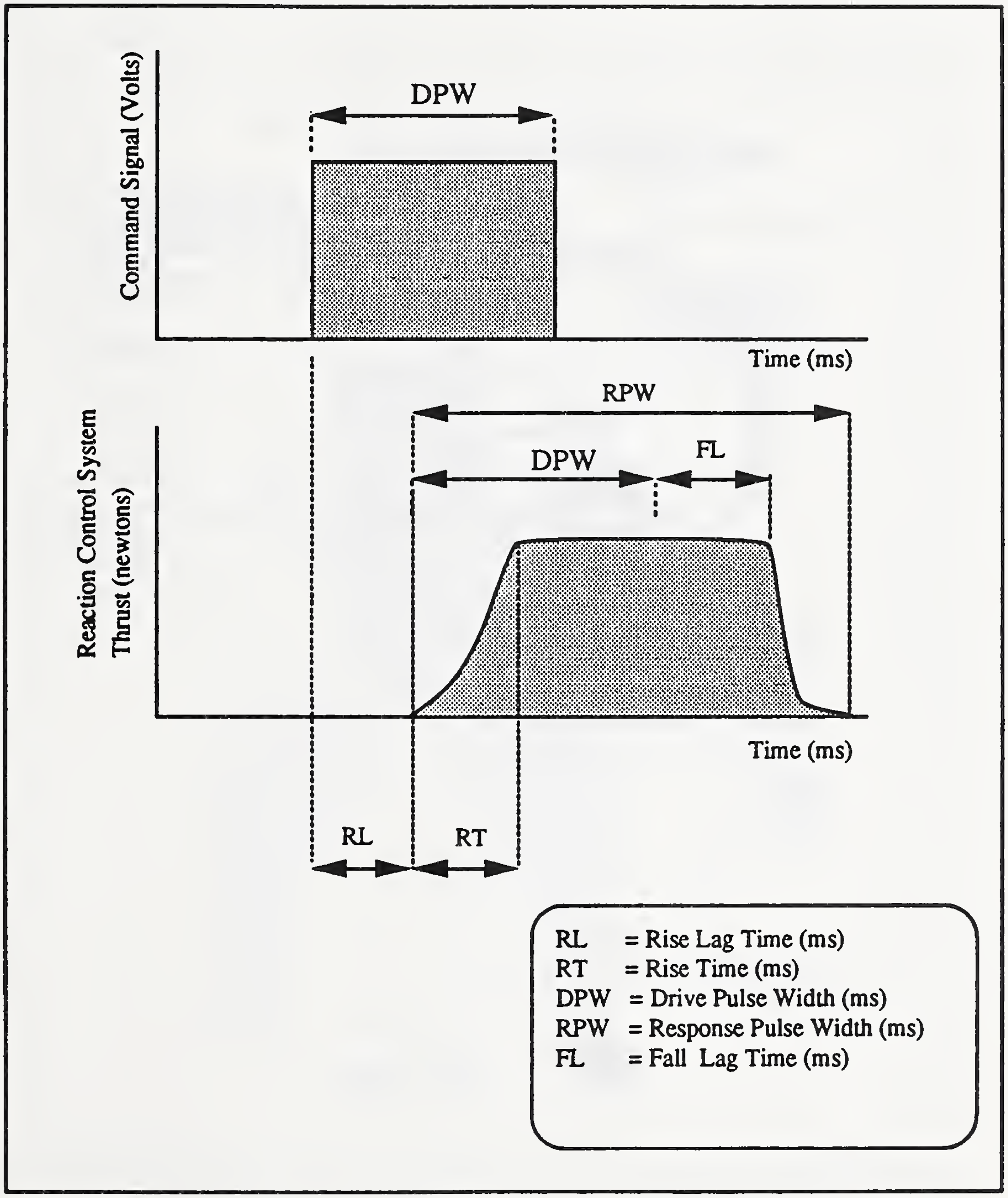


Fig. 2.2: Command and response transient performance definitions

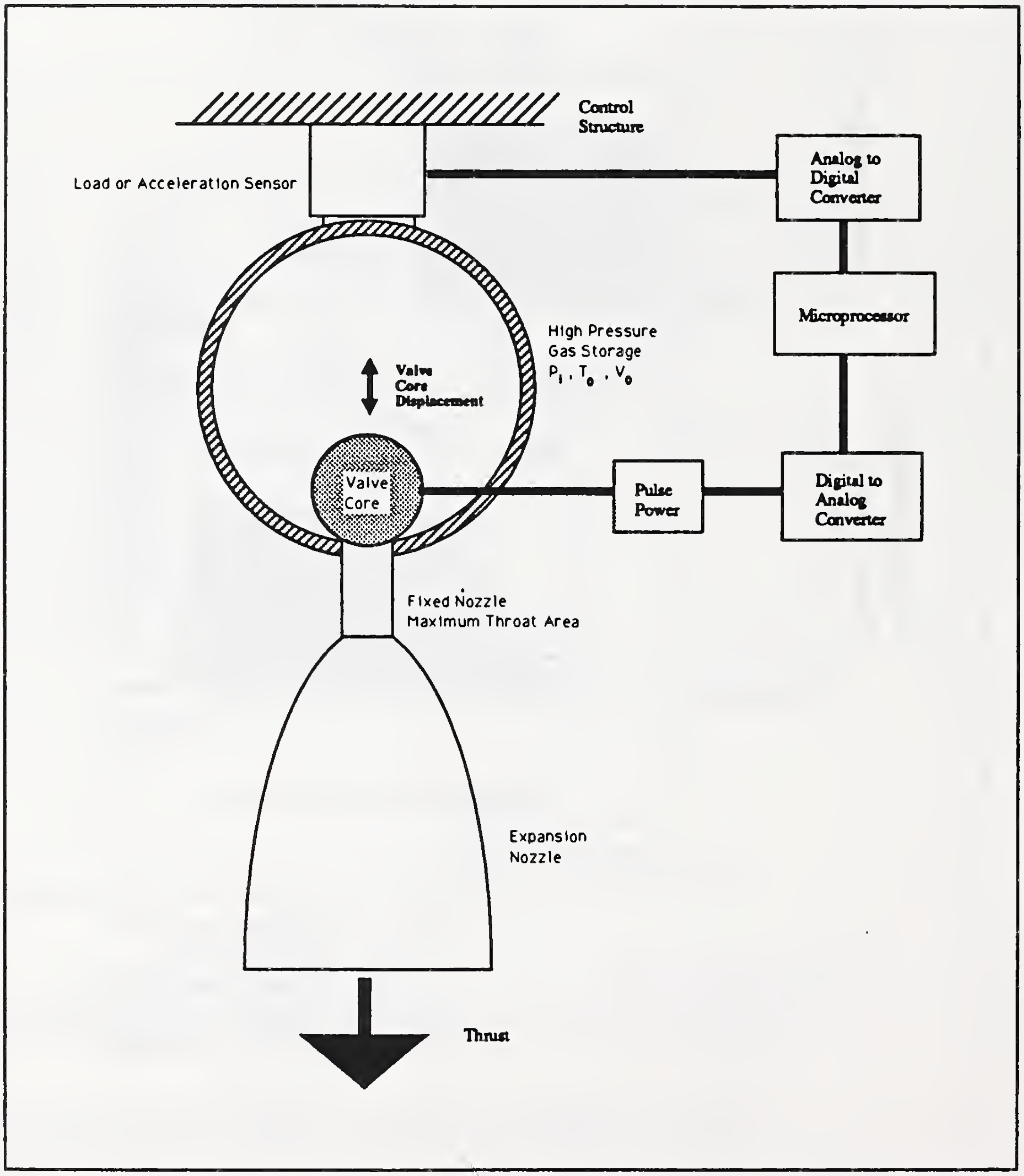


Fig. 2.3: HSILS variable amplitude control concept.

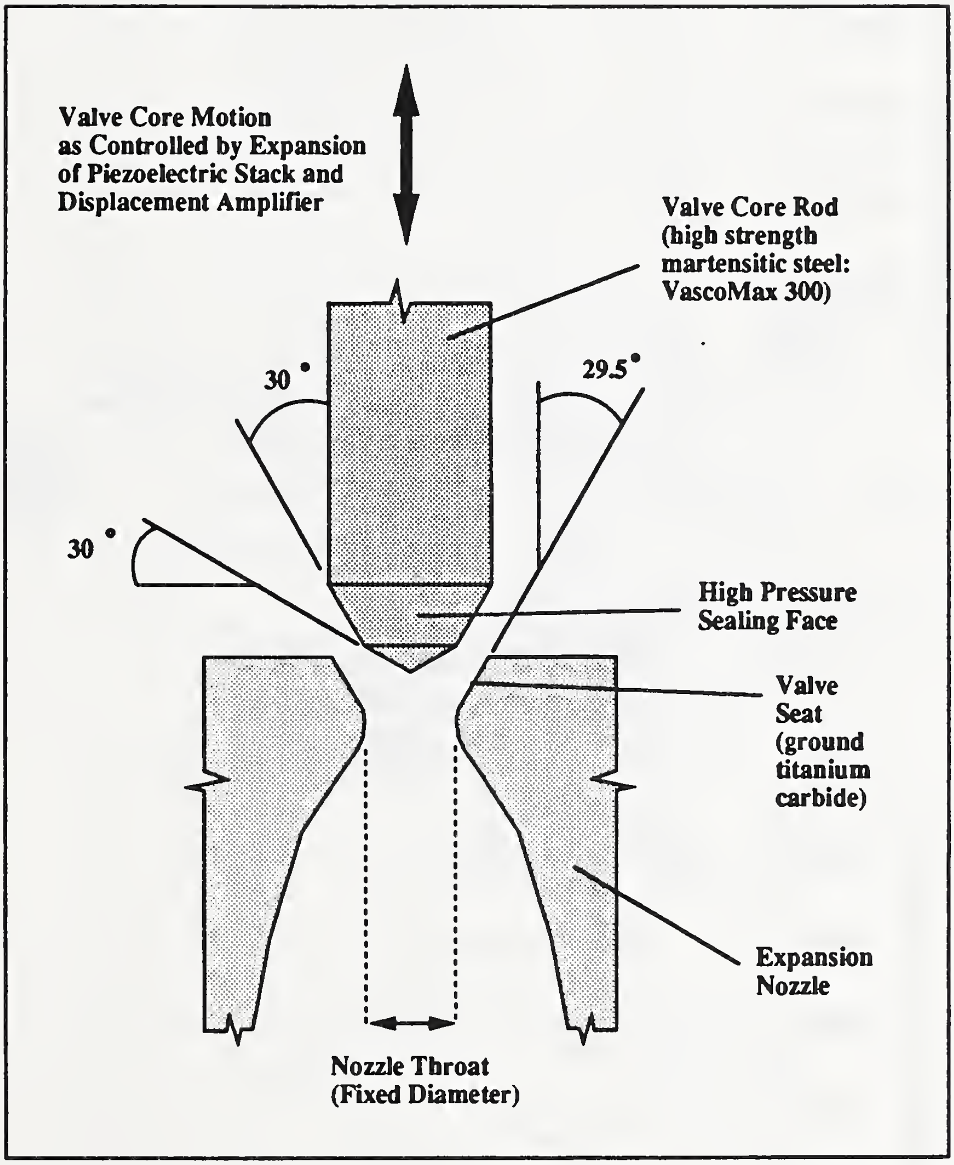
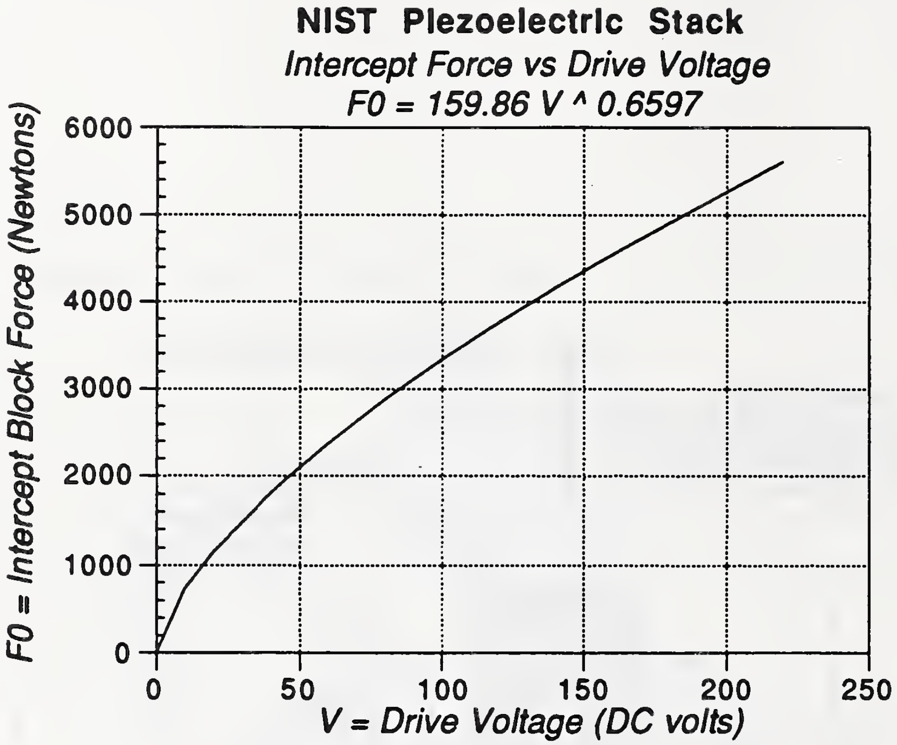


Fig. 2.4: High pressure variable mass flow rate valve seat.



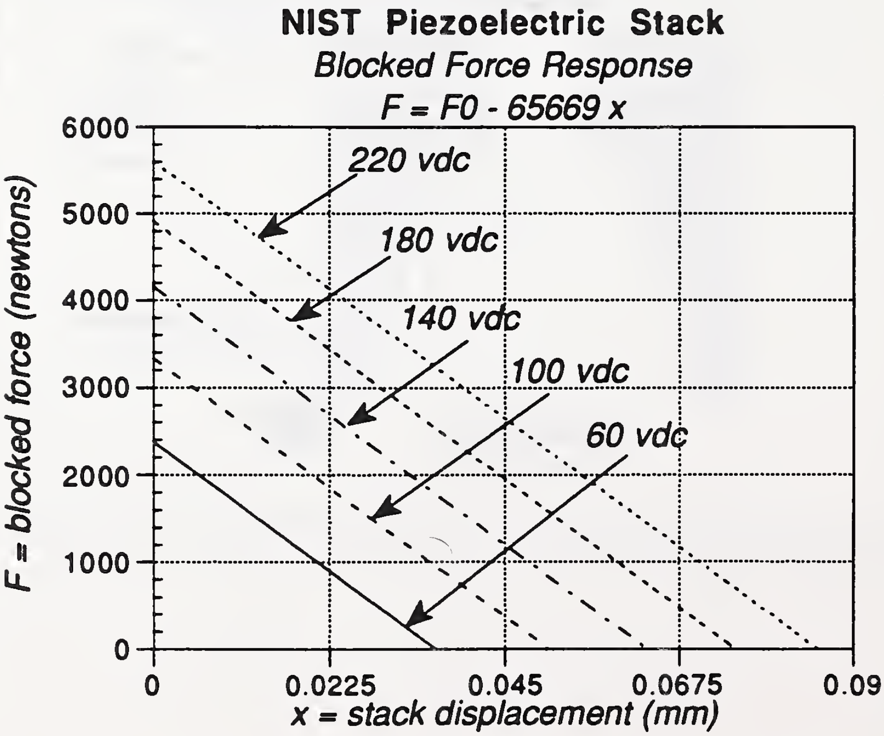


Fig. 2.5: NIST piezoelectric custom stack characteristics.

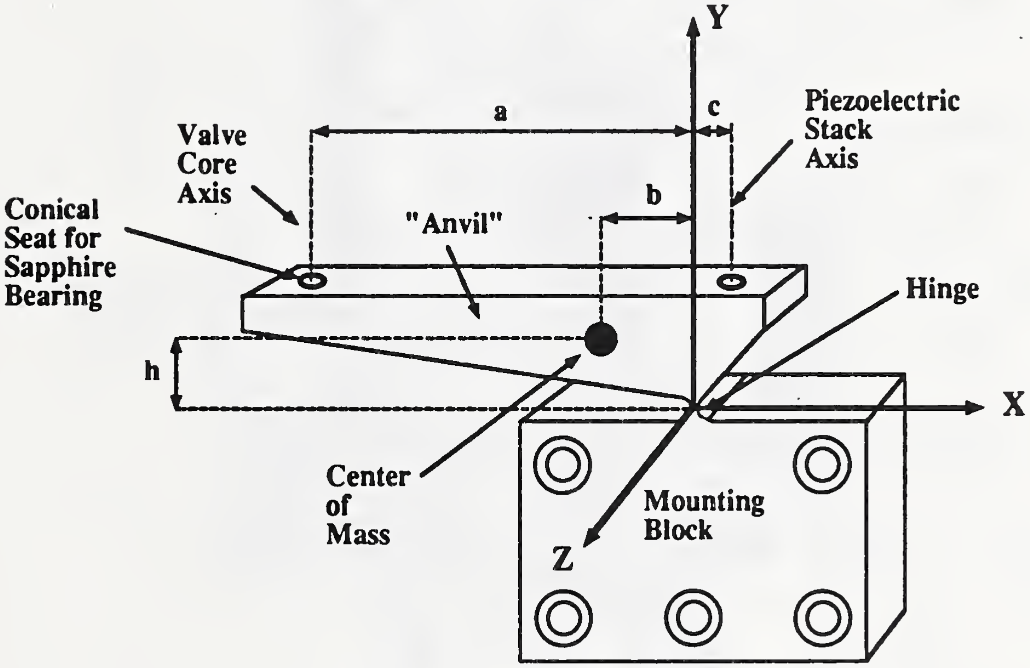


Fig. 2.6: HSILS Mechanical displacement amplifier.

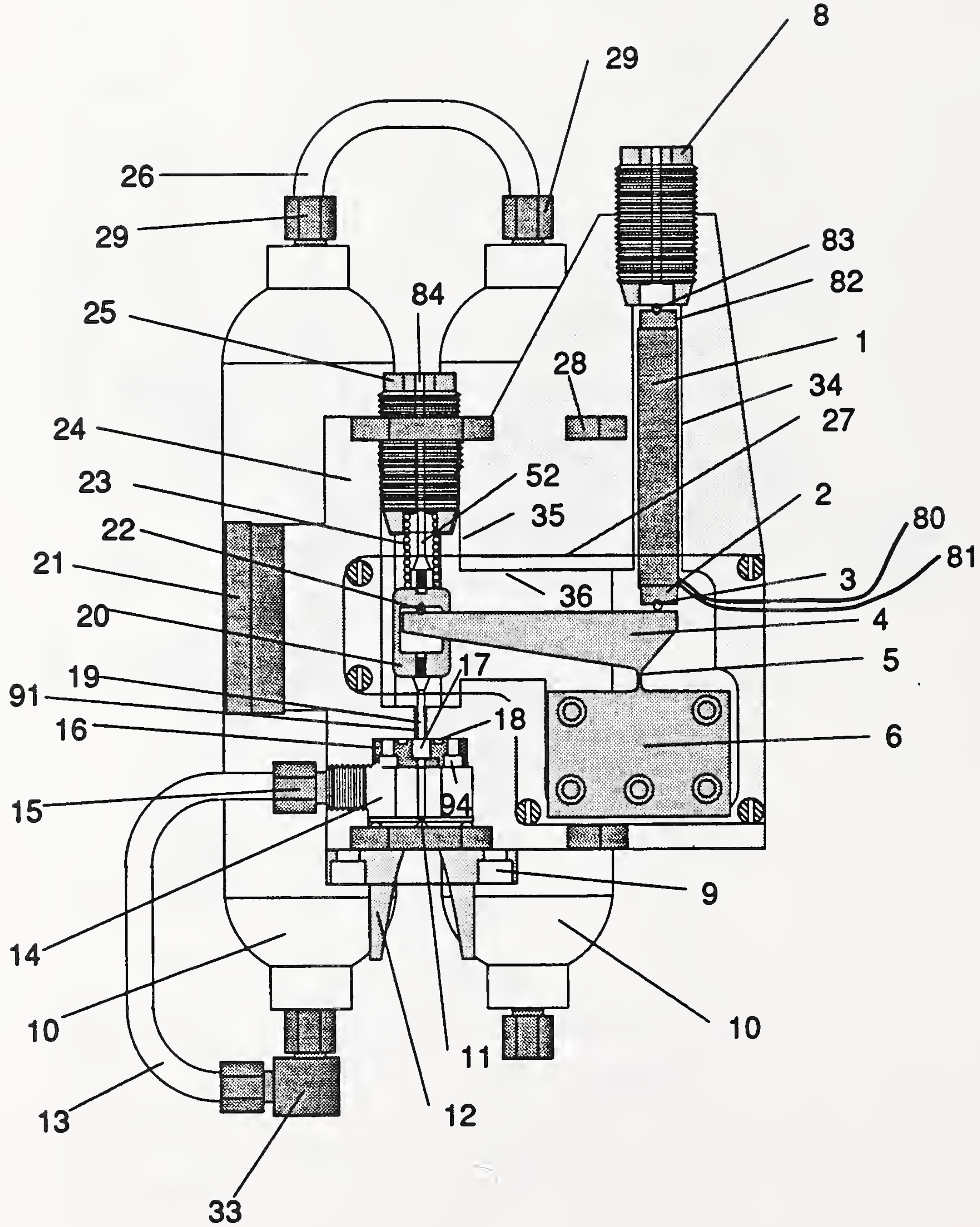


Fig. 2.7: Side view of HSILS showing all internal moving components (descriptions of numbered features are provided in the text).

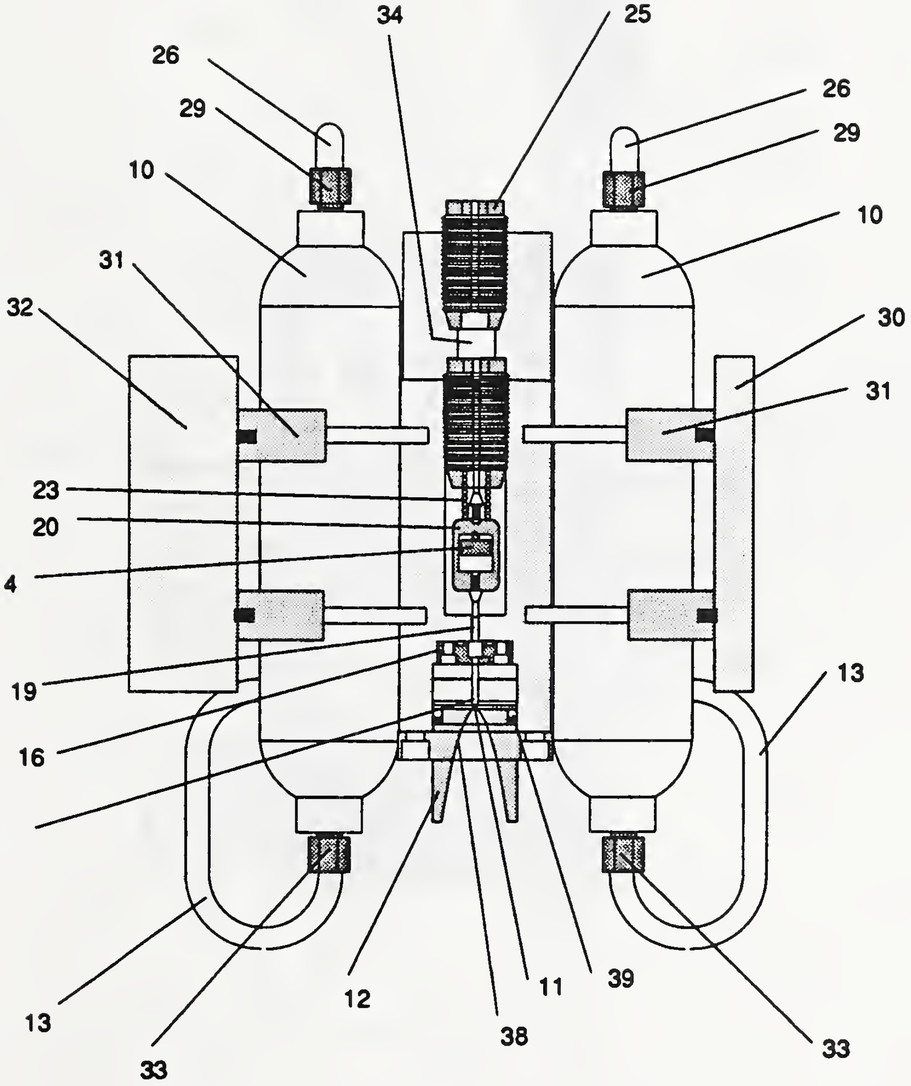


Fig. 2.8: End view of HSILS showing internal moving components (descriptions of numbered features are provided in the text).

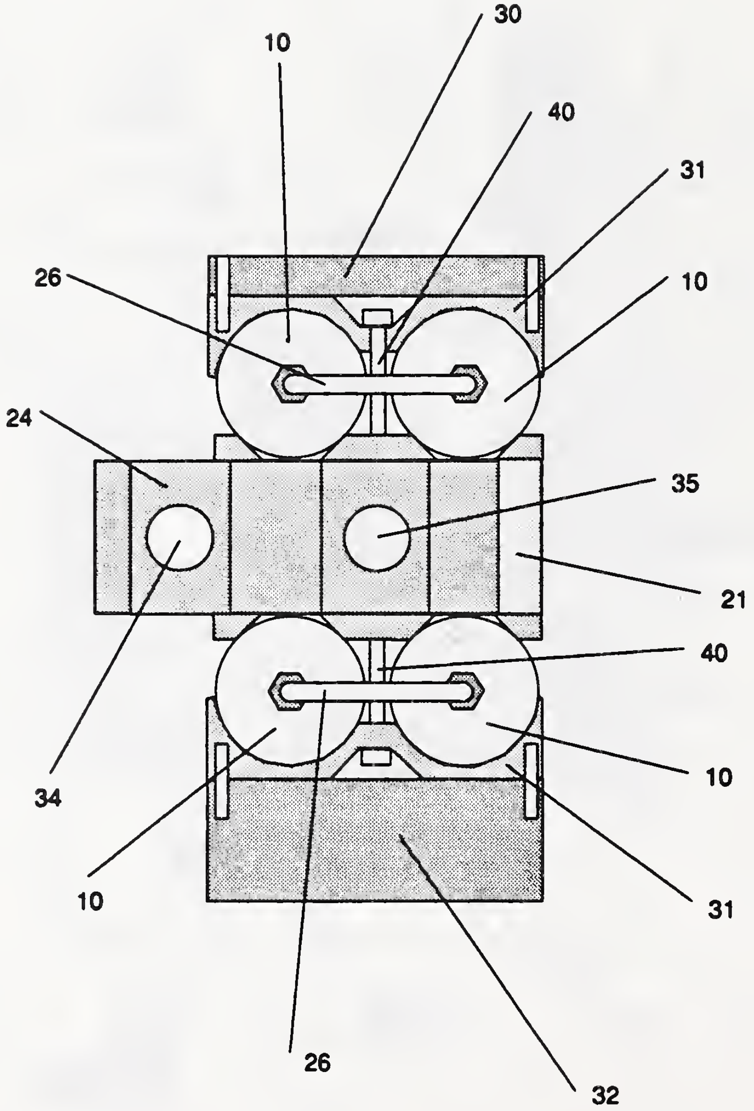


Fig. 2.9: Top view of HSILS showing gas accumulators, onboard pulse power and microcontroller subsystems and primary axes of piezoelectric stack and valve core rod (descriptions of numbered features are provided in the text).

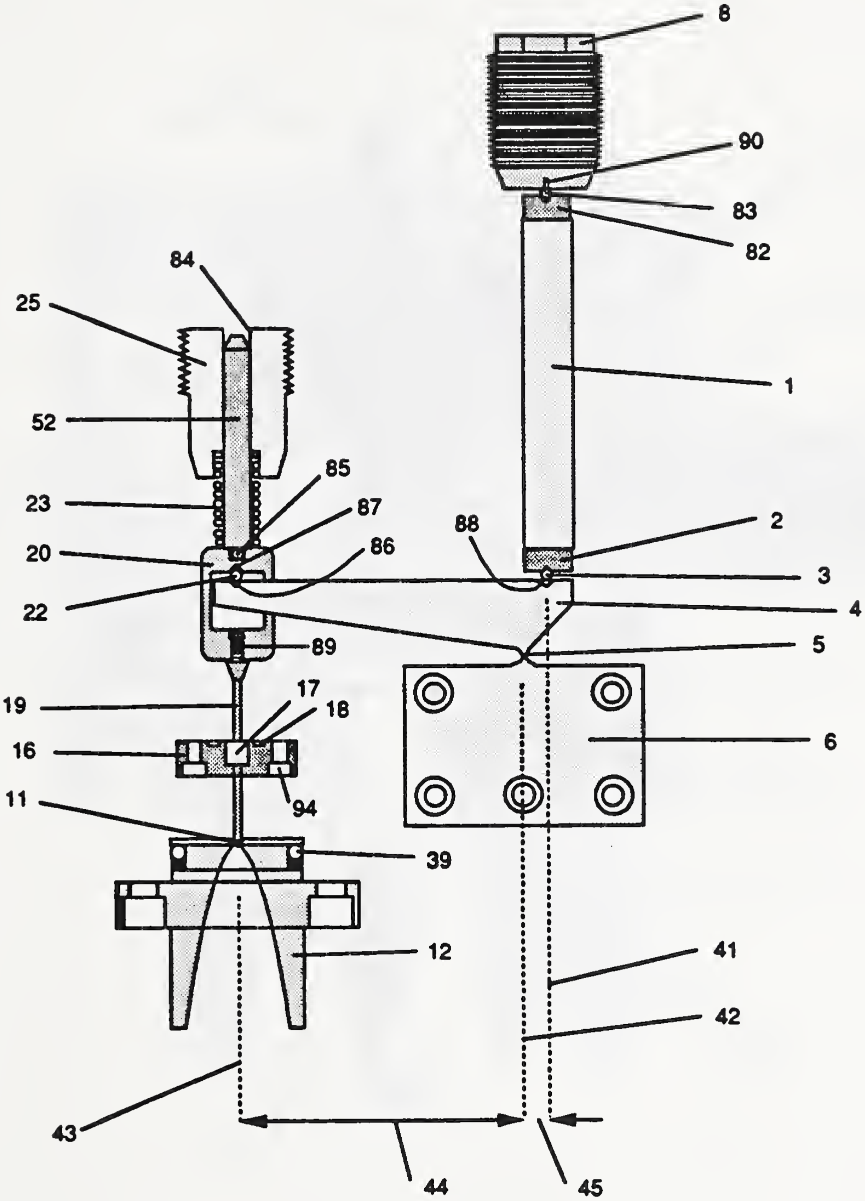


Fig. 2.10: Side view of HSILS showing all internal moving components, extracted from the housing and peripheral subsystems (descriptions of numbered features are provided in the text).

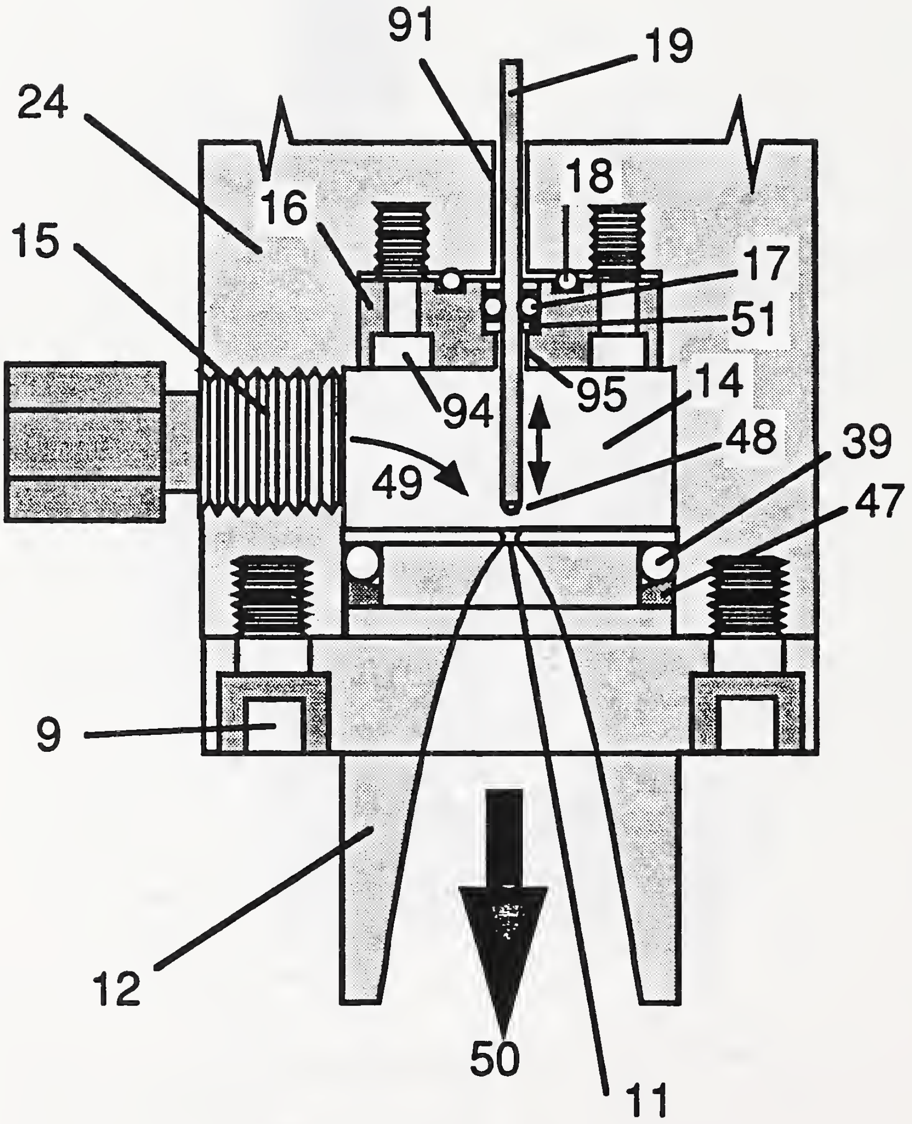


Fig. 2.11: Detail of the high pressure valve seat cavity of the HSILS unit showing valve core rod and nozzle throat seat; high pressure seals and backup rings for the valve core rod; and high volume high pressure feed line from the gas accumulators (detailed descriptions of numbered features are provided in the text).

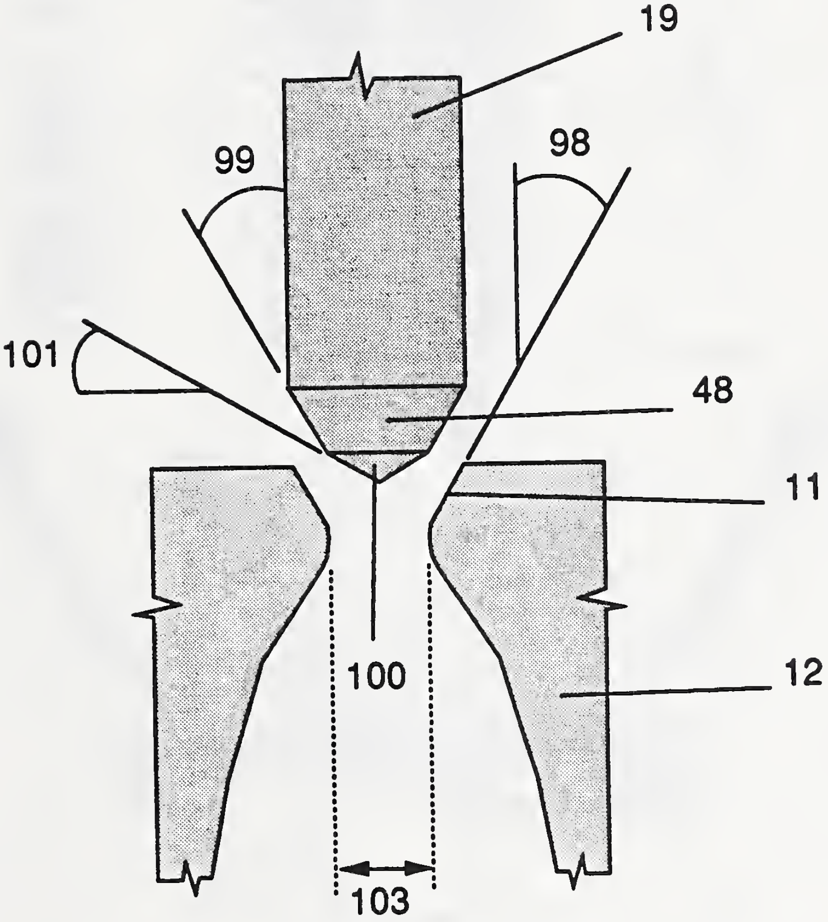


Fig. 2.12: Detailed view of the valve core rod and combination valve-seat/nozzle throat detal for the HSILS prototype (descriptions of numbered features are provided in the text).

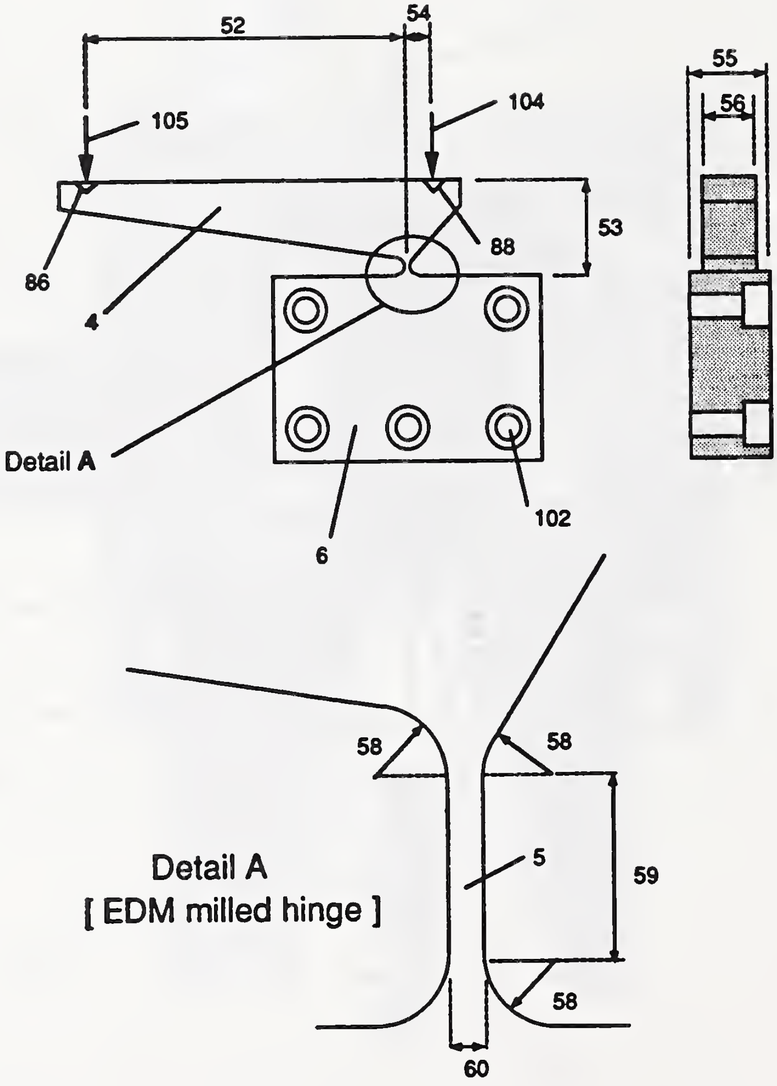


Fig. 2.13: Detail of HSILS displacement amplifier including EDM-milled hinge and conical seats for sapphire pivot bearings (descriptions of numbered features are provided in the text).

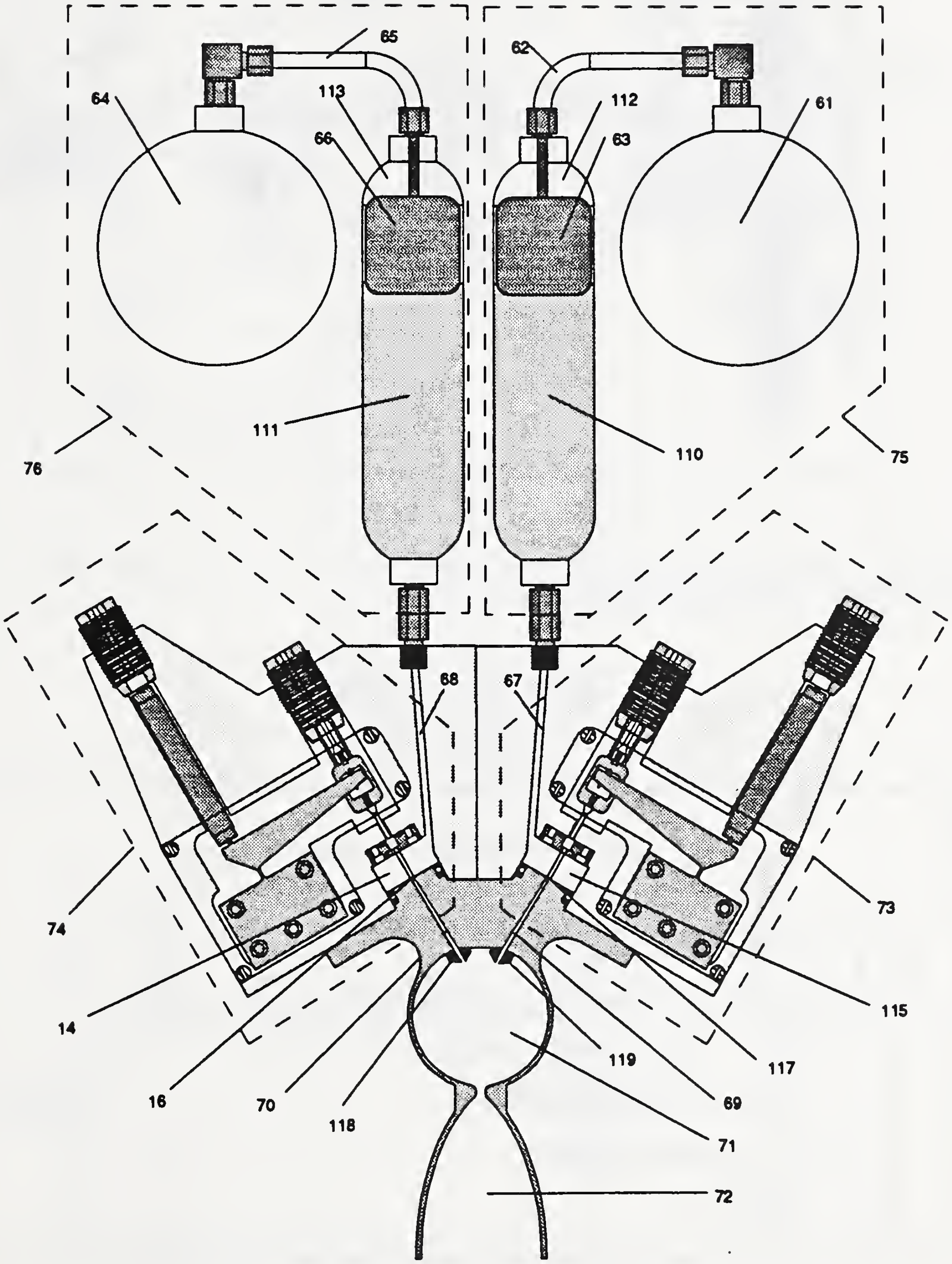


Fig. 2.14: Conceptual drawing of an HSILS-based variable fuel/oxydizer mixing ratio, variable thrust bi-propellant reaction control system for use in spacecraft control (descriptions of numbered features are provided in the text).

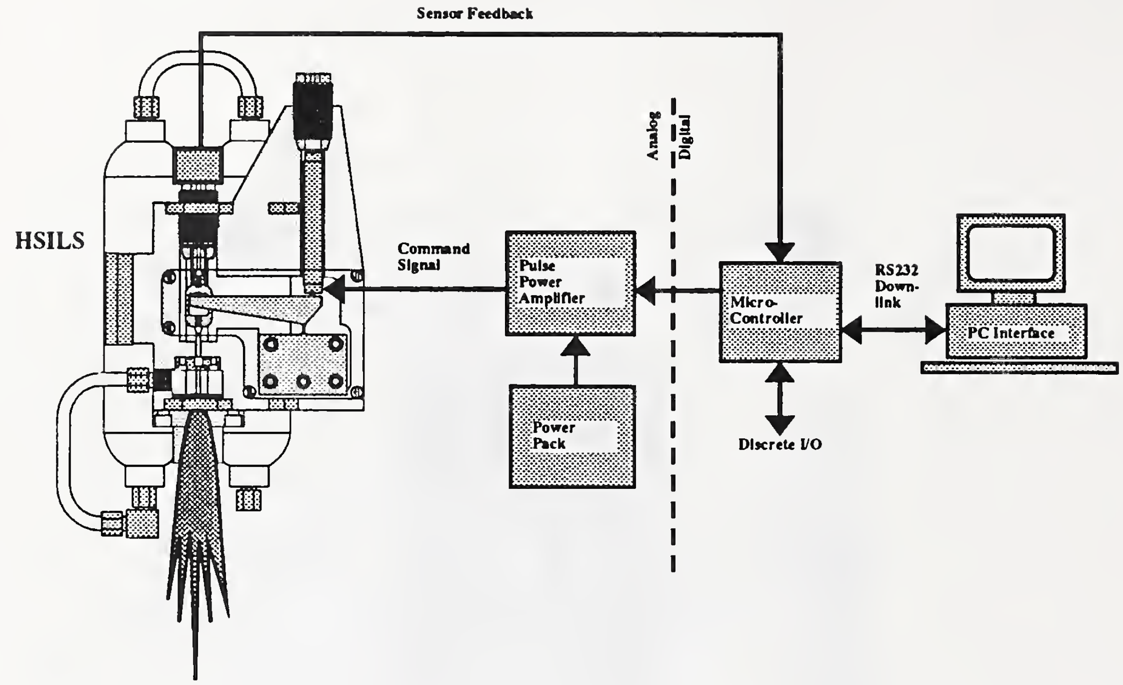


Fig. 2.15: HSILS control system block diagram.

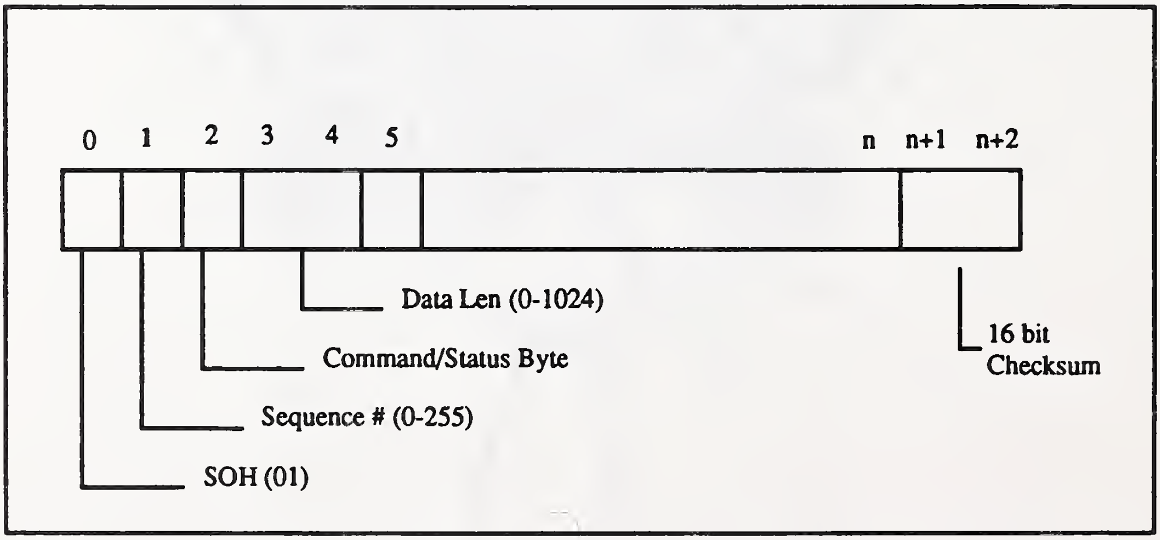


Fig. 2.16: Data packet format for PC-host to onboard microcontroller uplink/downlink communications.

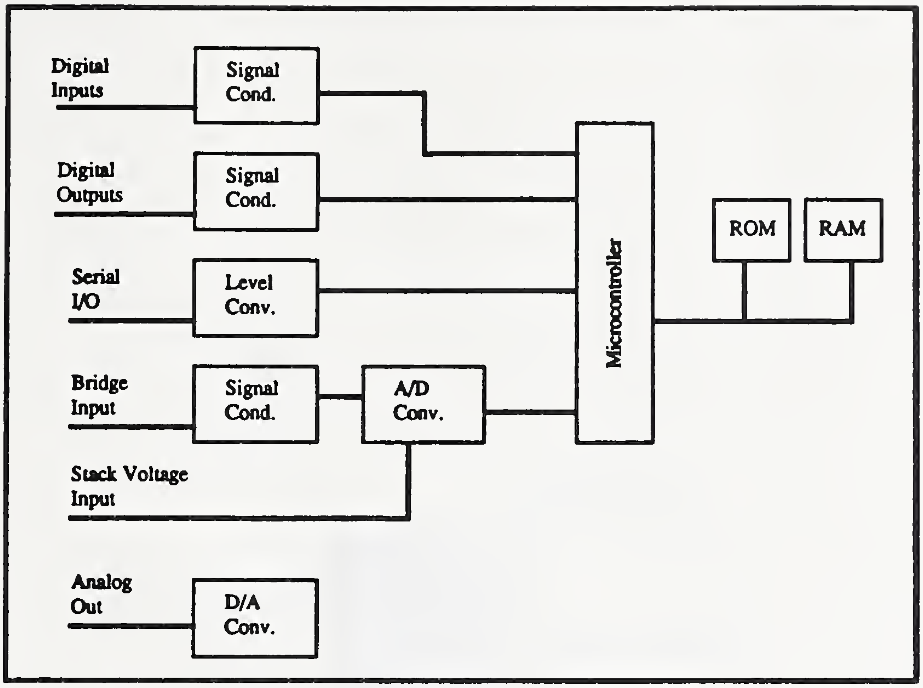


Fig. 2.17: HSILS microcontroller block diagram (see Appendix C for complete schematics).

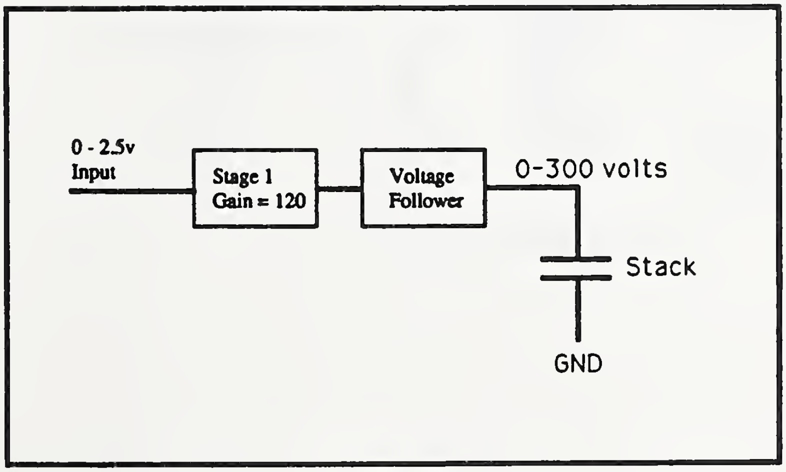


Fig. 2.18: HSILS pulse power amplifier block diagram.

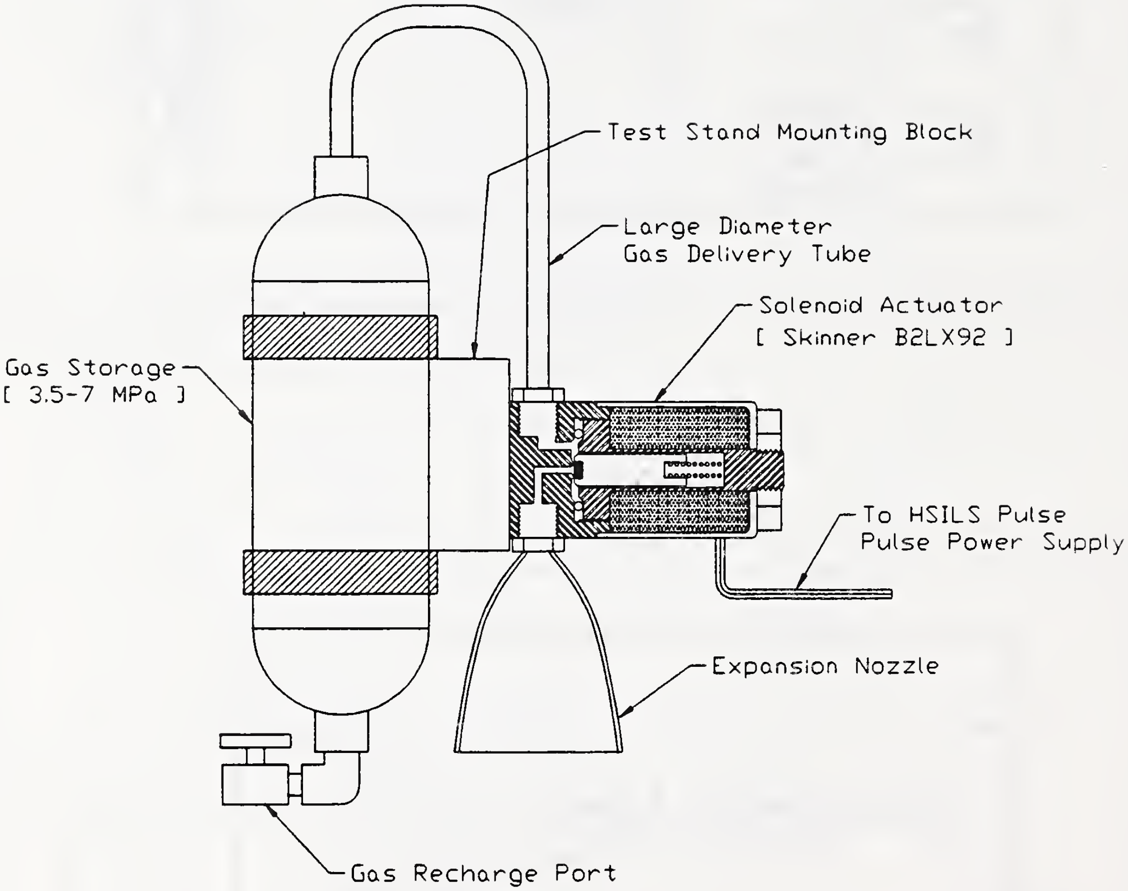


Fig. 2.19: Solenoid operated reaction control system (RCS) test bench.

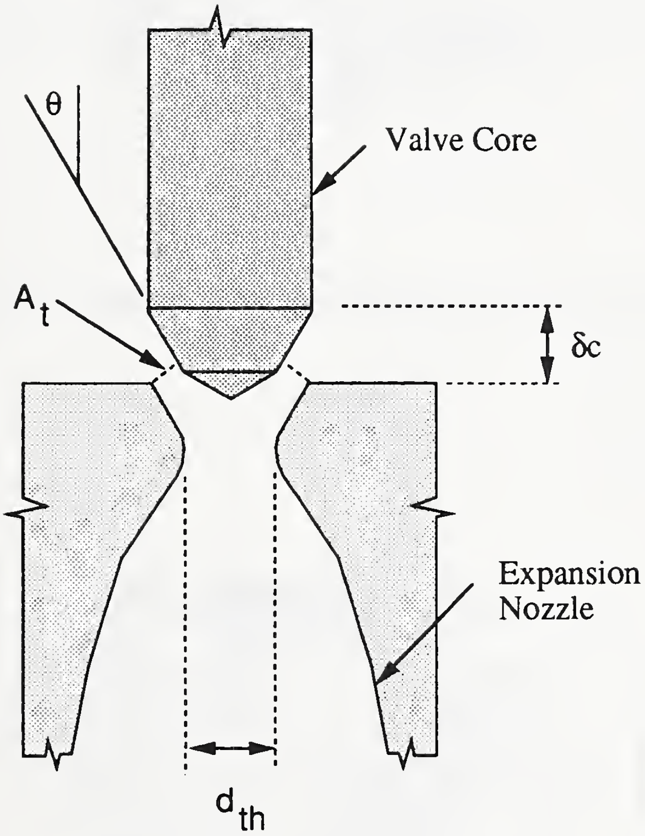


Fig. 2.20: Determination of effective nozzle throat area for HSILS prototype.

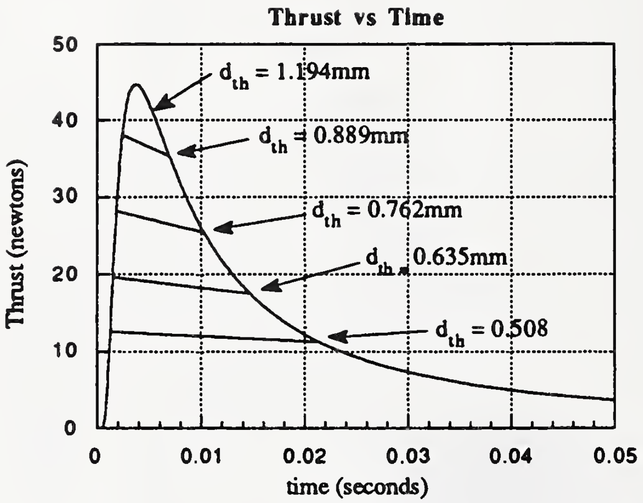


Fig. 2.21: Effect of nozzle throat diameter: thrust versus time simulation.

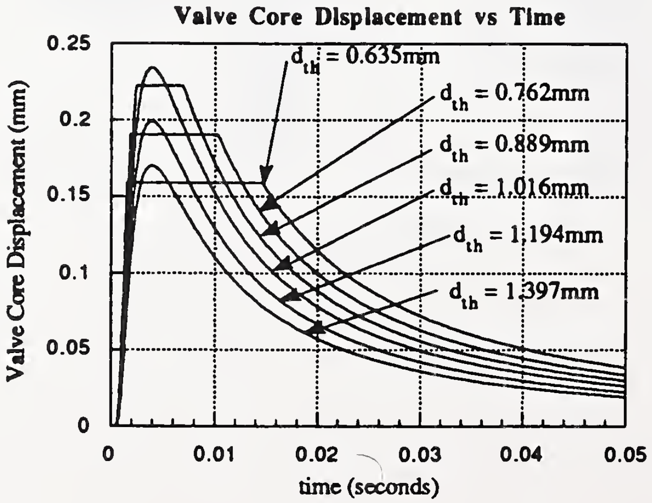


Fig. 2.22: Effect of nozzle throat diameter: core displacement versus time simulation.

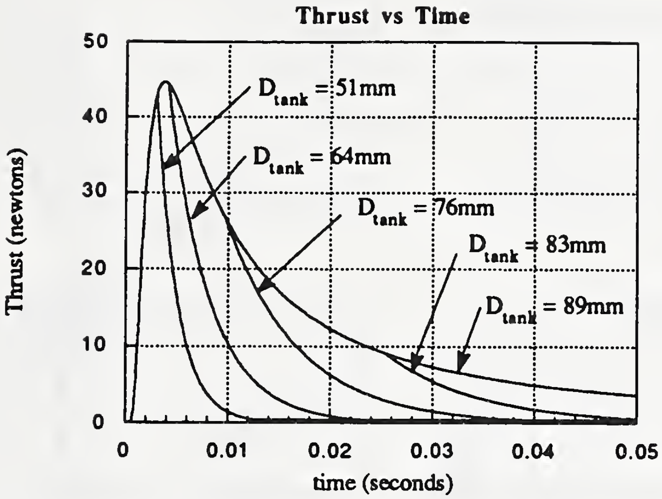


Fig. 2.23: Effect of pressure vessel volume: thrust versus time simulation.

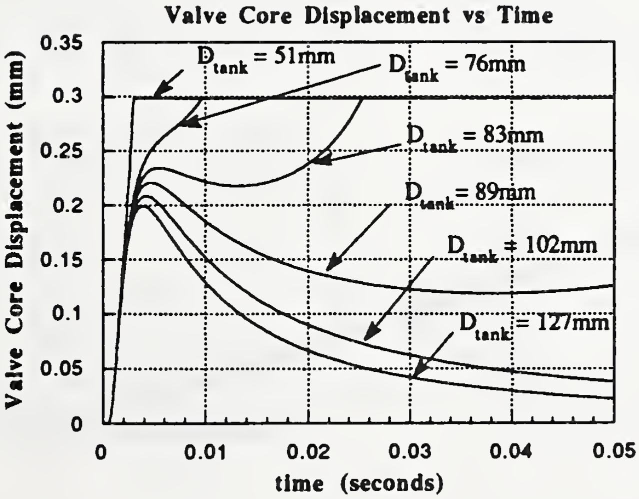


Fig. 2.24: Effect of pressure vessel volume: core displacement versus time.

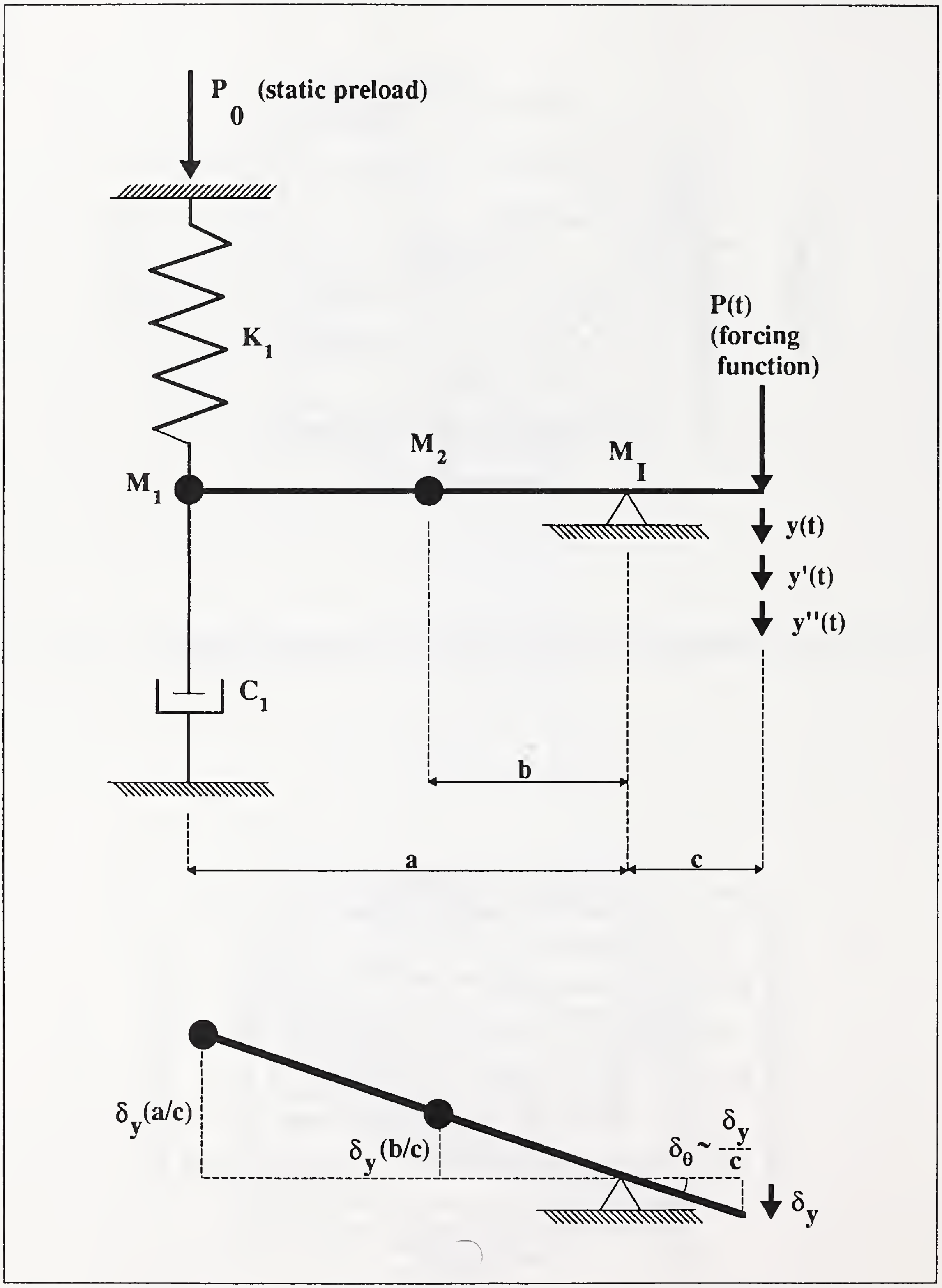


Fig. 2.25: Single degree of freedom model used for HSILS dynamic simulations.

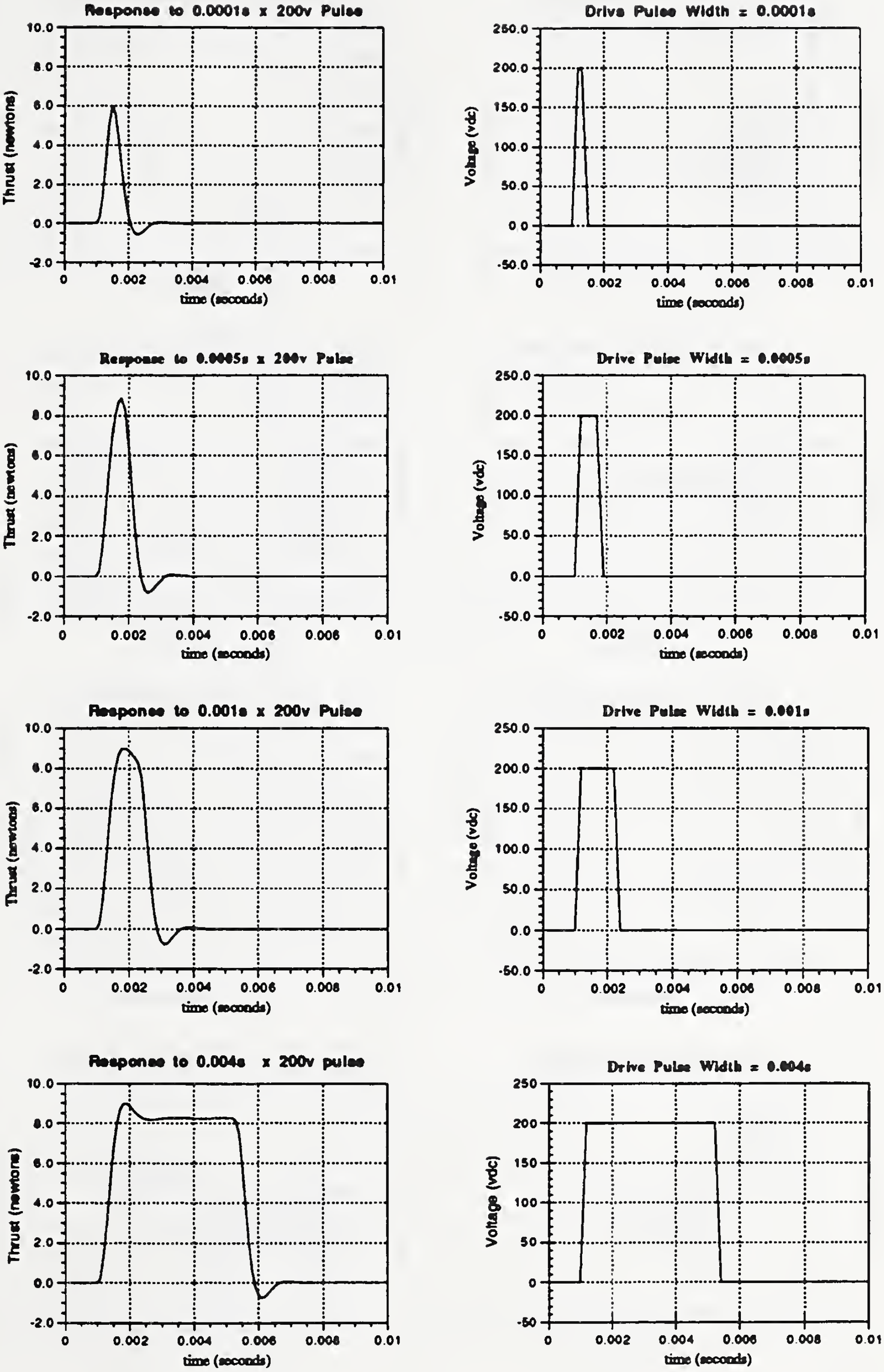


Fig. 2.26: HDYN-predicted HSILS response to various half square wave drive pulses assuming 388% of critical damping due to internal high pressure seals. HSILS predicted thrust versus time is plotted in the left column; drive pulse voltage-time histories are plotted on the right.

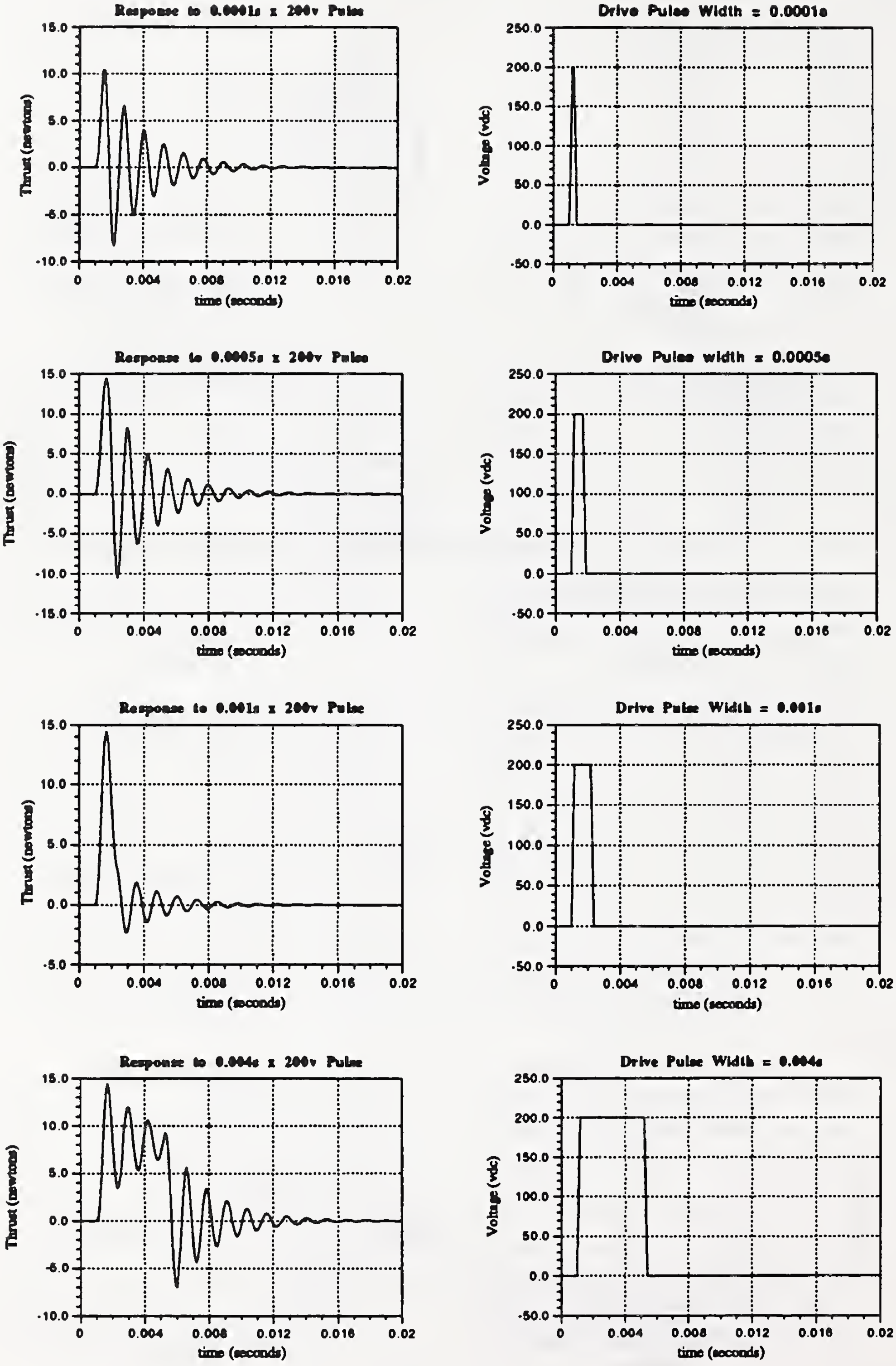
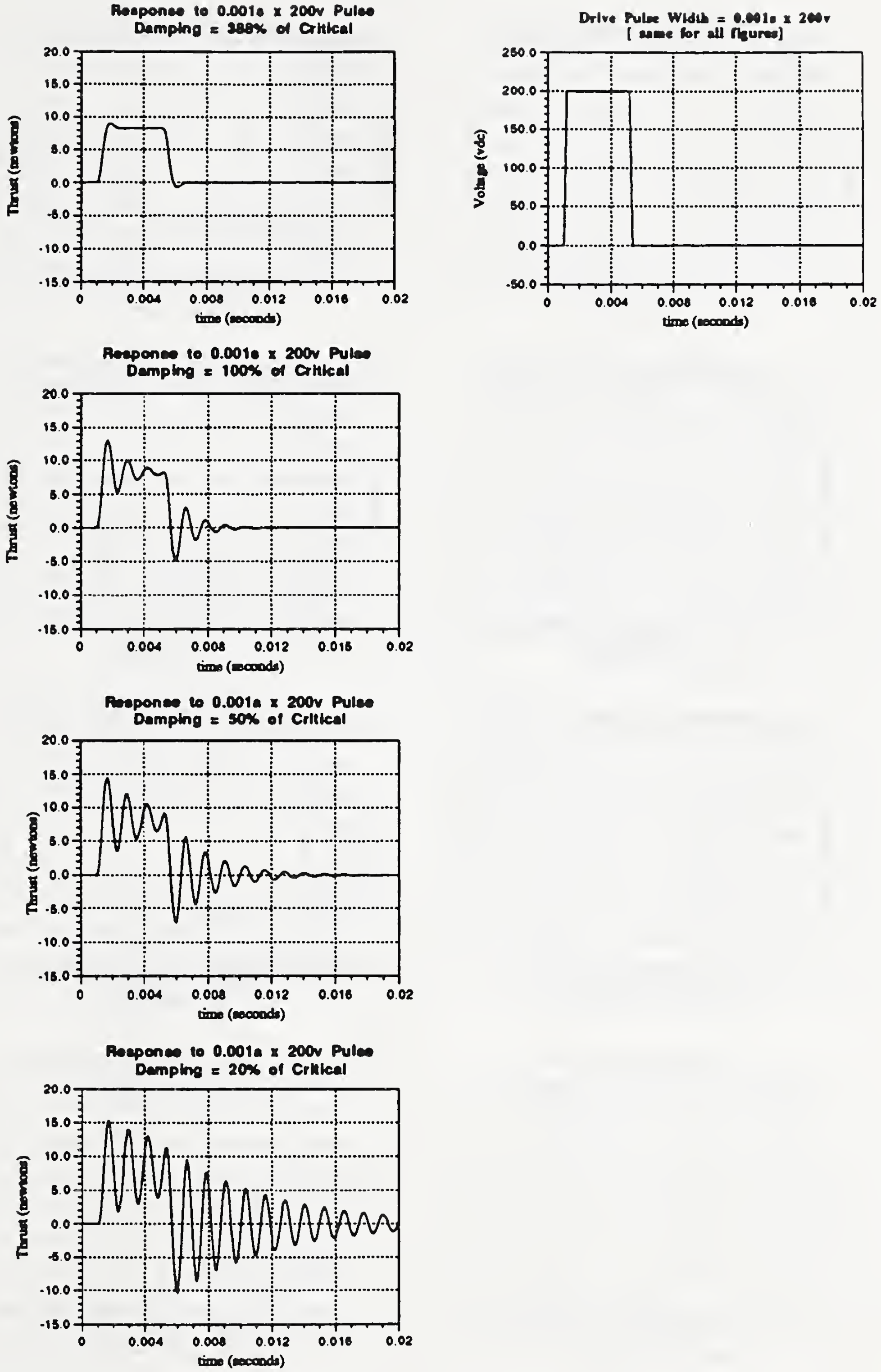


Fig. 2.27: HDYN-predicted HSILS response to various half square wave drive pulses assuming 50% of critical damping due to internal high pressure seals. HSILS predicted thrust versus time is plotted in the left column; drive pulse voltage-time histories are plotted on the right.


Response to 0.001s x 200v Pulse
Damping = 388% of Critical

Drive Pulse Width = 0.001s x 200v
[same for all figures]

Response to 0.001s x 200v Pulse
Damping = 100% of Critical

Response to 0.001s x 200v Pulse
Damping = 50% of Critical

Response to 0.001s x 200v Pulse
Damping = 20% of Critical

Fig. 2.28: HDYN-predicted HSILS response to various half square wave drive pulses for varying levels of viscous damping due to internal high pressure seals. HSILS predicted thrust versus time is plotted in the left column; drive pulse voltage-time history (common to all responses) is plotted on the right.

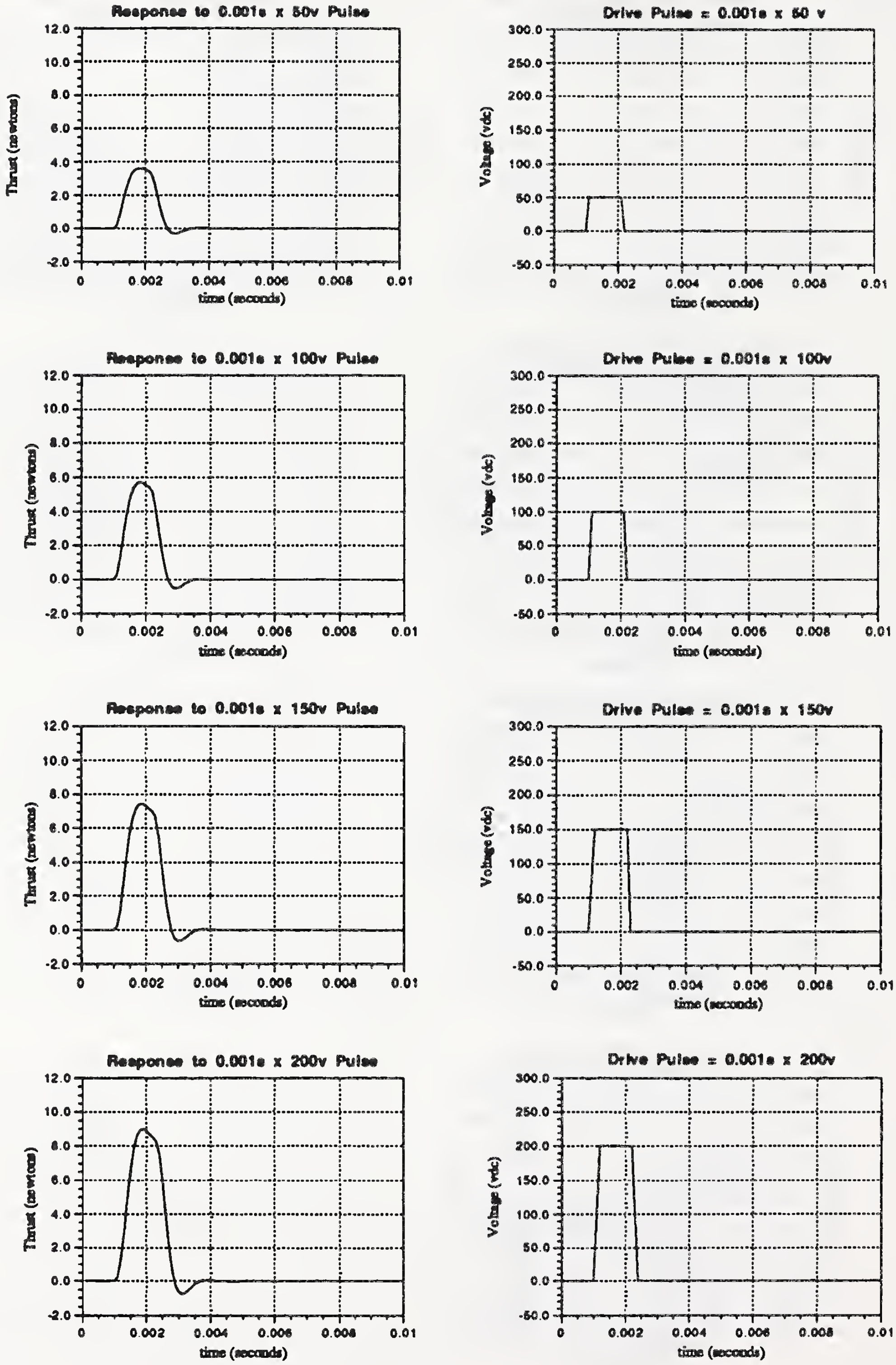


Fig. 2.29: HDYN-predicted HSILS response to various half square wave drive pulses of differing amplitude for 388% of critical damping due to internal high pressure seals. HSILS predicted thrust versus time is plotted in the left column; drive pulse voltage-time histories are plotted on the right.

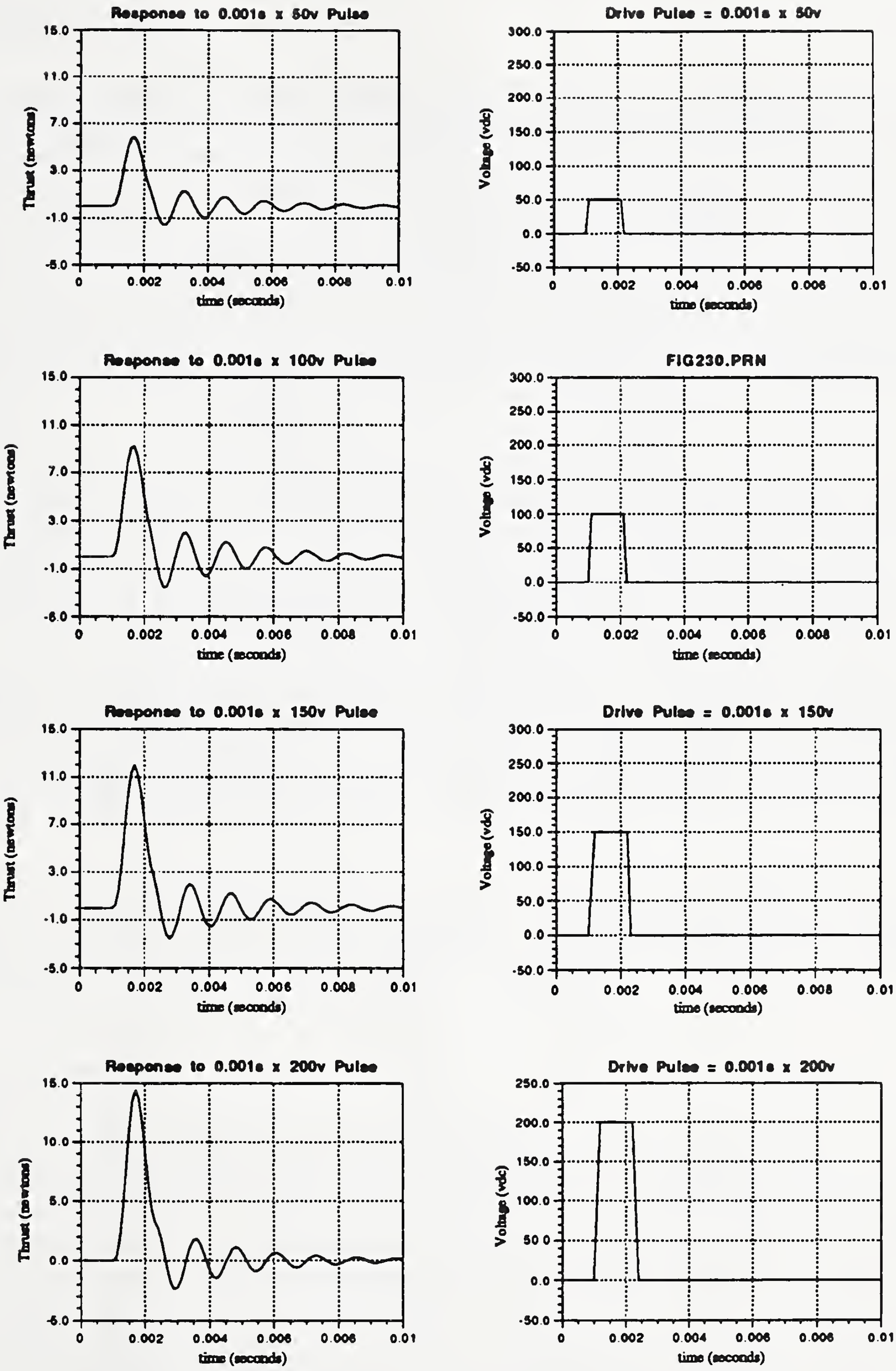


Fig. 2.30: HDYN-predicted HSILS response to various half square wave drive pulses of differing amplitude for 50% of critical damping due to internal high pressure seals. HSILS predicted thrust versus time is plotted in the left column; drive pulse voltage-time histories are plotted on the right.

3.0 Experimental Testing

A laboratory prototype HSILS unit, shown in Figs. 3.1 through 3.4, was constructed in order to test the feasibility of the concept and to determine some of its operating characteristics and limitations. The hardware was fabricated in accordance with the shop drawings provided in Appendix D. Operational principles for the prototype were described in Chapter 2. The device shown in Fig. 3.1 comprises the entire HSILS unit, which can be operated in stand-alone mode once the compressed gas and capacitor storage banks are charged. No great efforts were undertaken to minimize space and weight. Hence, the large box to the left of the thruster unit was used to house the onboard microcontroller, capacitor discharge banks, and power amplifiers. The smaller boxes comprised signal conditioning and power supplies for acoustic and pressure sensors used to measure pulse width and thrust.

Fig. 3.2 shows a side view of the HSILS unit including a 41.4 MPa dry nitrogen charging line connected to the onboard gas accumulator bank. Once the accumulators were charged this line could be disconnected and several tests conducted before recharging became necessary. The high voltage (0-300 V) output of the pulse power supply was connected to the piezoelectric stack via a jacketed BNC type connecter, shown at mid-height on the left of the unit. Fig. 3.3 shows a top view of the exhaust nozzle and the feed lines for the gas accumulators. The nozzle was designed for easy removal and access to the replaceable throat seat shown in Fig. E.18 in Appendix E. Fig. 3.4 shows a view of the internal working components of the HSILS including the titanium displacement amplifier, the piezoelectric stack and its associated sapphire spherical end bearings, and the connector/pivot mechanism which links the displacement amplifier to the valve core rod and its associated spring return system.

Fig. 3.5 shows a view of the solenoid operated reaction control system (RCS) which was used as a "control" reference with which to compare the performance of the HSILS unit. Operating pressures for this unit were limited to approximately 5.5 MPa. This device was described earlier in Chapter 2 (see Fig. 2.19).

3.1 Onboard Sensors

It was initially thought that a responsive load cell could be used to measure the thrust output of both the HSILS and RCS (solenoid reaction control system) systems. However, and as was subsequently ascertained independently by Hallauer and Smith (Hallauer and Smith, 1990), this led to unwanted high-level noise in the response signal which represented the vibration of the structural system consisting of the force actuator and the load cell. It was shown that this effect became worse as efforts were made to make the load cell more sensitive (and therefore more flexible). Hallauer and Smith ultimately made use of dynamic piezoelectric based load cells which were by nature quite stiff. However, at the time of testing at NIST piezoelectric cells were not available and alternative methods were developed for determining the minimum pulse width resolution and amplitude (thrust) reponse of the two candidate actuators.

3.1.1 Acoustic

The first sensor to be used consisted of a commonly available PC Mount condenser microphone element. The actual unit was manufactured by Tandy Corp. Cat. No. 270-090 and had a 2.0 to 10 V operating range; 1.0mA maximum current drain; 40dB minimum signal to noise ratio; -65 +/- 4dB sensitivity over the range of 20 Hz to 20 kHz; and an output impedance of 1k ohms. This element was placed at the exhaust lip of the actuator nozzle as shown in Figs. 3.6 and 3.7 for the HSILS and RCS units, respectively. This meant that there was an inherent delay from the time the valve core first lifted off the seat until the sensor detected perceptible thrust of approximately 75 microseconds, owing to the propagation speed of sound in air. This was included in the rise-lag times recorded in the data tables described below.

3.1.2 Static Pressure

While the acoustic sensor provided an accurate means for determining pulse width, it was incapable of providing quantitative thrust measurement data (qualitatively, the noise amplitude did appear to be proportional to thrust but there was no means of calibrating this). Given the low levels of anticipated thrust an approach was developed which employed a vented chamber that was attached to the end of the nozzle as shown in Figs. 3.8 and 3.9 for the HSILS and RCS (solenoid) thruster units, respectively. A Keller-PSI PAA-9, 10 bar semiconductor absolute diaphram pressure transducer was mounted on the face of the vented chamber opposite the actuator nozzle. Thrust was then measured indirectly by means of equivalent static exhaust pressure.

3.2 Pulse Width Tests

Pulse width tests for each type of actuator were conducted using the following procedure:

1) The acoustic sensor (or pressure sensor in the case of the second set of RCS tests) was positioned and the gas accumulator and power capacitors were charged.
2) A specific drive pulse of the form "P XXX YYY" was entered at a remote personal computer which was connected to the onboard microprocessor via a communications cable. This in turn, caused the onboard microcontroller to execute a power discharge to the actuator in the form of a half square wave of width "XXX" seconds and having a peak (plateau) amplitude of "YYY" volts, DC. The drive pulses had typical rising and falling slopes (switching rates) of 1 V/microsecond. Further information on the pulse downloading operation is presented in Appendix F.
3) A two channel digital signal analyzer acquired data from the acoustic sensor and also recorded the drive pulse at a rate of 100000 kHz beginning 50 microseconds prior to the rising edge of the drive pulse. These records were stored on disk for further processing.
4) The response of the actuator was examined using the parameters previously defined in Fig. 2.2. These included: drive pulse width; drive pulse amplitude; response rise lag; response pulse width; and response fall lag.

3.2.1 RCS Tests

Tables 3.1 and 3.2 present a sample of pulse width response data for the RCS (solenoid) actuator for drive pulse widths beginning at 0.1314 seconds and decreasing to 3.5ms. From these it may be observed that the correlation between the drive and response pulse widths begins to degrade for shorter pulse widths. That is to say, for example, that a 0.0036s drive pulse produces a 0.0301s response, not a 0.0036s response (as shown in Fig. 3.11, third row from the top). This may be interpreted as approaching the minimum "full-on" pulse width, which comprises the time to energize the inductive solenoid coil, for the magnetic core to lift off and compress the return spring, and for the field to collapse and the compression spring to return the core to its closed position. At some point a drive pulse becomes short enough to produce a "ballistic" trajectory from the valve core: that is, the energy is insufficient to pull the valve core away from the exhaust orifice to its fully retracted position, but it is sufficient to just cause lift off. This type of response can be seen in Fig. 3.11, bottom row and can be considered the minimum achievable pulse from the RCS actuator. Tests shown in Figs. 3.10 and 3.11 were conducted with a nominal drive pulse amplitude of 20 V and 39 V, respectively. The response characteristics that result from these different amplitudes are discussed in Chapter 4 below.

3.2.2 HSILS Tests

Fig. 3.12 presents the results of 16 representative tests of the HSILS prototype for drive pulse widths beginning at 0.01026s and decreasing to 0.00025s (250 microseconds). Complete tabular listings of pertinent data for these tests are contained in Table 3.3. Note in these figures the compressed time scale. Most of the responses are less than 3ms and the limiting pulse less than 1ms. As a result the nature of the acoustic noise that is superimposed on the signal is more evident than for the RCS tests. It has been determined from frequency analysis that the noise has a strong dominant frequency of approximately 4 kHz. This corresponds quite closely to the characteristic resonant frequency of the expansion nozzle. Since the nozzle continues to vibrate for a small period of time after ceasation of the delivery of gas there is to be expected a resonant decay (centered about the x-axis) at the tail end of each pulse. This is seen in most of the shorter HSILS response pulse records. In all such cases there is a dramatic reduction in overall amplitude at the point at which the valve closes. This point corresponds to the termination of thrust.

There is a limiting drive pulse amplitude, below which measureable thrust is not produced by the HSILS. This threshold can be seen in the last two rows of Fig. 3.12d for drive pulse amplitudes of 85 and 44 volts, respectively. The limiting drive voltage appears, for the current HSILS configuration, to be approximately 85 volts. That such a threshold drive voltage exists can be understood by considering the blocked force diagram for the piezoelectric stack presented earlier in Fig. 2.5. The stack must initially generate a force necessary to cause the valve core to lift off the valve seat by an amount sufficient to create measureable thrust. However, such motion is prevented by the return spring at the end of the displacement amplifier. The force necessary for lift off is therefore the return spring precompression times the amplification factor for the displacement amplifier, which in the case of the HSILS prototype unit was 12. Since experimental data indicated that an 85 volt amplitude was required to initiate measureable thrust, using the stack equations given in Fig.

2.5 one can back calculate that the valve core displacement at that time was 0.007 mm.

Further discussion on limiting pulse widths for both the HSILS and RCS actuators is presented in Chapter 4.

3.3 Amplitude Control Tests

The ability to "intelligently" create an output thrust profile from an active control system implicitly requires the ability to vary the thrust amplitude. Thus far we have addressed the response speed or reaction time of two actuator systems. This will, ultimately, determine the time-domain precision with which one can create an "intelligent" pulse. A subsequent series of tests were conducted on both the HSILS and RCS units to determine their steady state response to various amplitude control voltages. The test protocol was as follows:

1) The Keller absolute pressure sensor was positioned as previously described in Figs. 3.8 and 3.9 and the gas accumulator and power capacitors were charged.
2) A continuous DC voltage was applied to the respective actuators for a period of approximately 0.5 second (corresponding to the previous nomenclature of "P 500000 YYY", in which YYY was the drive pulse amplitude).
3) A two channel digital signal analyzer acquired data from the acoustic sensor and also recorded the drive pulse at a rate of 100000 kHz beginning 50 microseconds prior to the rising edge of the drive pulse. These records were stored on disk for further processing.
4) The mean thrust was determined from the average output signal after attaining steady state (valve open) response. This value, and the drive voltage amplitude, were then recorded.

3.3.1 RCS Tests

Table 3.4 contains the experimental data for this series of tests. Fig. 3.13 shows the amplitude response characteristics for the RCS (solenoid) actuator. This clearly indicates the presence of a threshold drive voltage of approximately 17 V. After this threshold has been reached the level of thrust is shown to be constant with increasing voltage. This type of performance, as previously discussed in Chapter 1, is the primary impediment to using solenoid based actuators for intelligent control. Other limitations, in the form of rise and fall lag times and minimum pulse width are discussed in detail in Chapter 4.

3.3.2 HSILS Tests

Table 3.5 contains the experimental data resulting from this test series. Fig. 3.14 shows the amplitude response characteristics for the HSILS prototype. This indicates linear amplitude response with increasing drive voltage. There is, as was discussed above, a threshold drive voltage of approximately 85 volts which corresponds to the drive voltage necessary to create stack compressive forces which can overcome the amplified valve core return spring force. There is some scatter to the data beyond a drive voltage of approximately 180 volts, although the trend (increasing thrust with increasing drive voltage) is still preserved. It is the author's belief that the scatter was a product of turbulence within the static pressure sensing attachment at higher gas flow rates and that, given more sophisticated means for measuring actual thrust, such variability could be eliminated. A regression equation relating output thrust to drive voltage is presented in Fig. 3.14. The manufacturer's maximum recommended drive voltage for the piezoelectric stack used in these tests was 300 V. However, for reasons of conservatism, the amplitude tests were limited to drive voltages less than 240 volts. At higher voltages there exists the potential for shorting (arc over) between the stack elements and the subsequent loss of either a stack block sub-assembly or the entire stack.

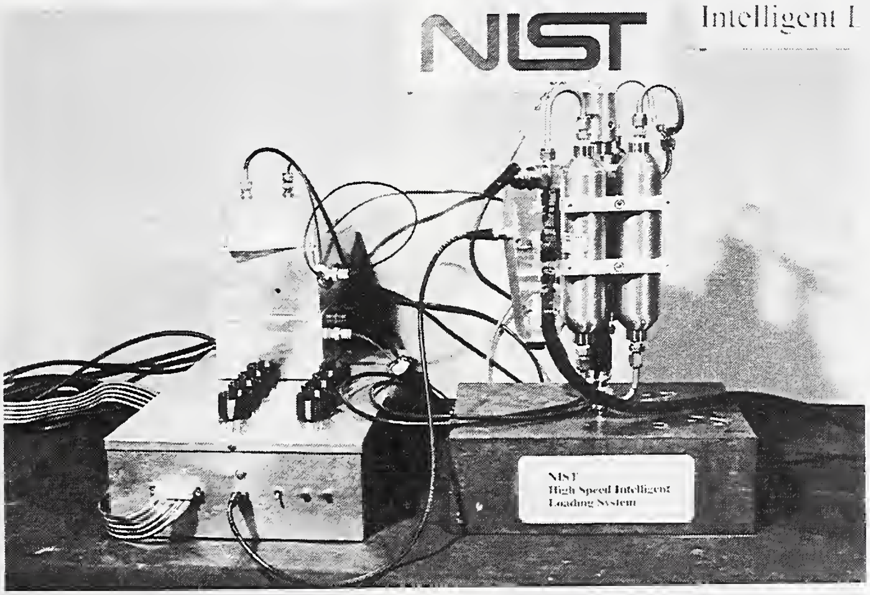


Fig. 3.1: HSILS microcontroller and pulse power subsystems (left) and HSILS cold gas thruster unit with attached high pressure accumulators (right).

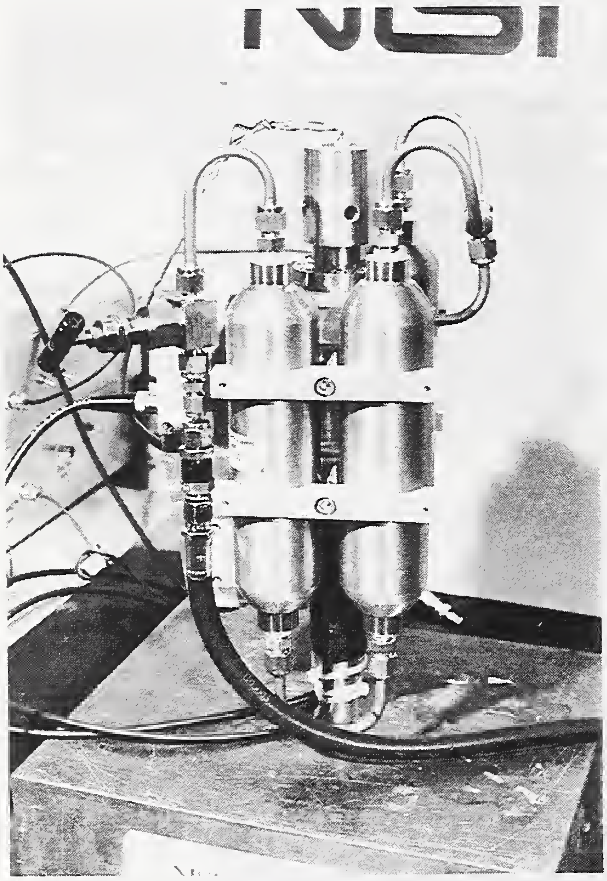
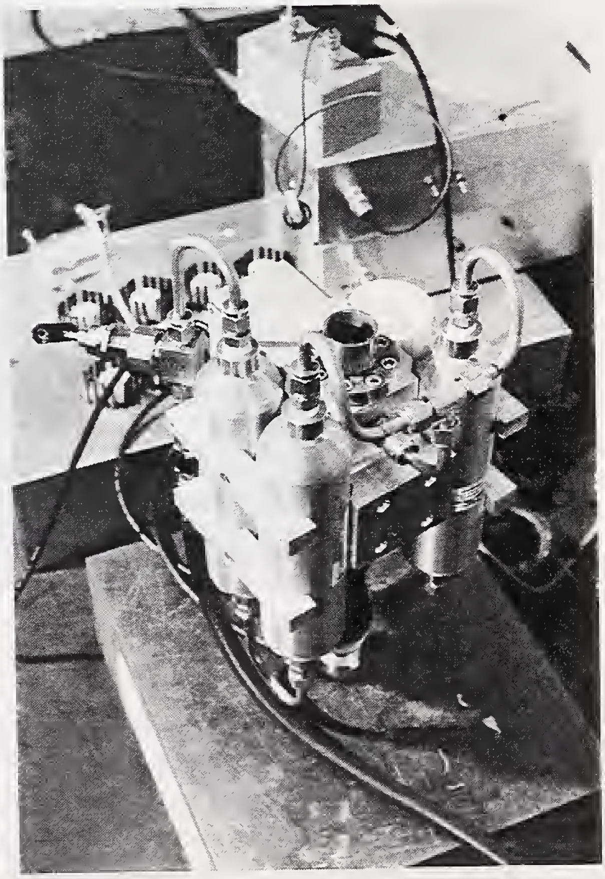


Fig. 3.2: (LEFT) Close-up side view of HSILS thruster unit. Visible from left are a) BNC high voltage connector from pulse power unit; b) high pressure umbilical charging line and onboard shutoff valve; c) high pressure onboard gas accumulators; and d) equivalent static thrust sensor unit mounted on top of the HSILS nozzle (top).

Fig. 3.3: (RIGHT) Close-up side view of HSILS thruster unit. Visible from left are a) BNC high voltage connector from pulse power unit; b) high pressure umbilical charging line and onboard shutoff valve; c) high pressure onboard gas accumulators; and d) equivalent static thrust sensor unit mounted on top of the HSILS nozzle (top).



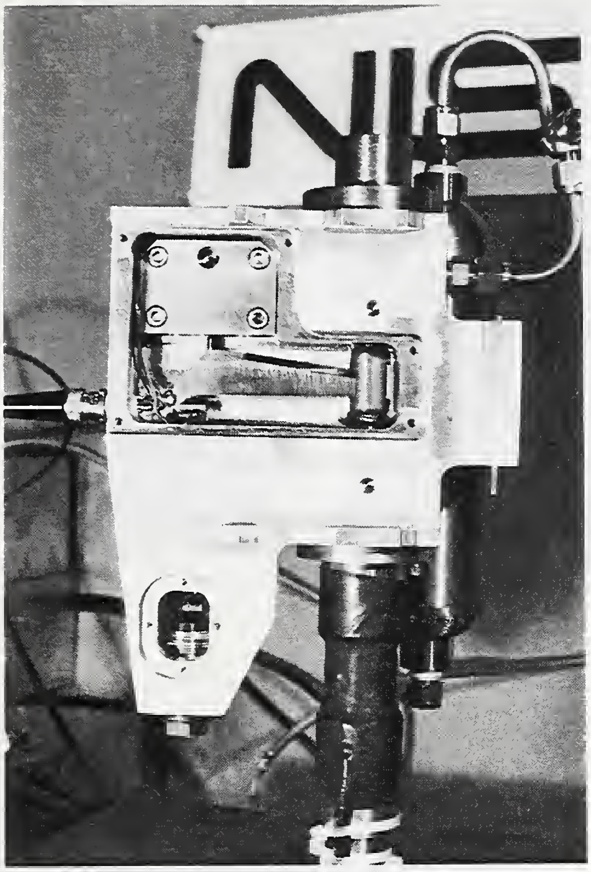
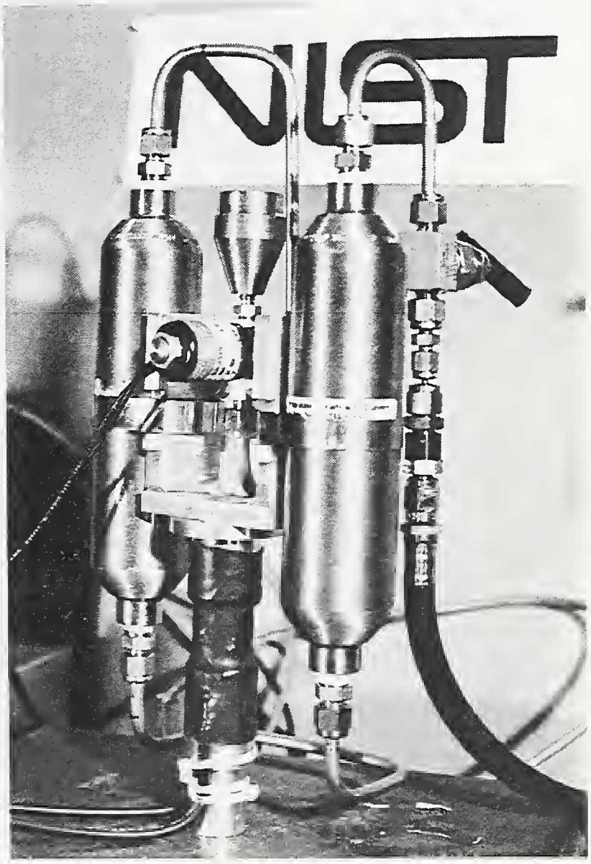


Fig. 3.4: (LEFT) Close-up side view of HSILS thruster unit with gas accumulators and cover plates removed. BNC high voltage connector from pulse power unit is visible at left. Lower port (with cover removed) allows access to piezoelectric stack adjustment screw and lower sapphire spherical pivot. Upper port shows upper piezoelectric stack connection to the short side of of the titanium displacement amplifier (and its associated sapphire pivot) and the right hand connection of the displacement amplifier to the valve core rod and its preload return spring.

Fig. 3.5: (RIGHT) Close-up side view of solenoid-based RCS unit. Visible from left are a) low voltage connector from pulse power unit; b) Solenoid valve block and nozzle (top); c) high pressure gas accumulators; and high pressure umbilical charging line with onboard shutoff valve.



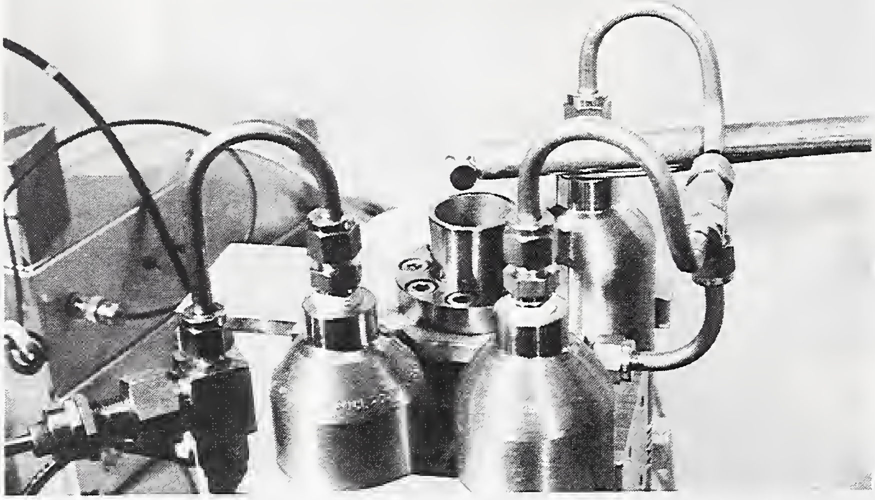
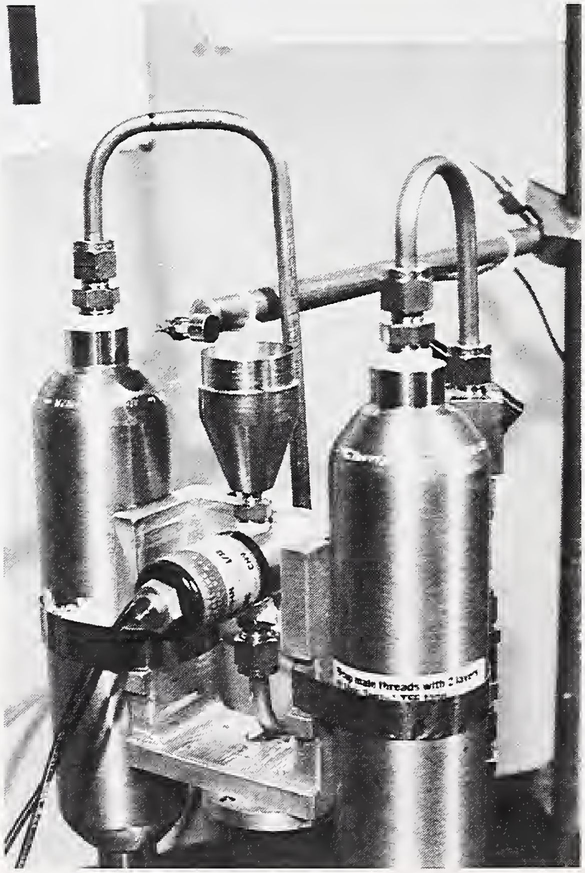


Fig. 3.6: (ABOVE) Close-up side view of HSILS thruster unit with acoustic sensor ready for pulse width resolution tests.

Fig. 3.7: (RIGHT) Close-up side view of solenoid-based RCS unit with acoustic sensor ready for pulse width resolution tests.



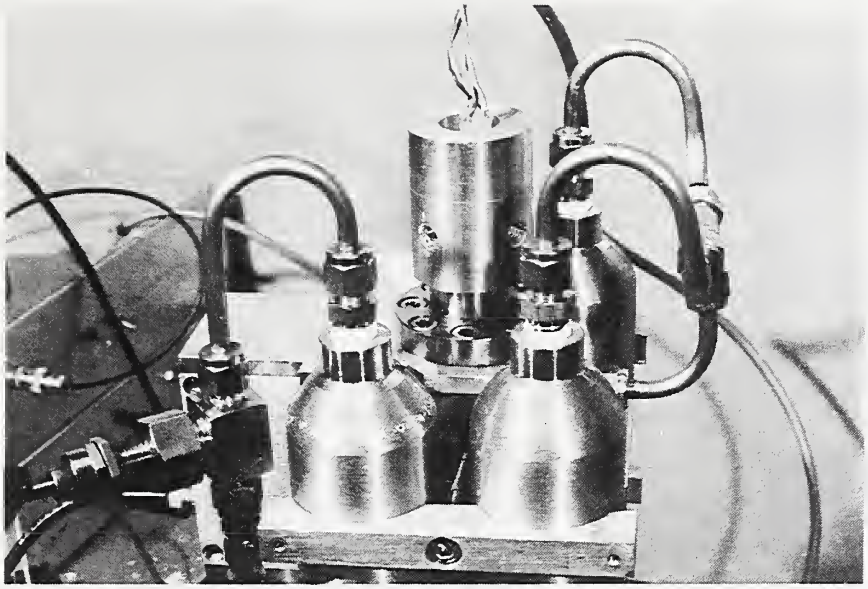


Fig. 3.8: (ABOVE) Close-up side view of HSILS thruster unit with pressure sensor cavity ready for amplitude control tests.

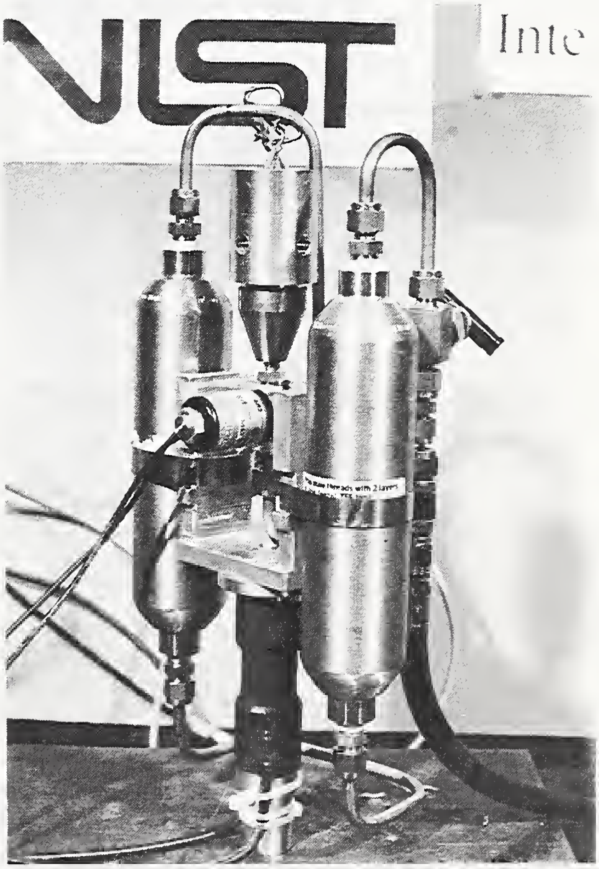


Fig. 3.9: (RIGHT) Close-up side view of solenoid-based RCS unit with pressure sensor cavity ready for amplitude control tests.

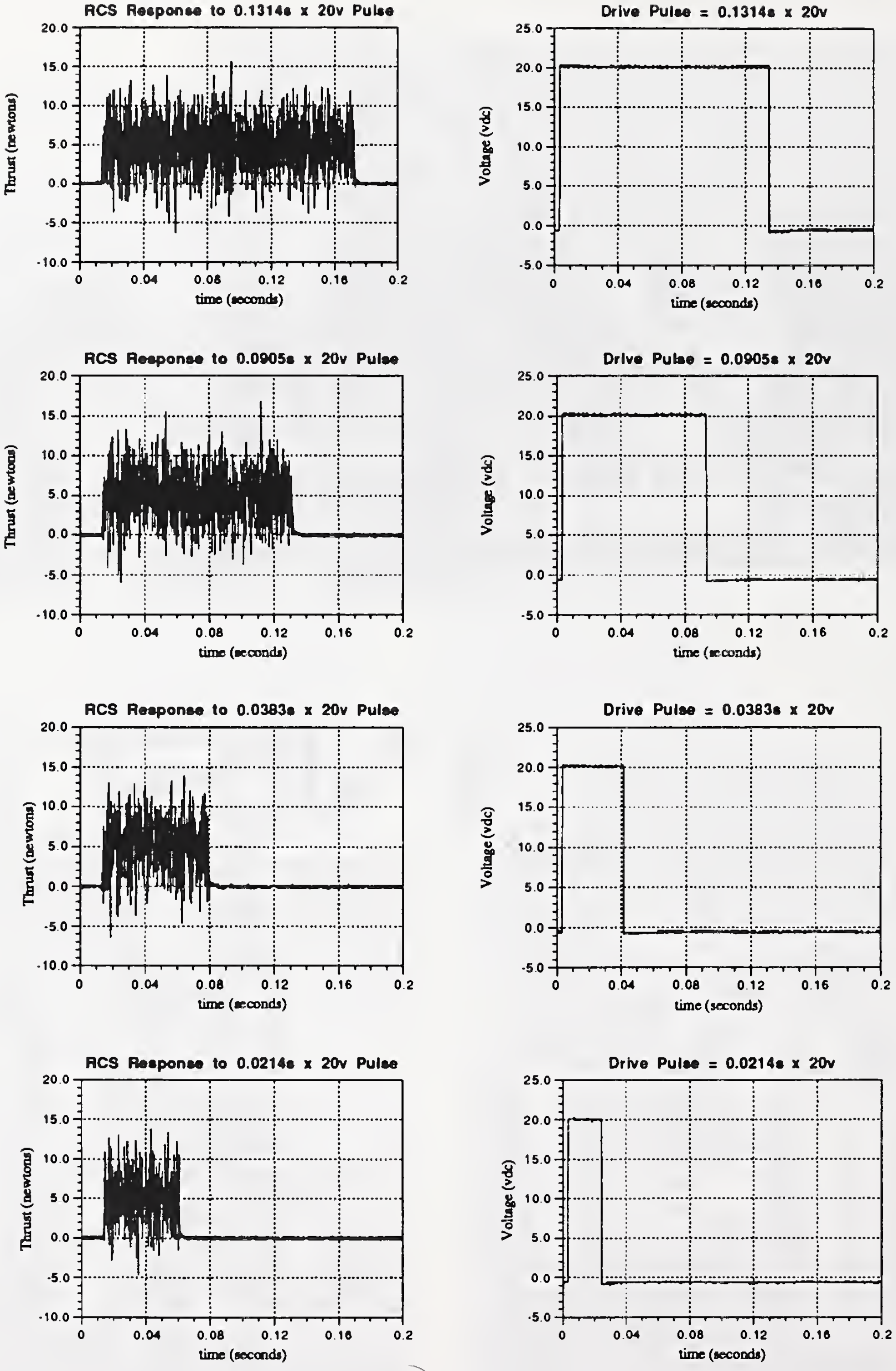


Fig. 3.10a: Pulse width response of solenoid-based RCS actuator to various half square wave drive pulses of decreasing duration (top to bottom). Thrust was measured indirectly by means of an attached pressure sensing cavity. Response thrust versus time is plotted in the left hand column; drive voltage-time history is plotted at right for each test. Nominal drive amplitude is 21 V.

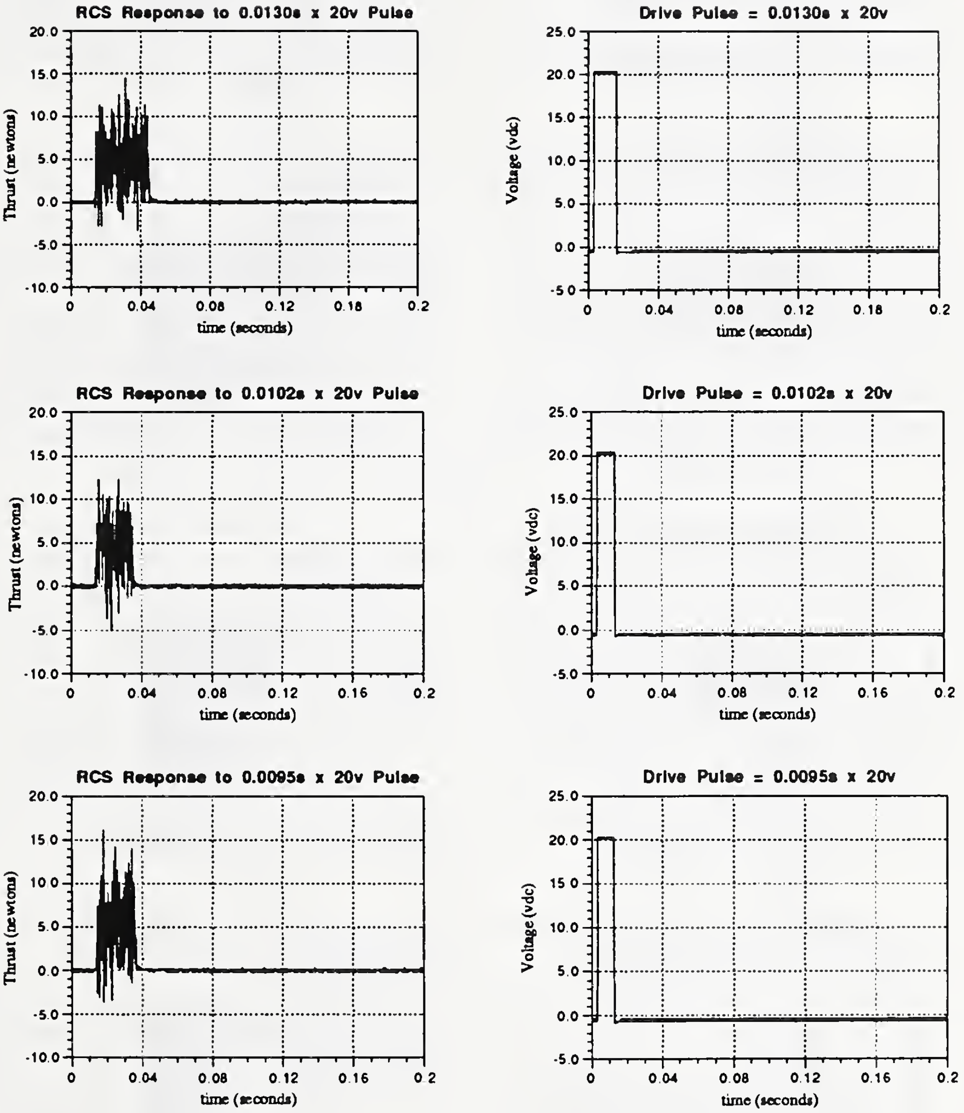


Fig. 3.10b: Pulse width response of solenoid-based RCS actuator to various half square wave drive pulses of decreasing duration (top to bottom). Thrust was measured indirectly by means of an attached pressure sensing cavity. Response thrust versus time is plotted in the left hand column; drive voltage-time history is plotted at right for each test. Nominal drive amplitude is 21 V. (continued from Fig. 3.10a).

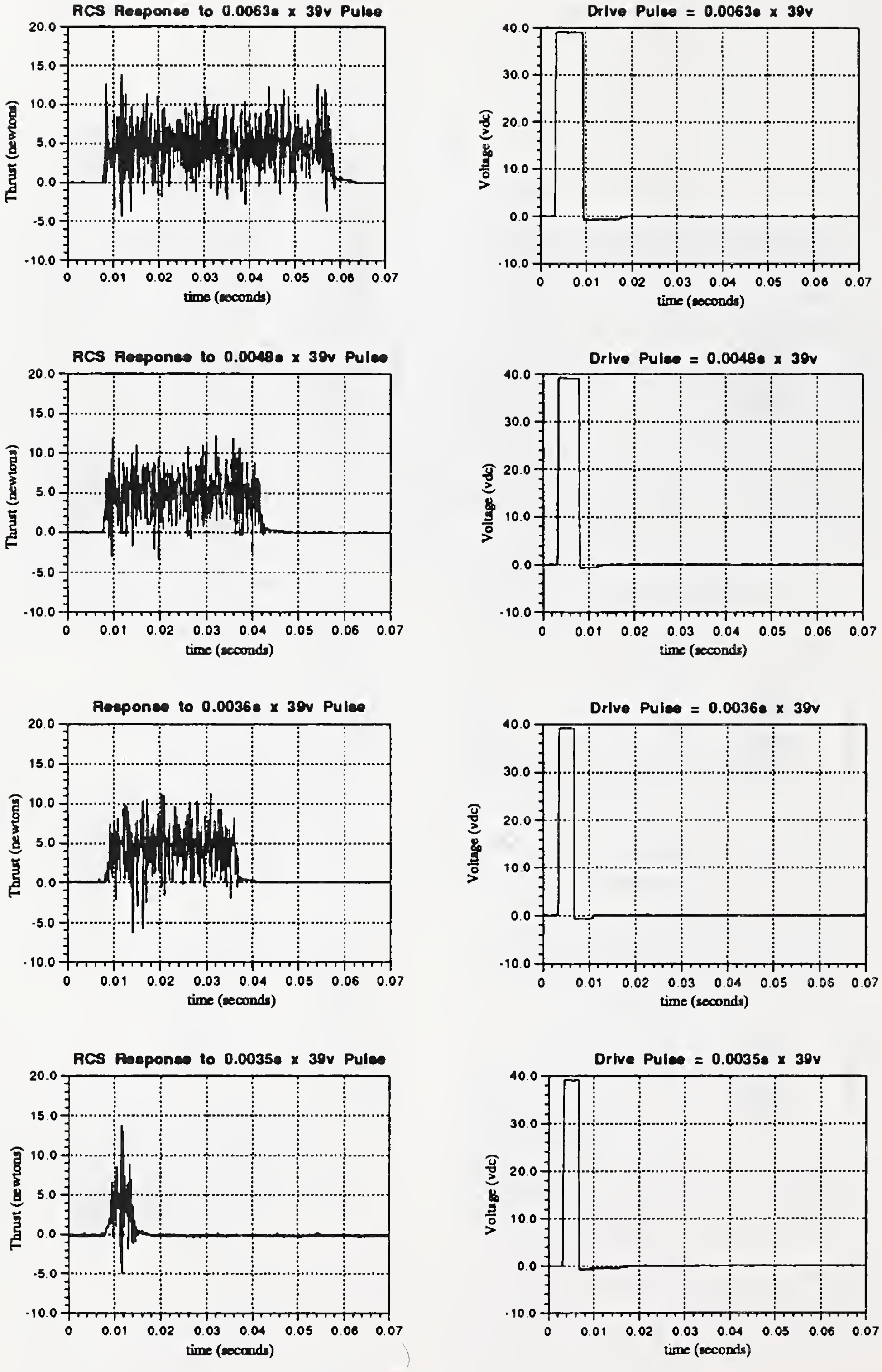


Fig. 3.11: Pulse width response of solenoid-based RCS actuator to various half square wave drive pulses of decreasing duration (top to bottom). Thrust was measured indirectly by means of an attached pressure sensing cavity. Response thrust versus time is plotted in the left hand column; drive voltage-time history is plotted at right for each test. Tests are similar to those shown in Fig. 3.10 with the exception of having a nominal drive amplitude of 39 V.

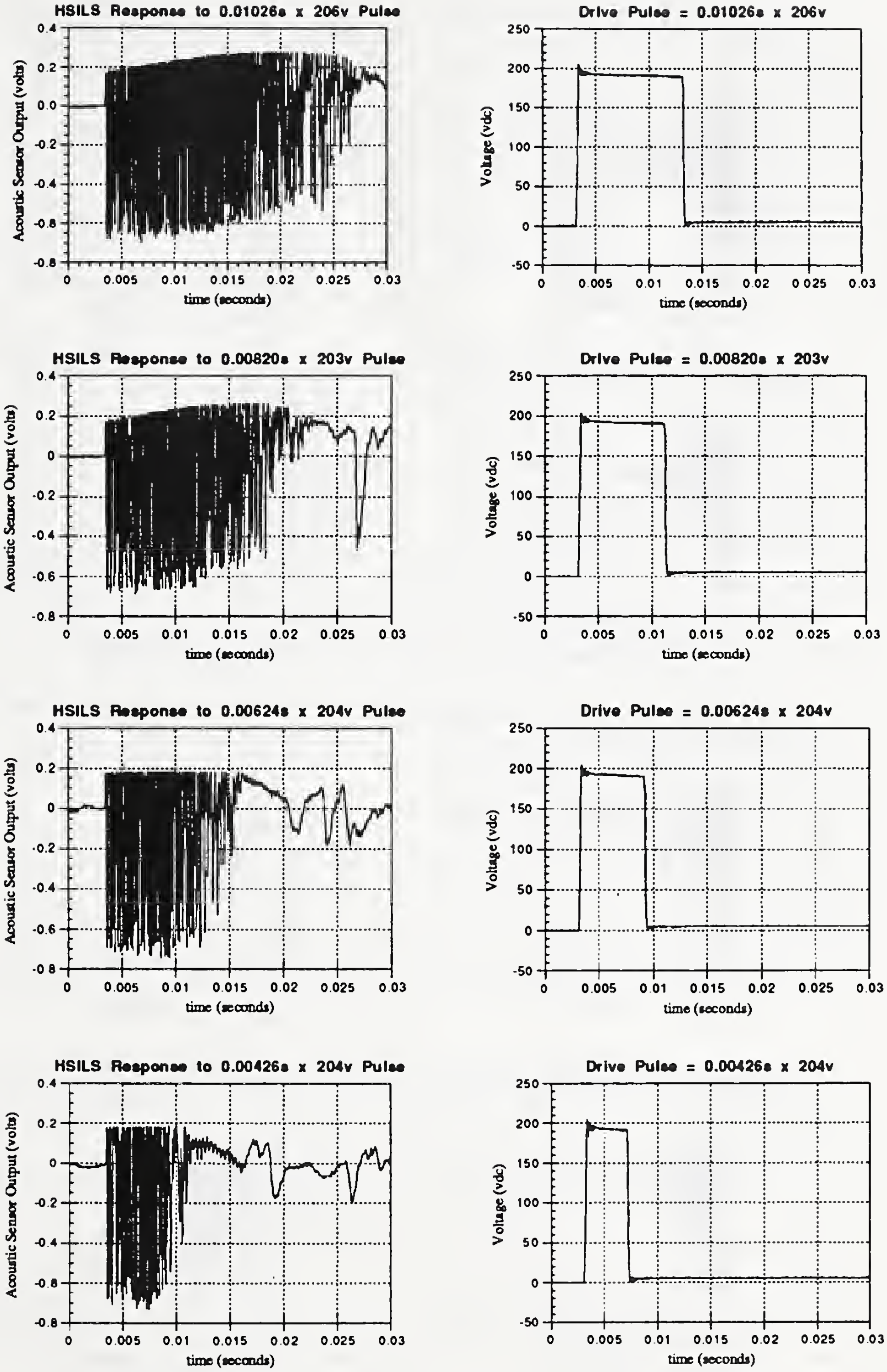


Fig. 3.12a: Pulse width response of HSILS actuator to various half square wave drive pulses of decreasing duration (10.2ms top to 4.3ms bottom). Pulse width was measured using an acoustic microphone. Acoustic sensor response versus time is plotted in the left hand column; drive voltage-time history is plotted at right for each test.

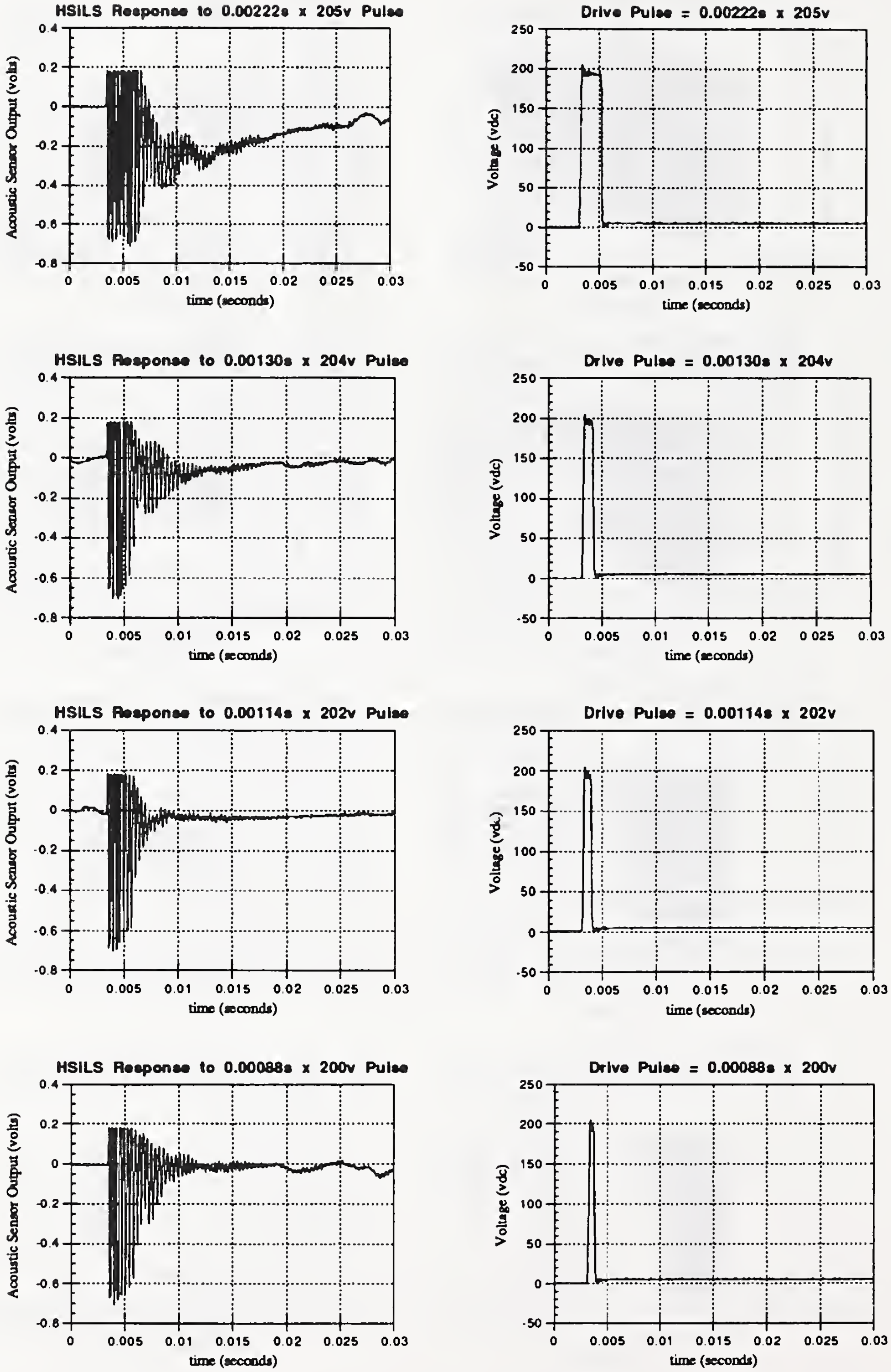


Fig. 3.12b: Pulse width response of HSILS actuator to various half square wave drive pulses of decreasing duration (2.2ms top to 0.8ms bottom). Pulse width was measured using an acoustic microphone. Acoustic sensor response versus time is plotted in the left hand column; drive voltage-time history is plotted at right for each test. (continued from Fig. 3.12a).

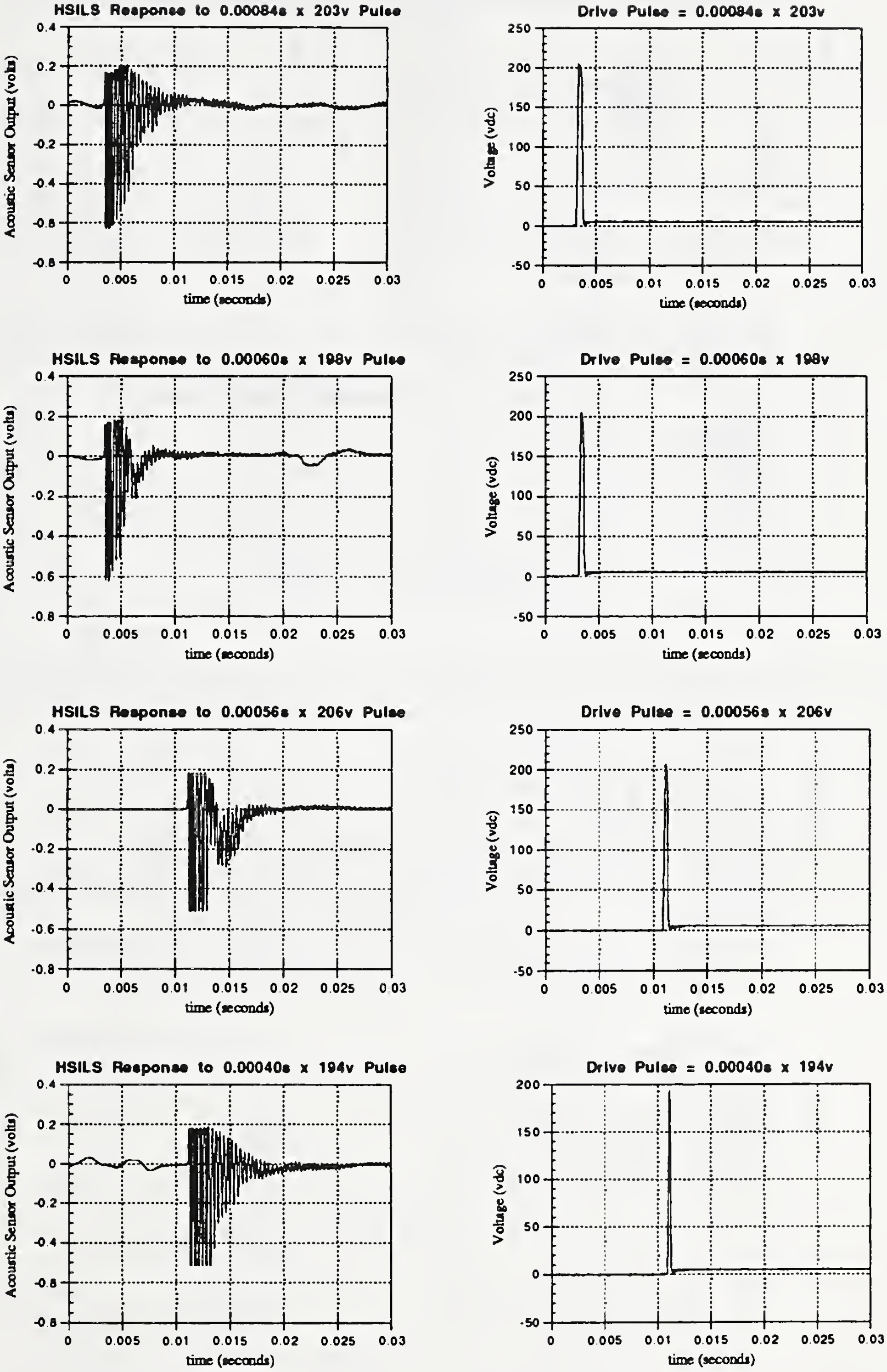


Fig. 3.12c Pulse width response of HSILS actuator to various half square wave drive pulses of decreasing duration (0.8ms top to 0.4ms bottom). Pulse width was measured using an acoustic microphone. Acoustic sensor response versus time is plotted in the left hand column; drive voltage-time history is plotted at right for each test. (continued from Fig. 3.12b).

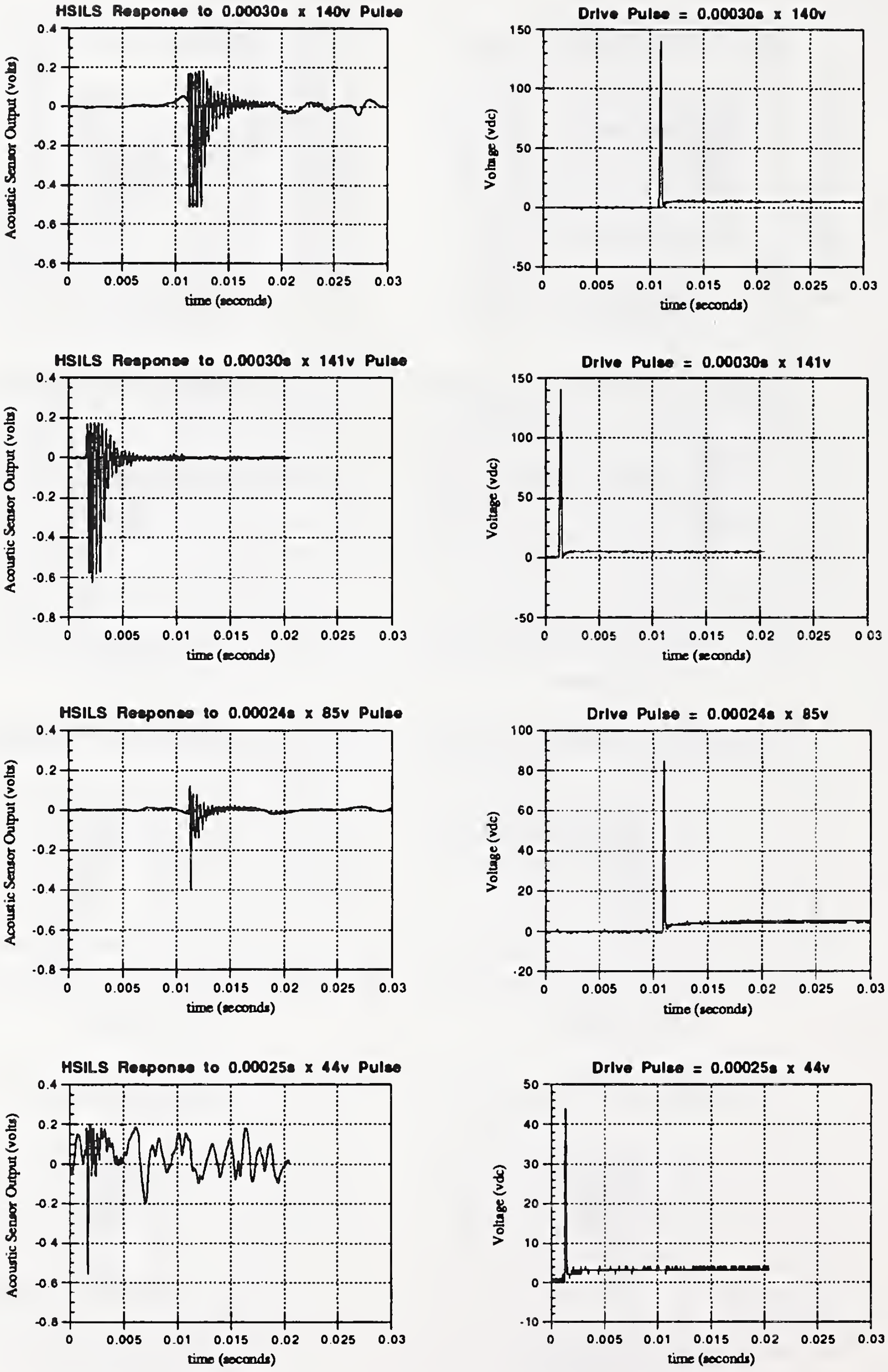


Fig. 3.12d Pulse width response of HSILS actuator to various half square wave drive pulses of decreasing duration (0.3ms top to 0.2ms bottom). Pulse width was measured using an acoustic microphone. Acoustic sensor response versus time is plotted in the left hand column; drive voltage-time history is plotted at right for each test. (continued from Fig. 3.12c).

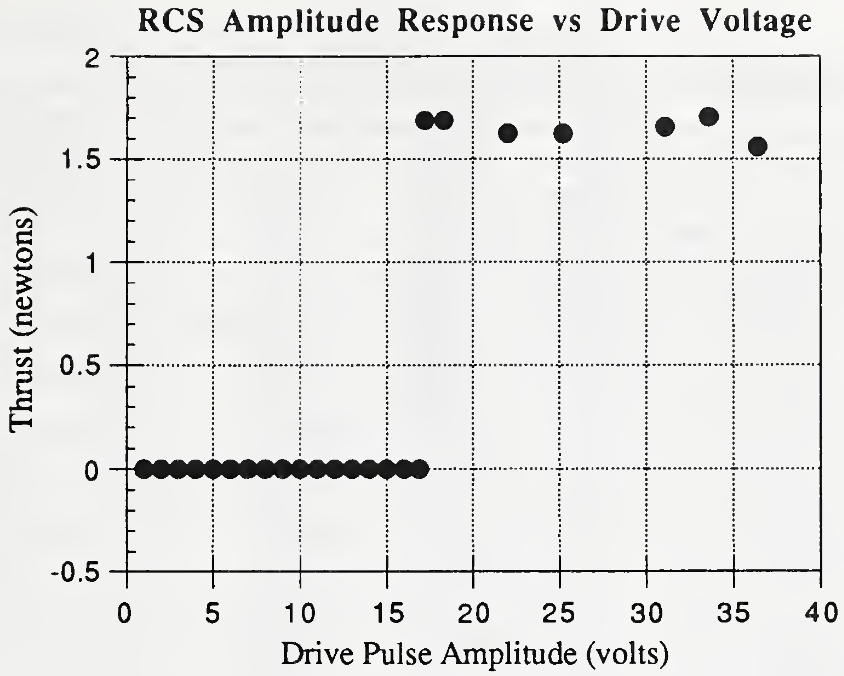


Fig. 3.13: Amplitude (thrust) response characteristics for solenoid-based RCS actuator.

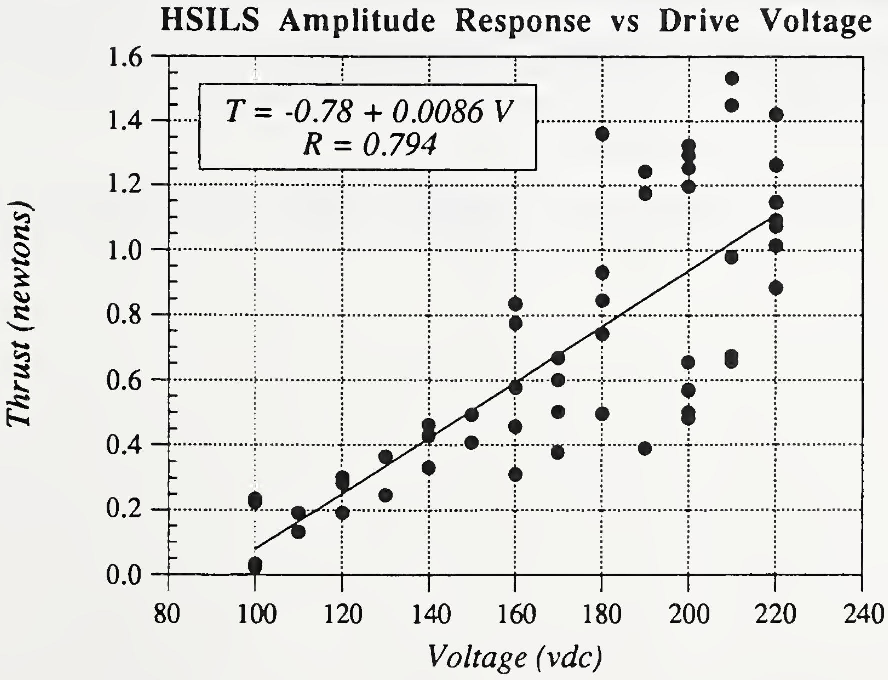


Fig. 3.14: Amplitude (thrust) response characteristics for HSILS prototype.

Table 3.1: Acoustic Sensor Data for RCS Pulse Width Resolution Tests

Nominal Pulse Description[*]	Drive Pulse Width (sec.)	Peak Drive Ampl. (volts)	Active Energy Xfer (joules)	Response Rise Lag (sec.)	Response Pulse Width (sec.)	Response Fall Lag (sec.)	Data File Name
P 13400 x 21	0.0134	21	0.0682	0.0076	0.1616	0.1482	rcs24.wq1
P 129300 x 21	0.1293	21	0.6584	0.0061	0.1906	0.0613	rcs25.wq1
P 55900 x 21	0.0559	21	0.2846	0.0080	0.1132	0.0573	rcs26.wq1
P 13100 x 21	0.0131	21	0.0667	0.0072	0.0682	0.0551	rcs28.wq1
P 7000 x 21	0.007	21	0.0356	0.0074	0.0431	0.0361	rcs29.wq1
P 7[illegible]00 x 21	0.0071	21	0.0361	0.0076	0.0524	0.0453	rcs211.wq1
P 10700 x 20	0.01055	20	0.0446	0.00825	0.11035	0.0998	qs23.prn
P 8200 x 20	0.00805	20	0.0341	0.0079	0.0991	0.09105	qs25.prn
P 13400 x 20	0.01325	20	0.0563	0.0081	0.1617	0.1485	qs27.prn
P 130500 x 20	0.1292	20	0.2740	0.0072	0.1879	0.0587	qs29.prn
P 56200 x 20	0.0557	20	0.1183	0.007	0.1135	0.0578	qs211.prn
P 23700 x 20	0.0234	20	0.0497	0.0069	0.0827	0.0593	qs213.prn

[*] For example P 400 x 200 = nominal half square wave pulse having a pulse width of 400 microseconds and a peak amplitude of 200 volts DC. The actual rising and falling slopes are limited to 1 volt/microsecond.

Table 3.2: Static Pressure Sensor Data for RCS Pulse Width Resolution Tests

Nominal Pulse Description*	Drive Pulse Width (sec.)	Peak Drive Ampl. (volts)	Active Energy Xfer (joules)	Response Rise Lag (sec.)	Response Pulse Width (sec.)	Response Fall Lag (sec.)	Data File Name
P 132000 x 20	0.1314	20	0.3090	0.0107	0.1588	0.0274	qs31.prn
P 90500 x 20	0.0905	20	0.2116	0.011	0.1179	0.0274	qs33.prn
P 38300 x 20	0.0383	20	0.0898	0.0106	0.0898	0.0275	qs35.prn
P 21400 x 20	0.0214	20	0.0502	0.0108	0.047	0.0256	qs37.prn
P 13000 x 20	0.0130	20	0.0307	0.0109	0.0307	0.0177	qs39.prn
P 10200 x 20	0.0102	20	0.0239	0.0110	0.0208	0.0106	qs311.prn
P 9500 x 20	0.0095	20	0.0221	0.0110	0.022	0.0125	qs313.prn
P 3500 x 39	**0.0035**	39	**0.0296**	**0.0048**	**0.0073**	**0.0038**	**qs41.prn**
P 3750 x 39	0.00357	39	0.0293	0.0046	0.0097	0.00613	qs43.prn
P 4800 x 39	0.0048	39	0.0409	0.0045	0.0346	0.0298	qs45.prn
P 6300 x 39	0.0063	39	0.0533	0.0046	0.0511	0.0448	qs47.prn
P 17000 x 39	0.017	39	0.1325	0.0046	0.0594	0.0424	qs49.prn
P 41200 x 39	0.0412	39	0.2832	0.0047	0.0831	0.0419	qs411.prn
P 3600 x 39	0.0036	39	0.0296	0.0047	0.0301	0.0265	qs415.prn

* For example P 400 x 200 = nominal half square wave pulse having a pulse width of 400 microseconds and a peak amplitude of 200 volts DC. The actual rising and falling slopes are limited to 1 volt/microsecond.

The bold data entry represents the minimum repeatable pulse width control exhibited by the RCS (solenoid valve) prototype during the laboratory tests reported herein.

Table 3.3: Acoustic Sensor Data for HSILS Pulse Width Resolution Tests

Nominal Pulse Description*	Drive Pulse Width (sec.)	Peak Drive Ampl. (volts)	Active Energy Xfer (joules)	Response Rise Lag (sec.)	Response Pulse Width (sec.)	Response Fall Lag (sec.)	Data File Name
P 300 x 200	0.00060	206	0.5729	0.00030	0.00088	0.00022	qh11.prn
P 400 x 200	0.00060	203	0.5563	0.00026	0.0015	0.00088	qh13.prn
P 350 x 200	0.00056	206	0.5729	0.00028	0.00182	0.00126	qh15.prn
P 400 x 200	0.00048	205	0.4977	0.00028	0.00214	0.00166	qh17.prn
P 250 x 200	0.00044	192	0.5081	0.00026	0.00212	0.00166	qh19.prn
P 200 x 200	0.00040	194	0.5081	0.00028	0.00206	0.00166	qh111.prn
P 175 x 150	0.00030	140	0.2646	0.00039	0.00124	0.00093	qh113.prn
P 100 x 100	0.00024	85	0.0975	0.00030	0.00034	0.00088	qh115.prn
P 400 x 200	0.00060	198	0.5293	0.00028	0.00148	0.00088	qh21.prn
P 500 x 200	0.00084	203	0.5563	0.00028	0.00308	0.00224	qh23.prn
P 600 x 200	0.00088	206	0.54	0.00028	0.000242	0.00154	qh25.prn
P 800 x 200	0.00114	202	0.5508	0.00028	0.00226	0.00112	qh27.prn
P 1000 x 200	0.00130	204	0.5618	0.00028	0.00208	0.00078	qh29.prn
P 2000 x 200	0.00222	205	0.5673	0.00028	0.00310	0.00088	qh211.prn
P 4000 x 200	0.00426	204	0.5618	0.00026	0.00562	0.00136	qh213.prn
P 6000 x 200	0.00624	204	0.5618	0.00026	0.00884	0.00260	qh215.prn
P 8000 x 200	0.00824	203	0.5563	0.00026	0.01496	0.00676	qh217.prn
P 10000 x 200	0.01026	206	0.5729	0.00028	0.02046	0.0102	qh219.prn
P 150 x 150	0.00032	139	0.2608	0.00042	0.00125	0.00093	qh31.prn
P 150 x 150	0.00033	140	0.2646	0.00037	0.00103	0.0007	qh33.prn
P 75 x 75	0.00025	46	0.0286	0.00039	0.00107	0.00082	qh37.prn
P 125 x 125	0.00028	107	0.1546	0.00039	0.00101	0.00070	qh39.prn
P 70 x 70	0.00024	45	0.0273	0.00039	0.00107	0.00093	qh311.prn
P 100 x 100	0.00033	86	0.0998	0.00030	0.00103	0.00070	qh313.prn
P 70 x 70	0.00025	44	0.0261	0.00025	0.00029	0.000040	qh315.prn
P 70 x 70	0.00025	43	0.0249	0.00022	0.00041	0.00013	qh317.prn
P 150 x 150	0.00032	140	0.260	0.00032	0.00103	0.00071	qh41.prn
P 100 x 100	0.00030	89	0.1066	0.00038	0.00102	0.00072	qh45.prn

Nominal Pulse Description*	Drive Pulse Width (sec.)	Peak Drive Ampl. (volts)	Active Energy Xfer (joules)	Response Rise Lag (sec.)	Response Pulse Width (sec.)	Response Fall Lag (sec.)	Data File Name
P 70 x 70	0.00023	46	0.0286	0.00039	0.00074	0.00051	qh47.prn
P 120 x 120	0.00033	107	0.1535	0.00038	0.00125	0.00092	qh49.prn
P 70 x 70	0.00023	45	0.0273	0.00039	0.00124	0.00101	qh411.prn
P 90 x 90	0.00027	86	0.0992	0.00040	0.00126	0.00099	qh413.prn
P 50 x 50	0.00025	44	0.0259	0.00025	0.00029	0.000040	qh415.prn
P 50 x 50	0.00021	44	0.0247	0.00023	0.00030	0.000090	qh417.prn
P 150 x 150	**0.00030**	**141**	**0.2651**	**0.00037**	**0.00098**	**0.00068**	**qh43.prn**

* For example P 400 x 200 = nominal half square wave pulse having a pulse width of 400 microseconds and a peak amplitude of 200 volts DC. The actual rising and falling slopes are limited to 1 volt/microsecond.

The bold data entry at the bottom of the table represents the minimum repeatable pulse width control exhibited by the HSILS prototype during the laboratory tests reported herein.

Table 3.4: Static Pressure Sensor Data for RCS (Solenoid) Amplitude Tests

Test No.	Drive Pulse Ampl. (volts)	Thrust (newtons)	Test No.	Drive Pulse Ampl. (volts)	Thrust (newtons)	Test No.	Drive Pulse Ampl. (volts)	Thrust (newtons)
4	3	0	9	8	0	17	16	0
2	1	0	14	9	0	18	16.9	0
3	2	0	11	10	0	18	17.2	1.689818
4	3	0	12	11	0	20	18.3	1.689818
5	4	0	13	12	0	21	22	1.624194
6	5	0	14	13	0	22	25.2	1.624194
7	6	0	15	14	0	23	31.1	1.657006
8	7	0	16	15	0	24	33.6	1.706224
							36.4	1.55857

Table 3.5: Static Pressure Sensor Data for HSILS Amplitude Tests

Test No.	Drive Pulse Ampl. (volts)	Thrust (newtons)	Test No.	Drive Pulse Ampl. (volts)	Thrust (newtons)	Test No.	Drive Pulse Ampl. (volts)	Thrust (newtons)
1	100	0.234278	19	160	0.457448	37	200	0.501203
2	140	0.032664	20	160	0.837116	48	200	0.654599
3	140	0.021615	21	160	0.577959	39	200	0.484715
4	100	0.225894	22	170	0.50396	40	200	1.197433
5	140	0.191901	23	170	0.601764	41	200	0.570109
6	110	0.131839	24	170	0.379311	42	200	0.570404
7	120	0.284149	25	180	0.668561	44	210	0.659111
8	120	0.301378	26	180	0.498988	44	210	1.450619
9	120	0.192032	27	180	0.745251	45	210	0.674992
10	130	0.247732	28	180	0.933034	46	210	0.980012
11	130	0.365854	29	180	0.845705	47	210	1.534371
12	140	0.430805	30	180	1.362682	48	220	1.146779
13	140	0.462042	31	190	0.390528	49	220	1.263262
14	140	0.332222	32	190	1.176556	50	220	1.091573
15	150	0.408714	33	190	1.242919	51	220	0.885104
16	150	0.494173	34	200	1.292793	52	220	1.073855
17	160	0.31068	35	200	1.254321	53	220	1.14842
18	160	0.776529	36	200	1.3238	54	220	1.015285
						55	220	1.422072

4.0 Discussion

4.1 Minimum Repeatable Pulse Width Parameters

A reasonable performance comparison between the HSILS and RCS systems is shown in Figs. 4.1 and 4.2 for the minimum repeatable pulse widths measured during the laboratory prototype tests described in Chapter 3. Table 4.1 presents this comparison data in tabular form.

Table 4.1: Comparison of RCS and HSILS Response Parameters

Parameter	RCS (solenoid)	HSILS (piezoelectric)	RCS/HSILS
Drive Pulse Width (seconds)	0.0035	0.0003	11.67
Response Rise Lag (seconds)	0.0048	0.00037	12.97
Response Pulse Width (seconds)	0.0073	0.00098	7.45
Response Fall Lag (seconds)	0.002	0.00068	2.94

All of the reponse variables for the RCS actuator presented in Table 4.1 compare favorably with the data presented by Hallauer and Smith (**Hallauer and Smith, 1990**) for a similar solenoid valve from a different manufacturer. The right-most column in Table 4.1 gives the ratios of the RCS response to those measured for the HSILS prototype. From these it can be seen that the HSILS actuator clearly out-performs the solenoid actuator on all counts. Pulse width control and rise lag for the HSILS are nearly an order of magnitude faster than for the solenoid actuator.

The lowest performance improvement for the HSILS unit is the fall lag which is only three times faster than for the solenoid and four times longer than the HSILS rise lag. This may be attributable to differences in the loading and unloading mechanisms used in the HSILS. Initial expansion of the piezoelectric stack occurs very rapidly (approximately 10 microseconds to full extension, based on the propagation speed of sound within the stack) and the stack is capable of generating extremely high levels of force (upwards of 8000 newtons) at low initial displacements. On the other hand, the return force needed to close the valve is provided by a compression spring with a constant force level of approximately 200 newtons. Given the high degree of damping in the system (provided by the high pressure valve core seals, as previously discussed in Chapter 2) this difference in force levels could account for the increased fall lag. Methods for improving this response might include low friction seals and a higher return spring force or, alternatively, a parallel piezoelectric stack used solely for closing the valve. Fig. 4.2 also provides a more detailed picture of the exhaust nozzle ringing (predominant frequency = 4 kHz) and decay following termination of thrust that was described in Chapter 3.

4.2 Parameter Correlations

A number of parameter studies were conducted based on the data presented in Tables 3.1, 3.2, and 3.3. These included correlation plots of response pulse width versus drive pulse width; response pulse width versus drive pulse energy; response rise lag versus drive pulse amplitude; response fall lag versus drive pulse energy and response fall lag versus drive pulse width. These plots are presented in Figs. 4.3 and 4.4 for the RCS and HSILS actuators, respectively.

4.2.1 RCS Results

The first correlation considered was that of response pulse width as a function of drive pulse width. The upper left hand plot in Fig. 4.3 indicates that for longer drive pulse widths there is a reasonable degree of correlation between the response and the drive pulse width. However, as observed in Chapter 3, this correlation appears to break down for drive pulse widths of less than 10ms.

The middle left plot in Fig. 4.3 shows the relationship between RCS response pulse width and drive pulse energy. For this comparison the active energy dissipation in the drive pulse was estimated as follows for the resistive-inductive load provided by the solenoid coil when driven by a constant current pulse power supply:

$$E_{RCS} = \sum_{i=1}^{n-1} \left(\frac{V(i+1) + V(i)}{2R} \right) (t(i+1) - t(i))$$

where:

n = *number of data scans (drive pulse record)*

$V(i)$ = *drive pulse voltage at scan* i

R = *solenoid resistance (ohms)*

$t(i)$ = *time at data scan* i *(seconds)*

As can be seen in Fig. 4.3 (middle, left graph) there appears to be a power-law relationship between solenoid response pulse width and the amount of energy provided to the coil. This is a somewhat intuitive observation since power must be continuously supplied to the coil to sustain the magnetic field that causes the valve to remain open once the initial opening threshold energy content has been reached. The same cannot be said for RCS fall lag (middle, right graph in Fig. 4.3) which shows no correlation to drive pulse energy nor to drive pulse width (lower, left graph Fig. 4.3). RCS rise lag, on the other hand, does appear to be correlated to the drive pulse amplitude (maximum drive voltage), with shorter rise lag (5ms vs 9ms) corresponding to the higher of the two voltage plateaus employed in these tests (39 and 21 volts, respectively).

4.2.2 HSILS

A similar set of correlation plots for the HSILS prototype is shown in Fig. 4.4. Here it can be seen (upper, left graph) that, unlike the RCS actuator, there is a good linear correlation (R^2=95.5%) between the HSILS response pulse width and the drive pulse width.
Unlike the RCS unit there is no correlation between HSILS response pulse width and drive pulse energy (middle, left graph Fig. 4.4). For the case of a capacitive load (for which a piezoelectric stack is a good emulator) in the face of a constant current pulse power supply the active energy dissipation is given by:

$$E_{HSILS} = \frac{1}{2} C V_{peak}^{2}$$

where:

C = *capacitance of the piezoelectric element*

V_{peak} = *maximum drive pulse voltage*

Once a plateau level has been reached in the drive pulse the piezoelectric stack will remain extended (and will therefore hold the valve open) with no expenditure of energy, as evidenced by the grouping of points at approximately 0.6 J in Fig. 4.4 (middle, left graph). Similar data could have been collected for any of the lower energy content pulses. The density of data at 0.6 J is a result of most of the pulse-width tests being conducted with a 200 V plateau amplitude drive pulse. A similar lack of correlation exists between response fall lag and dissipated energy, as shown in the middle, right graph in Fig. 4.4. Nor does there appear to be any correlation between response fall lag and drive pulse width.

The upper right hand graph in Fig. 4.4 indicates as well that there is no relationship between the drive pulse amplitude and the response rise lag. The data spread in rise lag for the HSILS unit are on the order of 0.2ms and do not appear to be a function of any drive pulse characteristics. The HSILS fall lag, on the other hand (middle right hand and lower left graphs, Fig. 4.4), shows local data spread of as much as 1.5ms, particularly for drive pulse widths in excess of 0.5ms. These may be related to the valve momentarily sticking in the open position following termination of the drive pulse. This would most likely be a result of differences in kinetic and static friction at the location where the valve core rod penetrates the high pressure chamber inboard of the exhaust nozzle throat. Along this section are several o-rings, backup rings, and a guide tube for the valve core which exhibit a high level of damping. As previously mentioned, it is more likely that fall lag and fall lag variances for the HSILS will be greater than rise lag (and its associated variances), due to the fact that the return spring used in the present HSILS prototype is incapable of developing the instantaneous high initial loads that the piezoelectric stack can upon initial valve opening. During initial cracking of the valve, as with closing from a stable open position, static friction at the high pressure seals will be greater than kinetic friction. The longer it takes to overcome the transition from static to kinetic friction the greater will be the expected lag and the level of variance. Since during valve opening the stack is able to adaptively increase the opening force, it can proceed through this friction transition faster, leading to lower variance in rise lag. Further research towards improving pulse width resolution for HSILS-type technology should thus be focused on techniques for reducing response fall-lag.

4.3 Amplitude Control using HSILS

As was shown in Fig. 3.14 there exists a linear relationship between output thrust and drive voltage for the HSILS prototype. The actual thrust levels involved were low, on the order of 1-2 N for an accumulator pressure of 5.5 MPa.

Supplementary testing was done at operating pressures as high as 41 MPa with a commensurate improvement in thrust amplitude (i.e. in the range of 7-15 N). At these higher thrust levels there were occasional instances where the valve did not close perfectly, as evidenced by a residual gas leak. This occurred more frequently for the initial valve seat configuration in which the expansion nozzle and valve seat were integrally machined from a rod of 304 stainless steel. The valve core rod was machined from VascoMax 300 maraging steel (yield strength 1884 MPa, hardness 15N-86). After several thousand test firings it was noted that the 304 stainless steel at the valve seat developed a localized spall due to impact fatigue (with subsequent detectable blow-by upon closing of the valve). Ultimately, the valve-seat/nozzle-throat segment of the expansion nozzle was re-designed as a replaceable unit. Two new materials were used for the replaceable valve seat: VascoMax 500 (yield strength 2760 MPa, hardness 92-HR15N [approx. HRC-65]) and Ferrotic alloy (45% titanium carbide, balance maraging steel binder; yield strength 3000 MPa MPa, hardness 2470-HK). Of these the Ferrotic seat performed best and there was no subsequent marring of the valve seat surface. However, after several hundred further firings it was discovered that the blow-by problem had reappeared. This time a small striated patch (approx. 0.05 mm^2) appeared on the valve core rod sealing surface at the valve tip. While the problem was resolvable in the laboratory by simply replacing the valve core rod, further research is waranted into the design of this detail of the HSILS, particularly for higher operating pressures.

Assuming that existing mechanical design problems can be resolved it may be desirable, particularly for non-spacecraft industrial active control applications, to substantially increase the maximum thrust level. Table 4.2 lists design parameters for scaled HSILS units. The gas accumulator operating pressure is presumed to be 41 MPa and the variables listed in Table 4.1 are defined as:

D_{tank} = accumulator tank diameter (mm)
d_{th} = nozzle throat diameter (mm)
h_{stack} = piezoelectric stack height (mm)
F_{spring} = return spring (closing force) level (N)
F_{stack} = maximum force level expected from piezoelectric stack (N)

The maximum impulse bit is defined as the product of peak thrust times the longest achievable pulse duration (assumed to be 0.1 second) for an approximate half square wave drive signal. The minimum impulse bit for the current HSILS prototype, for comparison, is approximately 0.0002 newton-seconds (corresponding to a 0.98ms x 0.2 newton pulse at an accumulator pressure of 5.5 MPa). All of the parameters listed in Table 4.2 are achievable with present technology. The larger piezo stacks would have to be custom fabricated. The use of longer actuators to achieve the necessary valve displacements for the larger thrusters should not incurr degradation in the speed of the valve, provided the lamination thickness and electroding procedures remain the same.

Table 4.2: Scaling of HSILS Technology to Industrial Control Levels

Thrust (newtons)	D_{tank} (mm)	d_{th} (mm)	h_{stack} (mm)	F_{spring} (newtons)	F_{stack} (newtons)	Max. Impulse Bit (n-sec)
15	102	1	84	209	2510	1.5
150	230	3.2	264	2042	24510	15
1500	500	10	835	20394	244750	150
15000	1000	32	2642	204255	2451950	1500

4.4 HSILS Amplitude Control Rate (ACR)

The development of "intelligent" control pulses requires both the ability to generate short packets of thrust (as measured by pulse width resolution discussed above) as well as the ability to rapidly change the thrust amplitude. A measure of this ability to change thrust is the amplitude control rate which can be estimated by the peak thrust generated by a response pulse divided by the rise time to reach that thrust, in units of newtons per second.

As previously noted, thrust amplitude measurements for the HSILS unit were based on the average of a band of readings taken while the HSILS unit was subjected to a steady state voltage. Rise time data were thus not available for those tests in which actual thrust was measured. However, rise times were available from the acoustic sensor pulse width resolution tests. From these we can infer a level of thrust by means of the thrust equation presented in Fig. 3.14. Table 4.3 presents data for several test firings of the HSILS which give an approximate indication of the amplitude control rate that was achievable with the HSILS laboratory prototype. There is a certain amount of imprecision with this method of estimating the ACR since the peak amplitudes in some of the acoustic data were truncated in the analog to digital conversion process. The ACR value of 1500-1600 N/s is relative to the present HSILS prototype when operated at 5.5 MPa accumulator pressure. Since the response times for the expansion mechanism are independent of accumulator pressure, an ACR of 12000 newtons/second would be anticipated when the HSILS was operated at its maximum accumulator pressure of 41 MPa.

Similar data for the RCS (solenoid) actuator indicated an average ACR value of approximately 500 newtons/second when operated at 5.5 MPa (twice the recommended service pressure). The ACR improvement ratio for the HSILS relative to the RCS system, when both are operated at their maximum recommended pressures is thus (12000/250) = 48; i.e. the HSILS has an amplitude change rate 48 times faster than for the solenoid actuator.

Table 4.3: HSILS Amplitude Control Rate Tests (5.5 MPa operating Pressure)

Test Name	Peak Voltage	Rise Time (RT) (seconds)	Calculated Peak Thrust (N)	Amplitude Control Rate (ACR) (N/s)
qh33.prn	139	.00028	0.40	1430
qh31.prn	139	.00023	0.40	1782
qh41.prn	139	.00023	0.40	1782
			Average ACR:	1664

4.5 Damping Effects

Original laboratory measurements of viscous damping (see Chapter 2) yielded an estimate of the HSILS internal damping ratio of 388% of critical. The laboratory tests reported in Fig. 3.12 confirm this high level of damping. For example, the limiting repeatable HSILS response shown in Fig. 4.2 indicates negligible amplitude oscillation upon reaching the full-open position and likewise negligible amplitude oscillation following valve closing. The vibrations which decay following closing of the valve are, as previously reported, acoustic resonant ringing of the exhaust nozzle. While the high pressure seals which isolate the valve core drive mechanism from the pressurized gas chamber are likely a strong factor in determining the limiting pulse width resolution (due to increased fall lag), they are both necessary to the isolation of the pressurized portion of the operational device as well as to eliminate bounce as the valve closes (there are, in fact, two examples of situations in which a single "bounce" did occur; these are shown in Fig. 3.12c). Any future enhancements to the device will necessarily have to deal with the tradeoffs inherent in eliminating valve core bounce while minimizing response lag times.

4.6 Mechanical Design Observations

Future use of the HSILS unit on spacecraft, for the purposes of rapid attitude acquisition and precision pointing control, will require significant improvements to the laboratory prototype described in this report.

As was mentioned earlier, no effort was made to optimize the mass or volume of the prototype. It is the author's opinion that a 25 N thruster employing the HSILS concept, with full onboard intelligence, power and gas storage could be constructed with a mass of under 2 kg and a volume of approximately 0.015 m^3, using composite materials and hybrid IC circuitry. This device is energy efficient and could be scaled to meet higher thrust requirements while not sacrificing control speed.

For long duration use the nozzle seat problem must be addressed and consideration given to possible inclusion of tough elastomers for the valve core tip. It was confirmed, through visual observation, that the core rod had developed a slight permanent curvature during the course of testing, most likely from slightly eccentric axial impact loads sustained during valve closing. This may have accounted for the fact that the previously described metal fatigue on the valve core tip only occurred on one side in a localized patch. Elimination of valve core rod warp would require a stiffer element. In order to minimize mass a hollow composite element should be considered.

Assuming that the above mentioned valve core seating problems can be met, it is the author's opinion that operating pressures as high as 40 MPa should not constitute a problem. In its present configuration the HSILS unit could be operated not only as a cold gas unit but also with standard monopropellants (e.g. MMH).

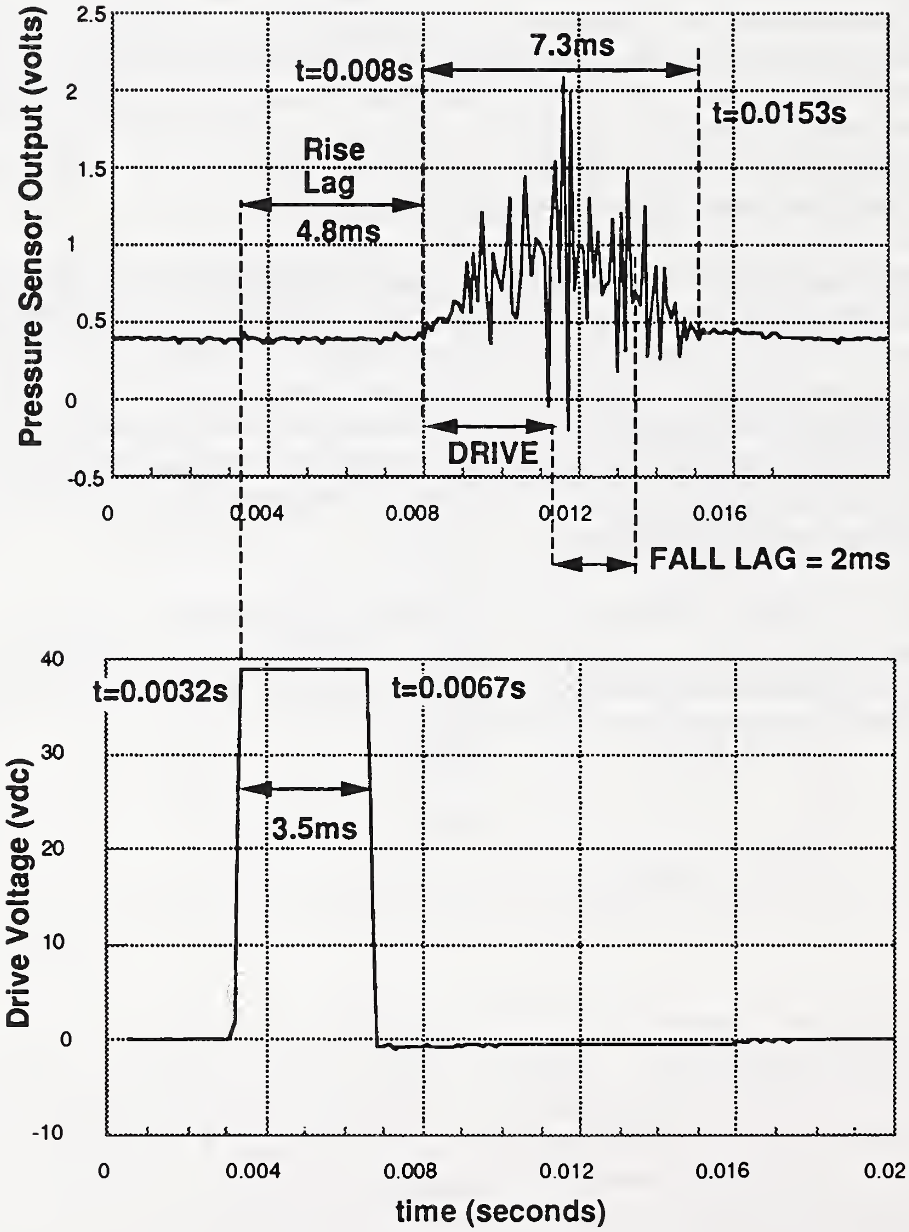


Fig. 4.1: Limiting pulse width response for solenoid-based RCS actuator and the voltage-time drive pulse used to produce the actuator response.

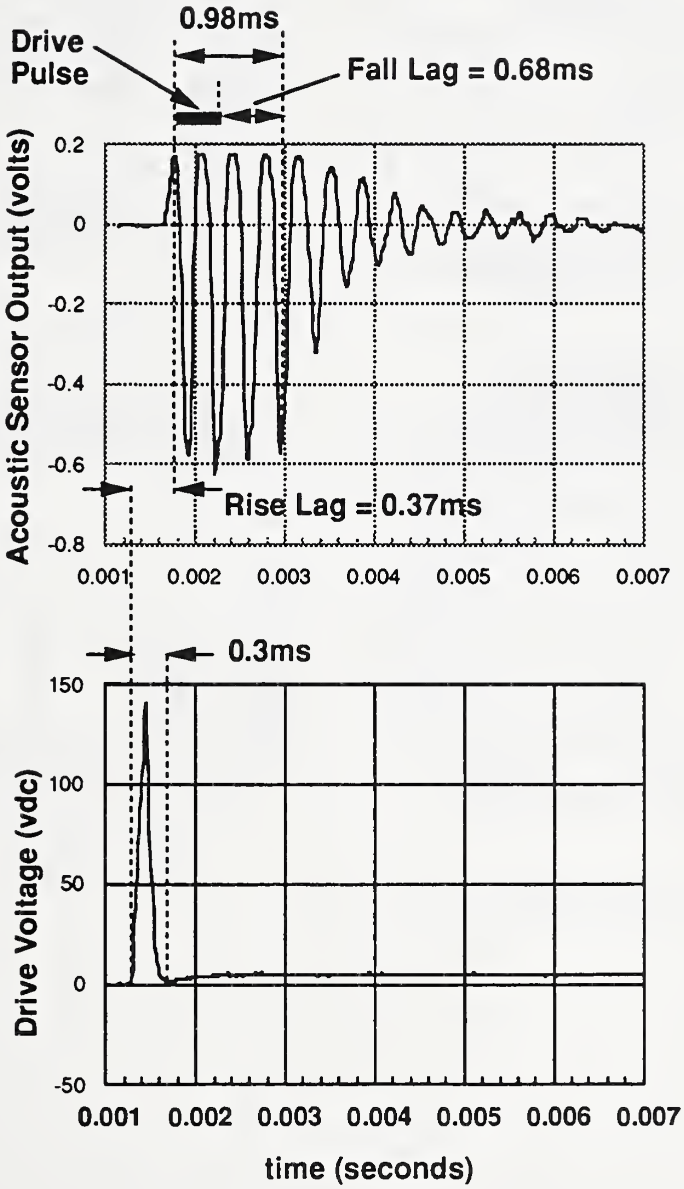


Fig. 4.2: Limiting pulse width response for the HSILS prototype and the voltage-time drive pulse used to produce the actuator response.

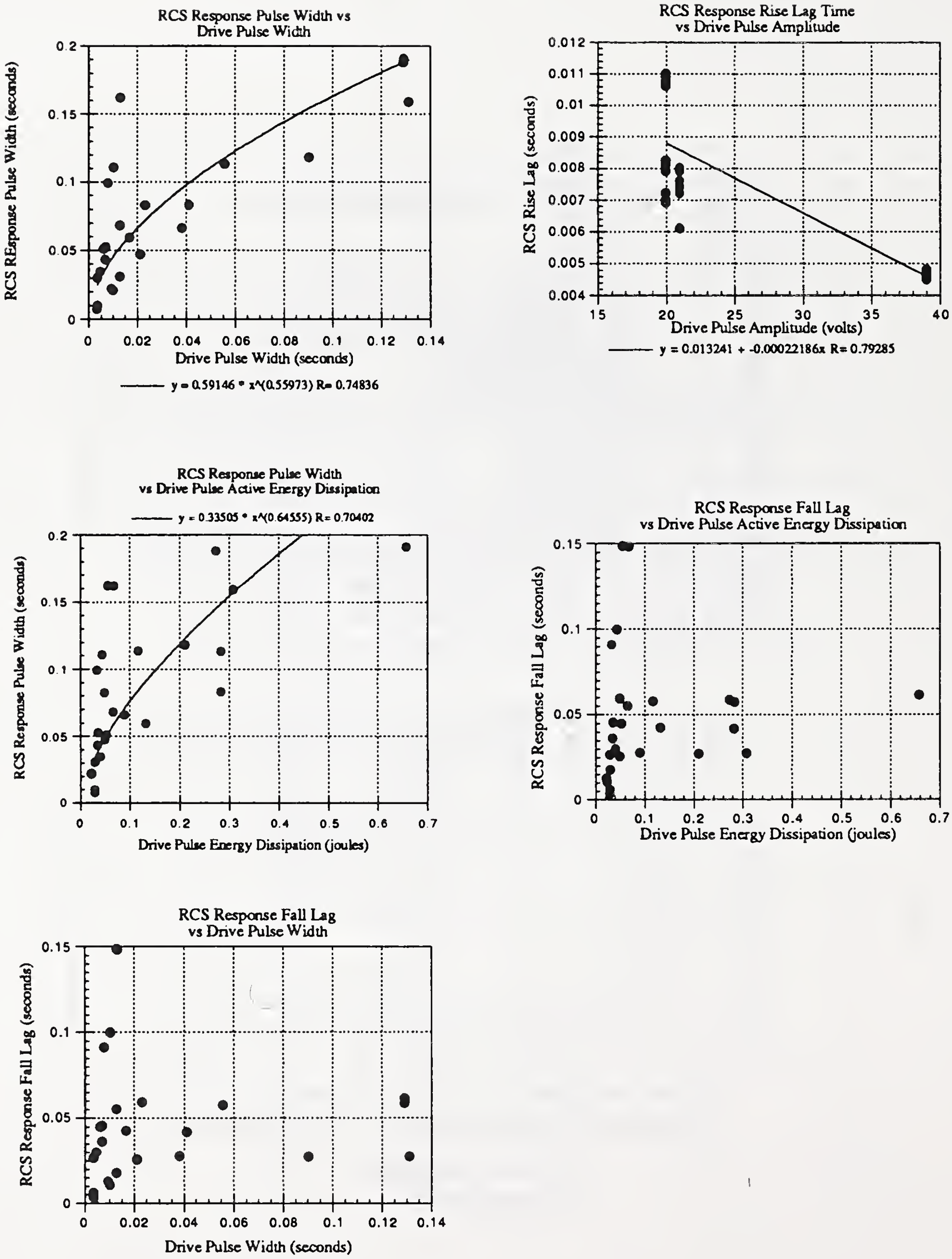


Fig. 4.3: Typical solenoid-based RCS actuator response characteristics. Parameters are defined in Chapter 2 (see Fig. 2.2); drive pulse energy is as defined above in Chapter 4.

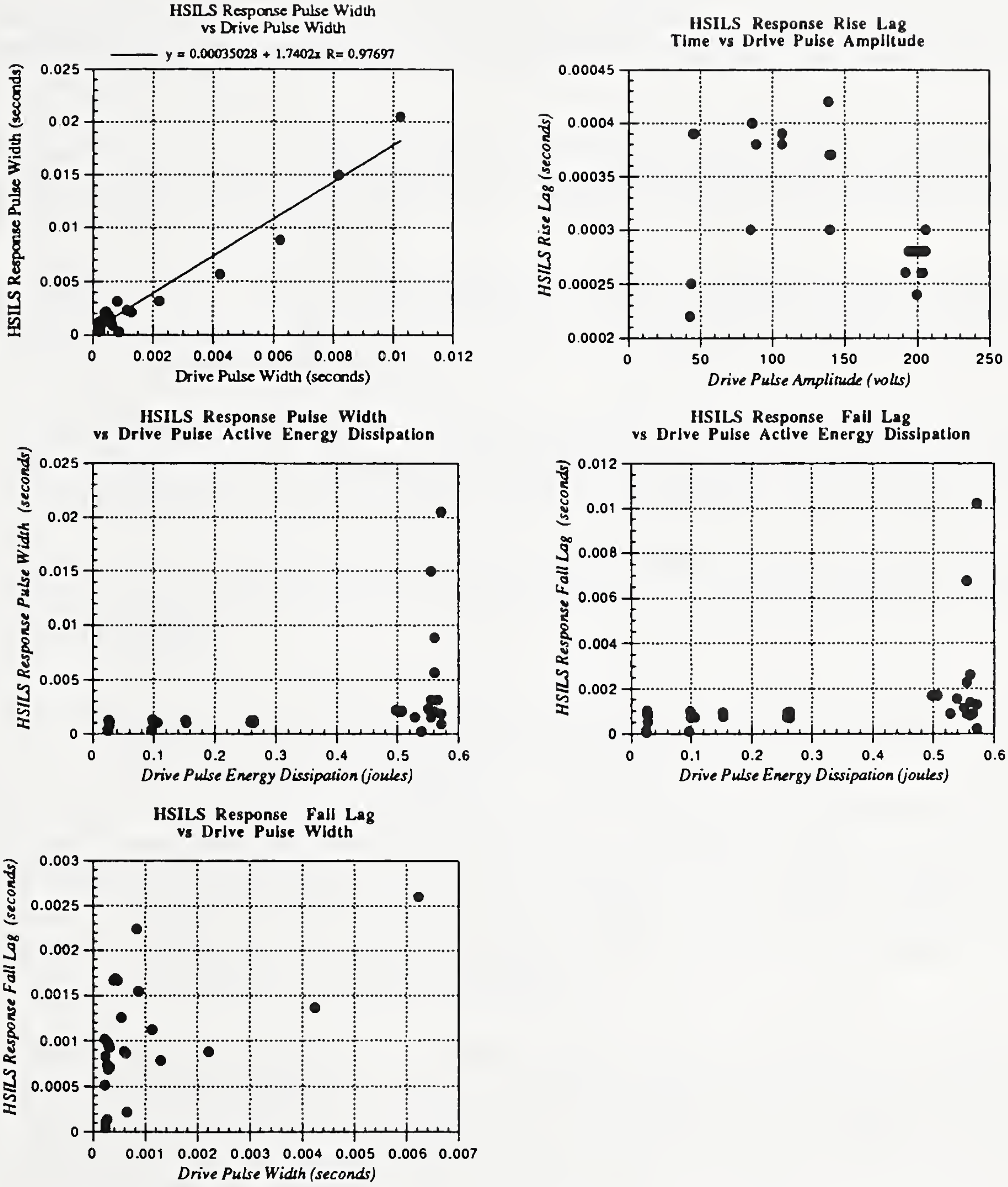


Fig. 4.4: Typical HSILS response characteristics. Parameters are defined in Chapter 2 (see Fig. 2.2); drive pulse energy is as defined above in Chapter 4.

5.0 Conclusions

A working laboratory prototype High Speed Intelligent Loading System (HSILS) was developed and tested at NIST in side-by-side comparisons with traditional solenoid-based reaction control technology. The HSILS concept employed a mechanically amplified piezoelectric stack as the active valve element in a cold gas thruster as opposed to an electro-magnetically operated valve core. The HSILS unit exhibited the following levels of performance improvement:

- Minimum rise lag (time from initiation of drive pulse to the initiation of measureable output thrust) of 0.37 ms. This value is 13 times faster than for equivalent solenoid-based reaction control systems.

- Response pulse width resolution (minimum base to base elapsed time for a repeatable pulse) of 0.98 ms. This value is approximately 8 times faster than for an equivalent solenoid-based system.

- Minimum response fall lag (time from initiation of measurable output thrust to the point at which thrust termination begins minus the drive pulse width) of 0.68ms. This value is approximately 3 times faster than for an equivalent solenoid-based system.

- Amplitude control rate (thrust rate of change) of 12000 N/s when operated at rated pressure of 41 MPa. This value is approximately 50 times faster than for solenoid-based technology.

The HSILS unit was shown to exhibit a linear correlation between drive voltage and output thrust. This permits active thrust control within the regime of the performance statistics listed above. More specifically, this permits sub-millisecond "intelligent" (of arbitrary analytical shape) uni-directional control forces to be applied to a structural system. Solenoid-based technology was shown to have binary thrust behavior (either off or full-on). The HSILS laboratory prototype was shown to be capable of delivering a peak thrust of approximately 15 N at an accumulator pressure of 41 MPa. Scaling of the HSILS unit to achieve larger maximum thrust appears feasible without significant penalties to the performance characteristics determined in this study.

The HSILS system was controlled by a dedicated onboard microcontroller and pulse power system which, after being charged, could be operated in a stand-alone mode. Compressed nitrogen accumulators were also an integral part of the laboratory prototype so that the entire package could be attached to any structural system to serve either as a remote, independent active control or system identification node.

Several areas for future improvement and research were identified. Fall lag times were disproportionately high with respect to the observed rise lag times. This was traced to the differences in the opening and closing mechanisms used in the present HSILS design. Currently a fast-acting piezoelectric stack is used for valve opening and a preloaded compression spring for valve closing. The former is capable of dynamically increasing its force level, particularly at small displacements. The resistance to valve opening and closing arises from high pressure seals which separate the pressurized section (the valve seat area) from the unpressurized core actuator mechanism. Future improvements may involve twin piezoelectric stacks operated in opposition; the use of higher force levels in both the piezoelectric stack and the return spring; or the development of a low damping high pressure seal.

Additional problems were noted at the valve seat where high strength, high hardness valve elements had to mate precisely, and repeatedly, at closing velocities of 4-5 m/second. Reasonable sealing performance was obtained by using a 2000 MPa maraging steel valve core rod and an EDM milled and polished valve seat fabricated from a ferritic titanium carbide alloy. Visible warping of the valve core rod, which was quite slender, was observed following several thousand test firings, as was minor galling on the valve core rod tip. Future improvements may involve the design of a tubular composite valve core rod with hardened elastomeric or carbon coatings on the mating surfaces.

6.0 References:

Abe, S., Sakakibara, Y., Tomita, M, and Shinoda, K. (1985), "Piezoelectric Actuator and a Piezoelectric Pump Injector Incorporating the Same," U.S. Patent 4,555,059. U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Babitzka, R., Beck, W., and Linder, E. (1985), "Electrically Controllable Valve," U.S. Patent 4,494,727. U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Bailey, T. and Hubbard, J.E. (1985), "Distributed Piezoelectric-Polymer Active Vibration Control of a Cantilever Beam," Journal of Guidance, Control, and Dynamics, Vol. 8, No. 5, Sept.- Oct., 1985, American Institute of Aeronautics and Astronautics, Washington, DC.

Behmer, A., and Hartmut, H. (1986), "Typenraddrucker mit den Druck auslosenden Elementen," Deutsches Patentamt DE-3435024A1. Bundesrepublik Deutschland, 1986.

Belvin, W.K, and Edighoffer, H.H. (1986), "Dynamic Analysis and Experiment Methods for a Generic Space Station Model," Proceedings of the 27th Structures, Structural Dynamics and Materials Conference, May 19-21, 1986, San Antonio, Texas," Part 2, pp 10-18, AIAA paper #86-0838, American Institute of Aeronautics and Astronautics, Washington, DC.

Cannon, R.H. Jr., and Schmitz, E. (1984), "Initial Experiments on the End-Point Control of a Flexible One-Link Robot," The International Journal of Robotics Research, Vol. 3, No. 3, Fall 1984, pp 62-75, Massachusetts Institute of Technology, Cambridge, MA.

Carasso, A.S. (1991), "Probe Waveforms and the Reconstruction of Structural Dynamic Green's Functions," AIAA Journal, Vol. 29, No. 1, January 1991, pp 114-118, American Institute of Aeronautics and Astronautics, Washington, DC.

Carasso, A.S., and Simiu, E. (1988), "Estimation of Dynamic Green's Functions for Large Space Structures by Pulse Probing and Deconvolution," Computational Mechanics '88, S.N. Atluri et.al. eds., Vol 2, Springer-Verlag, New York, 1988.

Carasso, A.S., and Simiu, E. (1988), "Dynamic Characterization of Structures by Pulse Probing and Deconvolution,"Proceedings of the AIAA/ASME/ASCE/AHS 29th Structures, Structural Dynamics, and Materials Conference, Williamsburg, VA, April 1988, AIAA paper #88-2230, American Institute of Aeronautics and Astronautics, Washington, DC. American Institute of Aeronautics and Astronautics, Washington, DC.

Carasso, A.S., and Simiu, E. (1989), "Identification of Dynamic Green's Functions in Structural Networks," AIAA Journal, Vol 27, April 1989, pp 492-499, American Institute of Aeronautics and Astronautics, Washington, DC.

Crawley, E.F. and DeLuis, J. (1985), "Use of Piezo-Ceramics as Distributed Actuators in Large Space Structures," Proceedings of the AIAA/ASME/ASCE/AHS 26th Structures, Structural Dynamics, and Materials Conference, Orlando, Fl, April 1985, AIAA paper #85-0626, American Institute of Aeronautics and Astronautics, Washington, DC.

Glett, B., and Murphy, M. (1987), "Piezoelectrically Driven Fast Response High-Torque Clutch Unit," U.S. Patent 4,636,679. U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Hanugud, S., Obal, M.W., Meyyappa, M. (1985), "Electronic Damping Techniques and Active Vibration Control," Proceedings of the AIAA/ASME/ASCE/AHS 26th Structures, Structural Dynamics, and Materials Conference, Orlando, Fl, April 1985, AIAA paper #85-0752, American Institute of Aeronautics and Astronautics, Washington, DC.

Hallauer, W.L., and Smith, J.C. (1990) "Air-Jet Thrusters for Active Vibration Control: Design and Calibration," Journal of Experimental Techniques, November/December 1990, pp 47-50.

Hiratsuka, H., Kawasaki, K, and Miyo, Y. (1991), "Flow-Control Type Piezoelectric Element Valve,",U.S. Patent 5,029,610 U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Janner, K., and Gregorius, K. (1984), "Slit Nozzle and Fast-Acting Shutoff Valve," U.S. Patent 4,431,136. U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Kattchee, N. (1967), "Piezoelectric Injection System for Vernier Impulse Thrusters," Report No. AFRPL-TR-67-201, June 1967, Physics International Co., San Leandro, CA, 123pp.

Kippenham, D. (1962), "Electrostriction Valve," U.S. Patent 3,055,631, U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Knappe, L.F. (1972), "Betatigungsvorrichtung fur mit hoher Geschwindigkeit hin- und her zu bewegende Maschinenteile," Deutsches Patentamt DE-2134655. Bundesrepublik Deutschland, 1972.

Kushida, T. (1987), "High Speed Electromagnetic Valve," U.S. Patent 4,678,000. U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Mitsuhashi, S., Wakamatsu, K., Aihara, Y., and Okihara, N., 1985, "The Development of a Piezoelectric Relay with a Multilayer Piezoelectrtic Ceramic Actuator," Proceedings, 33rd Annual National Relay Conference, Oklahoma State University, Stillwater, Oklahoma, April 23-24, 1985. pp 7.1 - 7.11.

Moloney, P. (1986), "Valve Operating Mechanism for Internal Combustion and Like-Valved Engines,",U.S. Patent 4,593,658. U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Moog (1990), "760 Series Electrohydraulic Servovalve Service Manual," Mood, Inc., Industrial Division, East Aurora, NY 14052

Moritugu, M., Kawai, H., and Tokura, N. (1987), "Power Supply for Piezoelectric-Element Driving Device,",U.S. Patent 4,644,212 U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Piezoelectric Ceramics, J.V. Randeraat and R.E. Setterington, eds., Mullard Limited, Mullard House, Torrington Place, London WC1E 7HD, 2nd Ed. 1974.

Simiu, E., and Cook, G.R. (1989), "Impulse Response Functions for Elastic Structures with Rigid-Body Degrees of Freedom," Journal of Sound and Vibration, 135(2), pp 275-288,

Stone, W.C. (1990), "High Speed, Amplitude Variable Thrust Control," Patent Pending, U.S. Patent and Trademark Office, Crystal City, VA.; Ser.No. 07/873,020

Sutton, G.P. (1986), "Rocket Propulsion Elements -- An Introduction to the Engineering of Rockets," John Wiley & Sons, New York, ISBN 0-471-80027-9.

Takahashi, S., 1985, "Development of New Piezoelectric Ceramic Actuators," New Materials & New Processes, Vol. 3 (1985), pp 63-68.

Tomono, A. (1986), "Piezoelectric Actuator Using Bimorph Element," U.S. Patent 4,625,137. U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

von Flotow, A.H. (1984), "Disturbance Propagation in Structural Networks; Control of Large Space Structures," Ph.D. Dissertation, Stanford University, June 1984. 144pp.

Weber,K., and von Blanckenhagen, P. (1987), "Pulse Valve,",U.S. Patent 4,669,660. U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Weiger, G, Dorrie, D, and Schwerdt, P. (1986), "Actuating Device for Two Mutually Dependently Actuable Valves," U.S. Patent 4,617,969. U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Yamauchi, T., Nogi, T., and Ohyama, Y. (1991), "Fuel Injection Valve and Fuel Supply System Equipped Therewith for Internal Combustion Engines," U.S. Patent 5,029,610 U.S. Commissionar of Patents and Trademarks, Crystal City, VA.

Zimmerman, D.C., Horner, G.C., and Inman, D.J. (1986), "Microprocessor Controlled Force Actuator," Proceedings of the 27th Structures, Structural Dynamics and Materials Conference, May 19-21, 1986, San Antonio, Texas," Part 1, pp 243-251, AIAA paper #86-0916, American Institute of Aeronautics and Astronautics, Washington, DC.

Appendix A
Program: THRUSTER

Parametric Design of HSILS Components

```
C***************************************************************
C
      PROGRAM THRUSTER
C
C      PURPOSE: TO CALCULATE THE TIME TO ZERO INTERNAL PRESSURE
C              FOR A COLD GAS THRUSTER CONNECTED TO A PRESSURE
C              VESSEL AND NOZZLE OF KNOWN PROPERTIES AND (FOR THIS
C              EXPERIMENTAL PROGRAM ONLY) VARIABLE THROAT AREA.
C              PROGRAM ALSO PROVIDES TIME STEP OUTPUT FOR PRESSURE
C              AND ESTIMATE OF IDEAL THRUST.
C
C              PROGRAM ALSO DETERMINES REQUIRED DISPLACEMENT OF
C              VALVE CORE & OUTPUT VOLTAGE FROM A MICROPROCESSOR
C              DAC TO ACHIEVE AN INVERSE GAUSSIAN FORCE PULSE
C              INPUT BY THE USER.  PROGRAM PROMPTS USER FOR THE
C              THE KEY VARIABLES NECESSARY TO DEFINE THE INVERSE
C              GAUSSIAN PULSE.
C
C      BACKGROUND:
C
C      THE PROGRAM PROMPTS FOR SUPPLY VESSEL VOLUME AND STARTING INTERNAL
C      PRESSURE, THE THROAT DIAMETER OF THE NOZZLE AND THE DESIRED TIMESTEP
C      FOR NUMERICAL INTEGRATION.  IT THEN CALCULATES THE TIME TO ZERO
C      PRESSURE INSIDE THE PRESSURE VESSEL.  OPTIONAL OUTPUT AT EACH TIME
C      INCREMENT ARE M,P,RHO,VEL,DMDT, AND T.  THESE WILL PROVIDE THE
C      NECESSARY INFORMATION TO OBTAIN OUTPUT THRUST AT TIME "T".
C
C      1. THE PROGRAM USES A REPRESENTATIVE INVERSE GAUSSIAN PULSE TO
C        DRIVE THE PROGRAM AT TIME "T" BY CHANGING NOZZLE DIAMETER TO PROVIDE
C         THE REQUIRED FORCE LEVELS, TAKING INTO ACCOUNT THE FACT THAT
C         INTERNAL PRESSURE IS DROPPING.  PLOTS OF NOZZLE AREA VS TIME AND
C         FORCE VS TIME ARE AVAILABLE.   THE NOZZLE DISPLACEMENT VS TIME
C         DATA CAN BE USED TO BACK-CALCULATE THE MICROPROCESSOR DAC
C         OUTPUT DRIVE VOLTAGE VERSUS TIME.  THIS CAN THEN BE DOWNLINKED
C         TO THE ONBOARD MICROPROCESSOR TO SERVE AS AS A HARD CODE LOOKUP
C         TABLE DATA TO DRIVE THE MICROPROCESSOR DAC VOLTAGE IN THE ACTUAL
C         EXPERIMENTAL THRUSTER.
C
C      2. SEE APPENDIX B, PROGRAM HDYN FOR TRANSIENT MODELING OF HSILS.
C
C      3. PRESENT CALCULATIONS ARE BASED ON A MODIFICATION OF BERNOULLI
C         THEORY FOR NON-STEADY STATE FLOW (INCLUDES SONIC FLOW IN A
C         COMPRESSIBLE MEDIUM)
C
C      VERSION:    10-25-88       ORIGINAL
C                  11-03-88      ADDED PLOTTING
C                   1-23-89      ADDITIONAL PLOT CONTROL
C                   5-16-89      GENERAL DEBUGGING: SPLIT
C                              PROGRAM INTO INDEPENDENT MODULES
C      WRITTEN BY: BILL STONE
C
C        SUBROUTINES CALLED:
C
C        USER_INPUT            : INPUT DATA FOR INTERACTIVE RUN
```

```
C       PRESSURE_CALCS    : GAS RELATED CALCULATIONS
C       PIEZO_CALCS       : PIEZOELECTRIC STACK PARAMETERS
C       PRESSURE_PLOT     : MENU FOR GRAPHICS OUTPUT
C       SETUP_GRAPHICS    : INITIALIZES RASTER TECH TUBE
C       PLOTPREP          : PROMPTS FOR GRAPH PLOTTING DETAILS
C       POINTPLOT         : PLOTS ACTUAL DATA POINTS ON THE RASTER TECH
C       SORT              : SORTS PLOT DATA IN ASCENDING ORDER, X
C
C       NOTE: THE FOLLOWING LINK COMMAND IS NECESSARY TO ASSEMBLE
C             THE VARIOUS PROGRAM MODULES:
C
C       $LINK/NOMAP/EXECUTABLE=THRUSTER-
C       THRUSTER_MAIN,USER_INPUT,PRESSURE_CALCS,PIEZO_CALCS,-
C       PRESSURE_PLOT,SETUP_GRAPHICS,LINEFIT,PLOTPREP,-
C       CHGCOLOR,POINTPLOT,LINEPLOT,SLIDE1,GRAB3,SORT,GDATA,ORDER,-
C       OUTPUT,MINV,MULTR,USER:[RASTER.ONELIB]ONELIB/LIB
C
C       OPERATIONAL NOTES:
C
C       1.  "THRUSTER" WAS WRITTEN FOR A VAX 11/750 COMPUTER
C           AND A RASTER TECHNOLGIES ONE/80 1280 X 1024 PIXEL
C           HIGH RESOLUTION COLOR RASTER DISPLAY.  THE GRAPHICS
C           ARE HARD-CODED TO THIS MACHINE SETUP, BUT TRANSFER
C           TO MORE RECENT GRAPHICS BASED WORKSTATIONS, SUCH
C           AS A SILICON GRAPHICS 4D/420, IS RELATIVELY STRAIGHT
C           FORWARD WITH REGARDS TO GRAPHICS CODE; COMPUTATIONAL
C           CODE IS 100% COMPATIBLE FORTRAN.
C
C***********************************************************************
C
      COMMON TMASS(5000),P(5000),RHO(5000),VEL(5000),DMDT(5000)
      COMMON T(5000),CMASS(5000),CRHO(5000),CPRES(5000)
      COMMON THRUST(5000),AT(5000),DISP(5000)
      COMMON FGAUSS(5000),DELTA(5000),FTHRUST(5000)
      COMMON DELTASTACK(5000),VDAC(5000),ATEFF(5000)
C
C      MAIN MENU
C
5666  TYPE *,' '
      TYPE *,'.......................................'
      TYPE *,' '
      TYPE *,' PROGRAM THRUSTER:'
      TYPE *,' '
      TYPE *,' A SIMULATOR FOR PARAMETRIC DESIGN OF THE NIST'
      TYPE *,' INTELLIGENT HIGH SPEED LOADING SYSTEM'
      TYPE *,' '
      TYPE *,'MAIN MENU: '
      TYPE *,' '
      TYPE *,' 1 = ENTER NEW DATA SET'
      TYPE *,' 2 = CHANGE TANK PRESSURE'
      TYPE *,' 3 = CHANGE TANK DIAMETER'
      TYPE *,' 4 = CHANGE TEMPERATURE'
      TYPE *,' 5 = CHANGE NOZZLE DIAMETER'
      TYPE *,' 6 = CHANGE INTEGRATION TIME STEP'
      TYPE *,' 7 = CHANGE MAX GAUSS FORCE'
      TYPE *,' 8 = CHANGE SIGMA (SHAPE CONSTANT)'
      TYPE *,' 9 = CHANGE PULSE DURATION'
      TYPE *,'10 = CHANGE DISPLACEMENT MAG FACTOR'
      TYPE *,'11 = PLOT RESULTS (RASTER TECH)'
      TYPE *,'12 = STOP'
      TYPE *,' '
      TYPE *,'..........................................'
      READ *,K
      IF(K.EQ.12) GOTO 130
      IF(K.GE.1.AND.K.LE.10)THEN
              CALL USER_INPUT(K,TMASS0,P0,DT,D,VOL,FAMP,
     1        TMAX,REALK,TOTP,WTOT,NPTS)
                ENDIF
      IF(K.GE.1.AND.K.LE.10) GOTO 5666
      IF(K.EQ.11) THEN
              CALL PRESSURE_PLOT(NPTS)
              ENDIF
      GOTO 5666
130    CONTINUE
      TYPE *,' '
      TYPE *,'...............................................'
      TYPE *,' '
      TYPE *,'                  NORMAL TERMINATION'
      TYPE *,' '
      TYPE *,'...............................................'
      TYPE *,' '
      CALL QUIT
      CALL EXIT
      STOP
      END
```

```
C************************************************************
C
      SUBROUTINE USER_INPUT(NFLAG,TMASS0,P0,DT,D,VOL,FAMP,
     1 TMAX,REALK,TOTP,WTOT,NPTS)
C
C       PURPOSE: PROMPT USER FOR INTERACTIVE INFORMATION NEEDED
C                TO SET UP THE PROBLEM BOUNDS FOR PROGRAM THRUSTER.
C
C       VERSION:    10-25-88       ORIGINAL
C                   5-16-89        GENERAL DEBUGGING: SPLIT
C                                  PROGRAM INTO INDEPENDENT MODULES
C       WRITTEN BY: STONE
C
C************************************************************************
C
      COMMON TMASS(5000),P(5000),RHO(5000),VEL(5000),DMDT(5000)
      COMMON T(5000),CMASS(5000),CRHO(5000),CPRES(5000)
      COMMON THRUST(5000),AT(5000),DISP(5000)
      COMMON FGAUSS(5000),DELTA(5000),FTHRUST(5000)
      COMMON DELTASTACK(5000),VDAC(5000),ATEFF(5000)
      REAL*8 MOLWT
C
C      JUMP BASED ON VALUE OF "NFLAG" PASSED FROM MAIN MENU
C      [NOTE THAT NFLAG=1 IS DEFAULT = PROMPT FOR ALL INPUT]
C
603    CONTINUE                          !LOOP BACK FROM VERIFICATION MENU
      IF(NFLAG.EQ.2) GOTO 802                  !CHANGE TANK PRESSURE
      IF(NFLAG.EQ.3) GOTO 803                  !CHANGE TANK DIAMETER
      IF(NFLAG.EQ.4) GOTO 804                  !CHANGE TEMPERATURE
      IF(NFLAG.EQ.5) GOTO 805                  !CHANGE NOZZLE DIAMETER
      IF(NFLAG.EQ.6) GOTO 806                  !CHANGE INTEGRATION TIME STEP
      IF(NFLAG.EQ.7) GOTO 807                  !CHANGE MAX GAUSS FORCE
      IF(NFLAG.EQ.8) GOTO 808                  !CHANGE SIGMA (SHAPE CONST)
      IF(NFLAG.EQ.9) GOTO 809                  !CHANGE PULSE DURATION
      IF(NFLAG.EQ.10)GOTO 810             !CHANGE DISP AMPLIFICATION
C
C      PROMPT FOR INITIAL VALUES
C
      TYPE *,' '
      TYPE *,'..................................................'
      TYPE *,' '
      TYPE *,'PROBLEM DEFINITION:  ENTER VALUES REQUESTED FOLLOWED'
      TYPE *,'                     BY A CARRIAGE RETURN'
      TYPE *,' '
      TYPE *,'..................................................'
      TYPE *,' '
C
C......................................................................
C
C      STARTING COLD GAS STORAGE VESSEL INTERNAL PRESSURE
C
802    CONTINUE
      TYPE *,' '
      TYPE *,'ENTER INTERNAL PRESSURE OF STORAGE SPHERE (PSI)'
      TYPE *,' '
      READ *,TOTP
      IF(NFLAG.EQ.2) GOTO 8021
C
C..................................................................
C
C     INTERNAL VOLUME OF STORAGE VESSEL
C
803   CONTINUE
      TYPE *,'...............................................'
      TYPE *,' '
      TYPE *,'ENTER DIAMETER OF SPHERICAL PRESSURE VESSEL (INCHES)'
      TYPE *,' '
      READ *,SPHDIA
      R=SPHDIA/2.
      VOL = ((4*3.14159*R**3)/3)/(1728.)    !VOLUME IN CUBIC FEET
      IF(NFLAG.EQ.3) GOTO 8022
C
C..................................................................
C
C     AMBIENT TEMPERATURE (FOR DETERMINATION OF GAS MASS)
C
804   CONTINUE
      TYPE *,'...............................................'
      TYPE *,' '
      TYPE *,'ENTER AMBIENT TEMPERATURE OF GAS (DEGREES F)'
      TYPE *,' '
      READ *,TEMP
C
C     CONVERT TEMPERATURE TO DEGREES RANKIN
C
      TEMP = TEMP + 459.67
C
C     CALCULATE THE WEIGHT OF GAS (LBS) INITIALLY INSIDE PRESSURE VESSEL:
C     NOTE THAT DRY NITROGEN, MOLECULAR WEIGHT 14 LB/MOLE
C     IS ASSUMED. IDEAL GAS THEORY IS USED TO CONVERT INTERNAL PRESSURE,
C     VOLUME, AND TEMPERATURE, AND MOLECULAR WEIGHT TO WEIGHT OF GAS
C     IN POUNDS.  FORMULA IS FROM CHEMISTRY TEXT; WT = (P*V)*MOL.WT./RT
C     OUTPUT IS POUNDS WEIGHT (NOT MASS).
C
C     A NOTE ON UNITS:
C
C       TOTP    =     TANK STARTING PRESSURE  [POUNDS(FORCE)/SQUARE FOOT]
C       VOL     =     TANK INTERNAL VOLUME    [CUBIC FEET]
C       TEMP    =     TEMPERATURE             [DEG. R = DEG. F + 459.67]
C       RCON    =     UNIVERSAL GAS CONSTANT  [FT-LBF/LB-MOLES DEG.RANKIN]
C                     [NOTE THAT POUND-MOLES ARE USED]
C       MOLWT         =       MOLECULAR WEIGHT OF GAS [LB/MOLE]
C                     [FOR NITROGEN, M = 28 LB/MOLE]
C                     [FOR OXYGEN  , M = 32 LB/MOLE]
C       WTOT    =     WEIGHT OF GAS STORED IN THE TANK [LBF]
C
8021  CONTINUE
8022  CONTINUE
```

```
C
C     USE MOLECULAR NITROGEN FOR PRESENT DESIGN
C
      MOLWT = 28.                        !POUNDS/MOLE
      RCON  = 1545.                      !UNIV. GAS CONSTANT
801    WTOT  = (TOTP*VOL*MOLWT)/(RCON*TEMP)     !UNITS = POUNDS FORCE
C
C     CONVERT WEIGHT OF GAS TO MASS (SLUGS)
C
      TMASS0 = WTOT                          !USE WEIGHT HERE SINCE WE WILL
C                                  !USE POUND/SEC FLOW RATES IN
C                                  !SUBSEQUENT CALCULATIONS,
C                                  !FOLLOWING FAGET, 1962,P266.
      IF(NFLAG.EQ.3) GOTO 601                !JUMP TO VERIFICATION MENU
      IF(NFLAG.EQ.4) GOTO 601                !JUMP TO VERIFICATION MENU
C
C     CONVERT INITIAL INTERNAL PRESSURE TO PSF FROM PSI
C
      P0 = TOTP*144.       !PRESSURE NOW IN PSF
      IF(NFLAG.EQ.2) GOTO 601                !JUMP TO VERIFICATION MENU
C
C...........................................................................
C
C     GET NOZZLE THROAT DIAMETER (INCHES)
C
805    CONTINUE
      TYPE *,'...................................................'
      TYPE *,' '
      TYPE *,'ENTER FIXED NOZZLE THROAT DIAMETER (INCHES)'
      TYPE *,' '
      READ *,D
C
C     CALCULATE THROAT AREA (SQ.FT.)
C
      A = ((D/12.)**2)*3.14159/4.       !NOTE, DIAMETER CONVERTED TO FEET
C                              !FROM INCHES
      IF(NFLAG.EQ.5) GOTO 601           !JUMP TO VERIFICATION MENU
C
C...........................................................................
C
C     GET INTEGRATION TIME STEP (SECONDS)
C
806    CONTINUE
      TYPE *,'...................................................'
      TYPE *,' '
      TYPE *,'ENTER INTEGRATION TIME STEP, DECIMAL SECONDS'
      TYPE *,' '
      READ *,DT
      IF(NFLAG.EQ.6) GOTO 601           !JUMP TO VERIFICATION MENU
C
C...........................................................................
C
C     SET UP THE INVERSE GAUSSIAN PULSE EQUATION
C
```

```
807   CONTINUE
      TYPE *,'...............................................'
      TYPE *,' '
      TYPE *,'ENTER THE MAXIMUM IMPULSE FORCE (POUNDS) DESIRED'
      TYPE *,' '
      READ *,FIMP
      IF(NFLAG.EQ.7) GOTO 8071          !ADJUST FIMP, K BUT NOT SIGMA
      TYPE *,'...............................................'
      TYPE *,' '
808   CONTINUE
      TYPE *,'ENTER SHAPE CONSTANT SIGMA (TYPICAL VALUE IS 0.15)'
      TYPE *,' '
      READ *,SG
8071  CONTINUE
C
C     CALCULATE CONSTANT "K" (SEE INVENTION NOTEBOOK #2, PP 66-67)
C
      REALK= (FIMP*SQRT(4*3.14159*((SG**2)/6.)**3))/(0.033469524)
      TYPE *,' '
      TYPE *,'.................................'
      TYPE *,' '
      TYPE *,' REALK [IN USER-INPUT] = ',REALK
      TYPE *,' '
      IF(NFLAG.EQ.7) GOTO 601           !JUMP TO VERIFICATION MENU
      IF(NFLAG.EQ.8) GOTO 601           !JUMP TO VERIFICATION MENU
C
C......................................................................
C
C     GET DESIRED PULSE WIDTH IN SECONDS
C
809   CONTINUE
      TYPE *,'...............................................'
      TYPE *,' '
      TYPE *,'ENTER DESIRED PULSE WIDTH IN SECONDS'
      TYPE *,' '
      READ *,TMAX
      IF(NFLAG.EQ.9) GOTO 601
C
C.............................................................'
C
C     GET THE FIXED DISPLACEMENT MAGNIFICATION FACTOR FOR
C     THE PIEZOELECTRIC STACK [THIS AFFECTS HARDWARE DESIGN]
C
810   CONTINUE
      TYPE *,'...............................................'
      TYPE *,' '
      TYPE *,'ENTER FIXED DISPLACEMENT AMPLIFICATION FACTOR '
      TYPE *,' '
      READ *,RA
C
C.............................................................
C
C     REVIEW THE INPUT, PERMIT EDITING
C
```

```
601   TYPE *,' '
      TYPE *,'..............................................'
      TYPE *,' '
      TYPE *,' ECHO OF DATA INPUT'
      TYPE *,' '
      TYPE *,'1:OK'
      TYPE *,'2:INITIAL PRESSURE (PSI)    = ',P0/144.
      TYPE *,'3:TANK DIAMETER (INCH)      = ',SPHDIA
      TYPE *,'4:TEMPERATURE (DEG. F)      = ',TEMP-459.67
      TYPE *,'5:NOZZLE DIAMETER (INCH)    = ',D
      TYPE *,'6:TIME STEP (SECONDS)       = ',DT
      TYPE *,'7:MAX GAUSS FORCE (POUNDS) = ',FIMP
      TYPE *,'8:SHAPE CONSTANT SIGMA      = ',SG
      TYPE *,'9:PULSE DURATION (SECONDS) = ',TMAX
      TYPE *,'10:DISPLACEMEMT MAG FACTOR = ',RA
      TYPE *,' '
      TYPE *,'ENTER A NUMBER [2-10] IF CHANGES ARE TO BE MADE,'
      TYPE *,'OTHERWISE, TYPE 1 TO CONTINUE'
      TYPE *,' '
      TYPE *,'..........................................'
      TYPE *,' '
      READ *,NFLAG
      IF(NFLAG.EQ.1) GOTO 604          !EVERYTHING OK, PROCEED WITH CALCS
      GOTO 603                         !CHANGES TO BE MADE, LOOP BACK
604   CONTINUE
C
C     SEND INPUT TO MAIN CALCULATION MODULES
C
      CALL PRESSURE_CALCS(TMASS0,P0,DT,D,VOL,FAMP,TMAX,REALK,TOTP,
     1 WTOT,NPTS,SG,DELTAMAX,RA)
C
      CALL PIEZO_CALCS(TMASS0,P0,DT,D,VOL,FAMP,TMAX,REALK,TOTP,
     1 WTOT,NPTS,DELTAMAX,RA)
C
C     EXIT TO MAIN PROGRAM
C
      NFLAG=1                        !INSURE FLAG IS SET TO NORMAL
      RETURN                         !FOR COMPLETE DATA ENTRY UNLESS
      END                            !OVERRIDDEN BY USER
```

```
C*************************************************************
C
      SUBROUTINE PRESSURE_CALCS(TMASS0,P0,DT,D,VOL,FAMP,
     1 TMAX,REALK,TOTP,WTOT,NPTS,SG,DELTAMAX,RA)
C
C      PURPOSE: PERFORMS CALCULATIONS FOR PROGRAM THRUSTER
C               REGARDING TRANSIENT PRESSURE.  THESE INCLUDE:
C
C               1. TIME TO ZERO PRESSURE
C               2. PRESSURE AND THRUST AS A FUNCTION OF TIME
C
C      PARAMETERS:
C
C      TOTP   : STARTING INTERNAL PRESSURE FOR SUPPLY VESSEL(    )
C      TEMP   : AMBIENT TEMPERATURE (       )
C      VOL    : INTERNAL VOLUME OF SUPPLY PRESSURE VESSEL (       )
C      D      : DIAMETER OF NOZZLE THROAT (       )
C      DT     : TIME STEP (DECIMAL SECONDS) FOR NUMERICAL INTEGRATION
C      M      : REMAINING MASS OF PRESSURIZED GAS IN THE PRESSURE
C               VESSEL AT TIME "T" (        )
C      P      : INTERNAL ("CHAMBER") PRESSURE AT TIME "T"   (        )
C      RHO    : GAS DENSITY AT TIME "T" (INSIDE "CHAMBER")  (           )
C      VEL    : VELOCITY AT NOZZLE THROAT AT  TIME "T"  (           )
C      DMDT   : MASS FLOW RATE AT TIME "T" THROUGH NOZZLE  (           )
C      T      : TOTAL ELAPSED TIME FROM MOMENT NOZZLE IS "OPENED" (SECONDS)
C      FGAUSS : THEORETICAL LOAD NEEDED AT TIME T(M) TO FOLLOW
C               THE INVERSE GAUSSIAN PROFILE DESCRIBED BY THE
C               THE USER.
C      DELTA  : CORE DISPLACEMENT REQUIRED AT TIME T(M) IN ORDER
C               TO PROVIDE THE THRUST "FGAUSS".  NOTE THAT BEYOND
C               A FIXED VALUE, CONTROLLED BY THE FIXED NOZZLE
C               THROAT DIAMETER, INCREASING CORE DISPLACEMENT
C               WILL NOT PROVIDE FURTHER INCREASES IN THRUST.
C      FTHRUST: ACTUAL THRUST AT TIME T(M), ACCOUNTING FOR
C               THE LIMITING CASE FOR DELTA.
C      RA     : MAGNIFICATION (LEVER) RATIO NEEDED FROM PIEZO
C               STACK TO ACHIEVE THE REQUIRED DELTA AT TIME T(M)
C
C      VERSION:    10-25-88       ORIGINAL
C                   5-16-89       GENERAL DEBUGGING: SPLIT
C                                 PROGRAM INTO INDEPENDENT MODULES
C      WRITTEN BY: STONE
C
C       SUBROUTINES CALLED: NONE
C
C*********************************************************************
C
C
      COMMON TMASS(5000),P(5000),RHO(5000),VEL(5000),DMDT(5000)
      COMMON T(5000),CMASS(5000),CRHO(5000),CPRES(5000)
      COMMON THRUST(5000),AT(5000),DISP(5000)
      COMMON FGAUSS(5000),DELTA(5000),FTHRUST(5000)
      COMMON DELTASTACK(5000),VDAC(5000),ATEFF(5000)
      REAL*8 MOLWT
```

```
C
      DIMENSION VL(5000),ACC(5000)
C
C      INITIALIZE INTEGRATION
C
      DELTAMAX=0.0
      M=0
12     M=M+1
      L=M-1
C
C...............................................................
C
C      FIRST STEP INITIALIZES ALL VARIABLES
C
      IF(M.GT.1) GOTO 13
      TMASS(1) = TMASS0        !STARTING MASS OF GAS IN PRESSURE VESSEL
      P(1) = P0                !INITIAL PRESSURE. NOTE: UNITS FOR PRESSURE
C                              !ARE (PSF); MASS IS (LB-SEC**2/FT)
C
      RHO(1)=TMASS(1)/VOL                !DENSITY (LB-SEC**2/FT**4)
      VEL(1) = SQRT(1.405*P(1)/RHO(1))   !THROAT VELOCITY = SPEED OF SOUND
C                                        !IN FEET/SECOND
C                                        !(SEE PRANDTL, P261, EQ.4)
C                                        !"ESSENTIALS OF FLUID DYNAMICS"
C                                        !BLACKIE & SON, 1952 (NIST LIBRARY)
C                                        !UNITS ARE (FT/SEC)
C                                        !NOTE THAT DUE TO CHANGES IN DENSITY
C                                        !WITH TIME, SPEED OF SOUND ALSO
C                                        !CHANGES
C                                        !SEE PRANDTL, P267, EQ.11)
      T(1) = DT                             !INCREMENT TIME STEP
      CPRES(1) = P(1)/144.                     !CONVERT PRESSURE BACK TO PSI
C
C...............................................................
C
C      CALCULATE THEORETICAL INVERSE GAUSSIAN IMPULSE FUNCTION FORCE
C

      ACONST=EXP((-1.*SG**2)/(4.*T(1)))
      FGAUSS(1)=(REALK*SG/(SQRT(4*3.14159*T(1)**3)))*ACONST
C      TYPE *,’ACONST = ’,ACONST
C      TYPE *,’FGAUSS(1) = ’,FGAUSS(1)
C
C...............................................................
C
C      CORE DISPLACEMENT (LIMIT SET SUCH THAT EFFECTIVE THROAT AREA
C      DOES NOT EXCEED THAT OF THE FIXED NOZZLE)
C
      DELTA(1) = FGAUSS(1)/(CPRES(1)*3.14159*D*1.8)      !INCHES
C
      DLIM=D/4.0             !1/4 NOZZLE THROAT DIAMETER (INCHES)
      IF (DELTA(1).GT.DLIM) DELTA(1) = DLIM
C
C...............................................................
```

```
C
C     EFFECTIVE THROAT AREA
C
      ATEFF(1) = 3.14159*D*DELTA(1)
C
C..........................................................................
C
C     ACTUAL THRUST (ACCOUNTING FOR DISPLACEMENT LIMITS)
C
      FTHRUST(1)= CPRES(1)*ATEFF(1)*1.629   !SEE NOTEBOOK #2,P67,101,102
C
C..........................................................................
C
C     CHECK FOR MAX CORE DISPLACEMENT
C
      IF(DELTA(1).GT.DELTAMAX) DELTAMAX=DELTA(1)
C
C..........................................................................
C
C     DETERMINE ACTUAL AMPLIFICATION RATIO FOR PIEZOELECTRIC AMPLIFIER
C     NOTE THAT 0.0027 IS THE MAXIMUM PHYSICALLY REALIZABLE
C     DISPLACEMENT FROM THE PIEZO STACK AT 225 VDC.
C
      AMPACT = DELTAMAX/0.0027
C
C..........................................................................
C
C     NOW CALCULATE WEIGHT FLOW RATE (LB/SEC), GIVEN ATEFF(1)
C
C     FORMULA FROM FAGET, 1962, PP 266-267] SEE ALSO WORKBOOK #2, P.101
C
C     GAMMA VALUES FOR VARIOUS GASES USED FOR PROPULSION:
C
C     NAME:     G=GAMMA:    MOLWT:  SPEC. GAS CONST = RCON2    ISP(SEC)
C     H2          1.410      2.015       766.37             296
C     N2          1.404     28.016        55.15              80
C     He          1.659      4.003       385.96             179
C
     MOLWT = 28.016          !MOLECULAR NITROGEN
     G     = 1.404           !MOLECULAR NITROGEN
     RCON2 = 1545/MOLWT      !SPECIFIC GAS CONSTANT FOR N2
C
C     THERE IS A DIRECT FORMULA FOR CALCULATION OF FUEL USAGE RATE
C     IN LBF/SEC DESCRIBED IN MY NOTES, BOOK #2, PAGE 101, USING
C     PROBLEM SPECIFIC PARAMETERS: GAMMA, CHAMBER PRESSURE, NOZZLE
C     THROAT AREA, AND TEMPERATURE.
C
C     HOWEVER, WEIGHT/SEC FLOW RATE CAN ALSO BE DETERMINED FROM
C     THE RELATION:
C
C     DMDT = THRUST/Isp      [WHERE Isp IS THE SPECIFIC IMPULSE]

      DMDT(1)= (FTHRUST(1))/80.   !LBF/SEC FLOW RATE FOR NITROGEN
      GOTO 12                     !Isp N2 = 80 SEC.
```

```
C
C...............................................................
C...............................................................
C
C     UPDATE NEW MASS, PRESSURE, DENSITY, VELOCITY, AND MASS FLOW RATE
C     FOR NEXT TIME STEP (SEE DETAILED COMMENTS ABOVE FOR DEFINITION
C     OF VARIABLES.
C
13    CONTINUE                    !DECREMENT THE MASS USING PRESENT FLOW
     TMASS(M)=TMASS(L)-DMDT(L)*DT !NOTE UNITS = (LB-SEC**2/FT); SLUGS
     P(M)=P0*TMASS(M)/TMASS0      !SCALE PRESSURE LINEARLY WITH MASS
     RHO(M) = TMASS(M)/VOL        !DENSITY
     VEL(M)=SQRT(1.4*P(M)/RHO(M)) !VELOCITY
     T(M) = T(L) + DT             !ELAPSED TIME
     CPRES(M) = P(M)/144.         !CONVERT PRESSURE BACK TO PSI
     ACONST=EXP((-1.*SG**2)/(4.*T(M)))    !CALCULATE IDEAL THRUST
     FGAUSS(M)=(REALK*SG/(SQRT(4*3.14159*T(M)**3)))*ACONST

     DELTA(M) = FGAUSS(M)/(CPRES(M)*3.14159*D*1.8)  !CORE DISPLACEMENT
     DLIM=D/4.0               !1/4 NOZZLE THROAT DIAMETER (INCHES)
     IF (DELTA(M).GT.DLIM) DELTA(M) = DLIM   !LIMITING CORE DISPL.
     ATEFF(M) = 3.14159*D*DELTA(M)           !EFFECTIVE NOZZLE AREA
     FTHRUST(M)= CPRES(M)*ATEFF(M)*1.629     !ATTAINABLE THRUST
     IF(DELTA(M).GT.DELTAMAX) DELTAMAX=DELTA(M)  !CHECK MAX DISPL.
     AMPACT = DELTAMAX/0.0027        !ACTUAL PIEZO MAG FACTOR
     DMDT(M)=(FTHRUST(M))/80.        !WEIGHT FLOW RATE (LBF/SEC)
C
C...............................................................
C
C     CHECK TO SEE IF ELAPSED TIME IS GREATER THAN THE REQUESTED
C     PULSE WIDTH
C
     IF(T(M).GE.TMAX) THEN
               TYPE *,'TIME = REQUESTED PULSE WIDTH'
               TYPE *,'NORMAL TERMINATION'
               ENDIF
     IF(T(M).GE.TMAX) GOTO 103
C
C...............................................................
C
C     CHECK IF LIMITING PRESSURE CUT OFF HAS BEEN REACHED
C
     IF(CPRES(M).LE.30)THEN          !30 PSI IS ARBITRARY SHUTDOWN
                TYPE *,'PRESSURE BELOW 30 PSI;ENDING CALCS'
                TYPE *,'LARGER GAS BOTTLE NEEDED, OR '
                TYPE *,'INCREASE INITIAL CHARGING PRESSURE'
                ENDIF
     IF(CPRES(M).LE.30) GOTO 103 !KICK OUT OF LOOP WHEN PRESSSURE TOO LOW.
C
C...............................................................
C
C     PRINT OUT RESULTS (OPTIONAL)
C
C      TYPE *,'TMASS      PRESS      RHO      VEL      DMDT     TIME'
```

```
C      WRITE(6,14) TMASS(M),CPRES(M),RHO(M),VEL(M),DMDT(M),T(M)
C14     FORMAT(3X,F6.3,3X,F9.3,1X,F8.6,1X,F6.1,2X,F6.4,2X,F6.3,2X,F8.3)
C      TYPE *,’FGAUSS   DELTA   ATEFF’
C      WRITE(6,15) FGAUSS(M),DELTA(M),ATEFF(M)
C15     FORMAT(3X,F6.3,3X,F9.6,3X,F9.6)
102    CONTINUE
C
C..................................................................
C
C      CHECK ON MAX NUMBER OF ITERATIONS
C
      IF(M.GE.4999) THEN
                TYPE *,’ITERATIONS > 4999; ENDING CALCULATIONS’
                TYPE *,’INCREASE SIZE OF TIME STEP’
                ENDIF
      IF(M.GE.4999) GOTO 103            !TOO MANY ITERATIONS
      GOTO 12                           !NEXT ITERATION
C
103    CONTINUE
C
C      NOTE THAT FINAL "M" VALUE = NUMBER OF DATA POINTS = NPTS
C
      NPTS=M
C
C
C      CALCULATE PEAK ACCELERATION OF CORE ROD MASS
C      [DO THIS BY FINITE DIFFERENTIATION OF THE DISPLACEMENT VS
C       TIME CURVE]
C
      DO 6015 MM=2,M
      VL(MM)=(DELTA(MM)-DELTA(MM-1))/DT     !VELOCITY
6015   CONTINUE
      VL(1)=0             !NO VELOCITY AT START
      DO 6016 MM=1,M-1
      ACC(MM)=(VL(MM+1)-VL(MM))/DT         !ACCELERATION
6016   CONTINUE
      ACC(M)=0       !NO FURTHER ACCELERATION
C
C      DETERMINE MAX AND MIN ACCELERATION, EFFECTIVE MAXIMUM
C      WEIGHT OF CORE THAT CAN BE RETURNED BY MEANS OF A 50 POUND
C      COMPRESSION SPRING, AND AT WHAT TIME THE PEAK ACCELERATION
C      OCCURS.
      WMIN=0
      WMAX=0
      TMAX=0
      TMIN=0
      ACCMAX=0
      ACCMIN=0
      DO 6017 MM=1,M
      IF(ACCMAX.LT.ACC(MM)) THEN
         ACCMAX=ACC(MM)
         TMAX=DT*MM
      ENDIF
      IF(ACCMIN.GT.ACC(MM)) THEN
```

```
          ACCMIN=ACC(MM)
          TMIN=DT*MM
      ENDIF
6017   CONTINUE
      WMAX= (50.*386)/ACCMAX
      WMIN= (50.*386)/ACCMIN        !F=50 LBS; 386 IN/S**2 CONVERT
                                    ! TO MASS UNITS.
      TYPE *,' '
      TYPE *,'.............................................'
      TYPE *,' '
      TYPE *,' SUMMARY OF RUN'
      TYPE *,' '
      TYPE *,' INITIAL WEIGHT OF GAS (POUNDS)    = ',WTOT
      TYPE *,' INITIAL INTERNAL PRESSURE (PSIA) = ',TOTP
      TYPE *,' PRESSURIZED VOLUME (CUBIC FEET)  = ',VOL
      TYPE *,' DIAMETER OF NOZZLE (INCHES)       = ',D
      TYPE *,' NUMBER OF POINTS (NPTS)           = ',NPTS
      TYPE *,' ACTUAL STACK MAG FACTOR           = ',AMPACT
      TYPE *,' WMAX (LBS, PIEZO EXTEND)          = ',WMAX
      TYPE *,' TMAX (SECONDS)                    = ',TMAX
      TYPE *,' WMIN (LBS, SPRING RETURN)         = ',WMIN
      TYPE *,' TMIN (SECONDS)                    = ',TMIN
      IF(AMPACT.GT.RA) THEN
      TYPE *,' '
      TYPE *,' [**************************************************]'
      TYPE *,' [ REQUIRED STACK MAGNIFICATION IS GREATER THAN     ]'
      TYPE *,' [ AVAILABLE WITH PRESENT FIXED AMPLIFIER.          ]'
      TYPE *,' [ INCREASE FIXED MAG FACTOR, OR NOZZLE DIAMETER,   ]'
      TYPE *,' [ OR DECREASE MAX IMPULSE.                         ]'
      TYPE *,' [**************************************************]'
      TYPE *,' '
      ENDIF
      TYPE *,' '
      TYPE *,' TYPE ANY KEY + CARRIAGE RETURN TO CONTINUE:'
      TYPE *,' '
      TYPE *,'.............................................'
      READ *,NJUNK
      GOTO 130                  !NORMAL TERMINATION
120   TYPE *,'ITERATION LIMIT REACHED: RESET TIME STEP TO LARGER VALUE'
130   CONTINUE
      RETURN
      END
```

```
C***********************************************************
      SUBROUTINE PIEZO_CALCS(TMASS0,P0,DT,D,VOL,FAMP,
     1 TMAX,REALK,TOTP,WTOT,NPTS,DELTAMAX,RA)
C
C      PURPOSE: PERFORMS CALCULATIONS FOR PROGRAM THRUSTER
C               REGARDING PIEZO CRYSTAL DRIVING CHARACTERISTICS
C               AND THEORETICAL THRUST VALUES.  THESE INCLUDE:
C      PARAMETERS:
C      DELTASTACK: ACTUAL DISPLACEMENT REQUIRED FROM THE PIEZO
C                  STACK AT TIME T(M), ASSUMING A FIXED AMPLIFICATION
C                  (LEVER) AT TIME T(M).
C      VDAC      : DRIVE VOLTAGE REQUIRED TO BE DELIVERED TO
C                  THE PIEZO STACK FROM THE DAC UNIT AT TIME T(M).
C      ATEFF     : EFFECTIVE NOZZLE THROAT AREA AT TIME T(M).
C                  THIS IS <= FIXED NOZZLE THROAT AREA AND IS
C                  DETERMINED BY VALVE CORE DISPLACEMENT, DELTA.
C      FIXNOZAREA: AREA OF FIXED NOZZLE CROSS SECTION.
C      VERSION:    10-25-88      ORIGINAL
C                  5-16-89       GENERAL DEBUGGING: SPLIT
C                                PROGRAM INTO INDEPENDENT MODULES
C      WRITTEN BY: STONE
C
C       SUBROUTINES CALLED: NONE
C
C**********************************************************************
C
      COMMON TMASS(5000),P(5000),RHO(5000),VEL(5000),DMDT(5000)
      COMMON T(5000),CMASS(5000),CRHO(5000),CPRES(5000)
      COMMON THRUST(5000),AT(5000),DISP(5000)
      COMMON FGAUSS(5000),DELTA(5000),FTHRUST(5000)
      COMMON DELTASTACK(5000),VDAC(5000),ATEFF(5000)
C
      DO 1001 M=1,NPTS
C.................................................................
C
C     DETERMINE REQUIRED PIEZOELECTRIC STACK EXPANSION AT TIME T
C     [NOTE THAT RA IS THE FIXED DISPL. AMPLIFICATION FACTOR DEFINED
C      IN USER_INPUT.  AN ERROR FLAG WILL BE SENT TO THE TERMINAL
C      IF THE ACTUAL REQUIRED AMPLIFICATION EXCEEDS THIS AMOUNT.
C      THIS CHECK TAKES PLACE IN PRESSURE_CALCS.
C
      DELTASTACK(M) = DELTA(M)/RA
C
C     DETERMINE DAC OUTPUT VOLTAGE (TO PIEZO STACK) AT TIME T TO
C     ACHIEVE DESIRED THRUST LEVEL.  NOTE THAT 0.000012 INCH/VOLT
C     IS THE CALIBRATED COMPLIANCE OF THE STACK; 50:1 IS THE
C     AMPLIFICATION PROVIDED BY THE LINEAR AMPLIFIER IN THE PRESENT
C     DESIGN TO BOOST THE DAC OUTPUT VOLTAGE TO THAT REQUIRED
C     TO DRIVE THE STACK.
C
      VDAC(M) = DELTASTACK(M)/((.000012)*50.
1001  CONTINUE
      RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE PRESSURE_PLOT(NPTS)
C      PURPOSE: POST-PROCESSOR PLOT MENU
C      GRAPHICS: PROGRAM MAKES USE OF RASTER TECH MODEL
C      ONE/80 COLOR GRAPHICS DEVICE FOR OUTPUT DISPLAY.
C      VERSION: 10-28-88 : ORIGINAL
C               5-16-89  : STAND-ALONE MODULE
C      WRITTEN BY: STONE
C***********************************************************************
      COMMON TMASS(5000),P(5000),RHO(5000),VEL(5000),DMDT(5000)
      COMMON T(5000),CMASS(5000),CRHO(5000),CPRES(5000)
      COMMON THRUST(5000),AT(5000),DISP(5000)
      COMMON FGAUSS(5000),DELTA(5000),FTHRUST(5000)
      COMMON DELTASTACK(5000),VDAC(5000),ATEFF(5000)
C
      DIMENSION X(5000),Y(5000),MDATA(12000)
      DIMENSION XBAR(11),STD(11),D(11,11),SUMSQ(11),ISAVE(11),RX(11,11)
      DIMENSION RY(10),B(10),SB(10),ANS(10),SUMX(1001)
      DIMENSION XP(1001),YP(1001)
C
      REAL*4 MDATA
C      CHARACTER DATATITLE*10,DATFILE*14
      INTEGER*2 TITLE(40),XTITLE(19),YTITLE(19)
      INTEGER*2 IX,IY,IRED,IGRN,IBLU,OFFSETX,OFFSETY,ACTRY,ACTRX
      REAL*8 XMAX,XMIN,YMAX,YMIN
C
C      PRESENT THE MENU
C
4300   CONTINUE
      TYPE *,'*****************************************************'
      TYPE *,'*'
      TYPE *,'*  PLOT MENU: '
      TYPE *,'*'
      TYPE *,'*  1: PREPARE AN X-Y PLOT'
      TYPE *,'*  6: RETURN TO MAIN MENU'
      TYPE *,'* '
      TYPE *,'*****************************************************'
      READ *,NANS
      IF(NANS.EQ.1)THEN
C
C      INTERACTIVE MENU FOR CHOOSING X AND Y AXIS PLOT PARAMETERS
C
      CALL PLOTPREP(NPTS,X,Y,XMIN,XMAX,YMIN,YMAX,TITLE,XTITLE,YTITLE)
C
C      INITIALIZE THE GRAPHICS DEVICE AND PLOT THE DATA
C
      CALL SETUP_GRAPHICS(NPTS,XMAX,YMAX,XMIN,YMIN,X,Y,TITLE,
     1 OFFSETX,OFFSETY,XFACTOR,YFACTOR,OFFX,OFFY,XTITLE,YTITLE)
      ENDIF
      IF(NANS.EQ.6) GOTO 10000
      GOTO 4300
10000  CONTINUE
      RETURN
      END
```

```
C************************************************************************
C
      SUBROUTINE PLOTPREP(NPTS,
     1 X,Y,XMIN,XMAX,YMIN,YMAX,TITLE,IXTITLE,IYTITLE)
C
C      PURPOSE: PERMITS USER TO CHOOSE FROM MENUS OF OUTPUT FROM
C               THRUSTER_MAIN AND TO CONSTRUCT X-Y PLOTS OF THOSE VARIABLES.
C               THIS ROUTINE COMPUTES MAXIMA AND MINIMA FOR THE VARIABLES
C               ASSIGNED TO THE PLOT VECTORS X AND Y.
C
C      IMPLEMENTATION:  CALL PLOTPREP
C
C      OTHER ROUTINES CALLED: SORT (FOR SORTING DATA PRIOR TO DETERMINING
C                             MAXIMA AND MINIMA)
C
C      WRITTEN BY: STONE
C      VERSION   : 10-28-88 ; 5-16-89
C
C**************************************************************************
C
      CHARACTER*38 XTITLE,YTITLE
      INTEGER*2 TITLE(40),IXTITLE(19),IYTITLE(19)
      DIMENSION X(5000),Y(5000),X1(5000),Y1(5000)
      COMMON TMASS(5000),P(5000),RHO(5000),VEL(5000),DMDT(5000)
      COMMON T(5000),CMASS(5000),CRHO(5000),CPRES(5000)
      COMMON THRUST(5000),AT(5000),DISP(5000)
      COMMON FGAUSS(5000),DELTA(5000),FTHRUST(5000)
      COMMON DELTASTACK(5000),VDAC(5000),ATEFF(5000)
      INTEGER*2 NPTS
      REAL*8 XMAX,XMIN,YMAX,YMIN
C
C      P         = CHAMBER PRESSURE AT TIME 'T' (PSI)
C      TMASS     = GAS MASS IN CHAMBER AT TIME 'T' (SLUGS)
C      RHO       = MASS DENSITY IN CHAMBER AT TIME 'T' (SLUGS/CU.FT.)
C      DMDT      = MASS FLOW RATE AT TIME 'T' (SLUGS/SECOND)
C      FGAUSS    = ANALYTICAL INVERSE GAUSSIAN FORCE AT TIME 'T' (POUNDS
C      DELTA     = VALVE CORE DISPLACEMENT AT TIME 'T' (INCHES)
C      FTHRUST   = ACTUAL THRUST AT TIME 'T', ACCOUNTING FOR LIMITING
C                  EFFECTIVE NOZZLE THROAT AREA AND INTERNAL PRESSURE
C    DELTASTACK = REQUIRED DISPLACEMENT OF PIEZOELECTRIC STACK (INCH)
C       VDAC     = REQUIRED OUTPUT VOLTAGE FROM DAC AT TIME 'T'
C                  TO INDUCE REQUIRED DISPLACEMENT IN PIEZOELECTRIC STACK (VOLT)

C      SET UP THE X-AXIS MENU
C
      TYPE *,' '
      TYPE *,'***************************************'
      TYPE *,' '
      TYPE *,' X-AXIS MENU: PLEASE  CHOOSE  ONE  OF '
      TYPE *,'              THE FOLLOWING VARIABLES:'
      TYPE *,' '
      TYPE *,'  1: INTERNAL PRESSURE'
      TYPE *,'  2: MASS OF REMAINING GAS'
      TYPE *,'  3: GAS DENSITY'
      TYPE *,'  4: MASS FLOW RATE'
      TYPE *,'  5: IDEAL (GAUSSIAN) THRUST'
      TYPE *,'  6: ACTUAL THRUST '
      TYPE *,'  7: THRUST ERROR (IDEAL-ACTUAL)'
      TYPE *,'  8: VALVE CORE DISPLACEMENT'
      TYPE *,'  9: PIEZO STACK DISPLACEMENT'
      TYPE *,' 10: DAC OUTPUT VOLTAGE TO STACK'
      TYPE *,' 11: ELAPSED TIME'
      TYPE *,' 12: STOP'
      TYPE *,' '
      TYPE *,'****************************************'
      READ *,NANSX
C
C      SET UP THE Y-AXIS MENU
C
      TYPE *,' '
      TYPE *,'***************************************'
      TYPE *,' '
      TYPE *,' Y-AXIS MENU: PLEASE  CHOOSE  ONE  OF '
      TYPE *,'               THE FOLLOWING VARIABLES:'
      TYPE *,' '
      TYPE *,'  1: INTERNAL PRESSURE'
      TYPE *,'  2: MASS OF REMAINING GAS'
      TYPE *,'  3: GAS DENSITY'
      TYPE *,'  4: MASS FLOW RATE'
      TYPE *,'  5: IDEAL (GAUSSIAN) THRUST'
      TYPE *,'  6: ACTUAL THRUST '
      TYPE *,'  7: THRUST ERROR (IDEAL-ACTUAL)'
      TYPE *,'  8: VALVE CORE DISPLACEMENT'
      TYPE *,'  9: PIEZO STACK DISPLACEMENT'
      TYPE *,' 10: DAC OUTPUT VOLTAGE TO STACK'
      TYPE *,' 11: ELAPSED TIME'
      TYPE *,' 12: STOP'
      TYPE *,' '
      TYPE *,'****************************************'
      READ *,NANSY
C
C      LOAD THE AXIS TITLES
C
C      X-AXIS TITLE
C
      IF(NANSX.EQ.1)  XTITLE = 'INTERNAL CHAMBER PRESSURE (PSI)   '
      IF(NANSX.EQ.2)  XTITLE = 'MASS OF REMAINING GAS (SLUGS)     '
      IF(NANSX.EQ.3)  XTITLE = 'GAS DENSITY (SLUGS/CU.FT.)        '
      IF(NANSX.EQ.4)  XTITLE = 'MASS FLOW RATE (SLUGS/SEC)        '
      IF(NANSX.EQ.5)  XTITLE = 'IDEAL (GAUSSIAN) THRUST  (LBS)    '
      IF(NANSX.EQ.6)  XTITLE = 'ACTUAL THRUST  (LBS)              '
      IF(NANSX.EQ.7)  XTITLE = 'THRUST ERROR (IDEAL-ACTUAL)       '
      IF(NANSX.EQ.8)  XTITLE = 'VALVE CORE DISPLACEMENT (MILS)    '
      IF(NANSX.EQ.9)  XTITLE = 'PIEZO STACK DISPLACEMENT(MILS)    '
      IF(NANSX.EQ.10) XTITLE = 'DAC OUTPUT VOLTAGE TO STACK       '
      IF(NANSX.EQ.11) XTITLE = 'ELAPSED TIME (SECONDS)            '
C
C      Y-AXIS TITLE
```

```
C
      IF(NANSY.EQ.1)  YTITLE = 'INTERNAL CHAMBER PRESSURE (PSI)   '
      IF(NANSY.EQ.2)  YTITLE = 'MASS OF REMAINING GAS (SLUGS)     '
      IF(NANSY.EQ.3)  YTITLE = 'GAS DENSITY (SLUGS/CU.FT.)        '
      IF(NANSY.EQ.4)  YTITLE = 'MASS FLOW RATE (SLUGS/SEC)        '
      IF(NANSY.EQ.5)  YTITLE = 'IDEAL (GAUSSIAN) THRUST (LBS)     '
      IF(NANSY.EQ.6)  YTITLE = 'ACTUAL THRUST  (LBS)              '
      IF(NANSY.EQ.7)  YTITLE = 'THRUST ERROR (IDEAL-ACTUAL)       '
      IF(NANSY.EQ.8)  YTITLE = 'VALVE CORE DISPLACEMENT (MILS)    '
      IF(NANSY.EQ.9)  YTITLE = 'PIEZO STACK DISPLACEMENT (MILS)   '
      IF(NANSY.EQ.10) YTITLE = 'DAC OUTPUT VOLTAGE TO STACK       '
      IF(NANSY.EQ.11) YTITLE = 'ELAPSED TIME (SECONDS)            '
C
C     CONVERT TITLES TO INTEGER FORMAT
C
      DECODE(38,1111,XTITLE)IXTITLE
      DECODE(38,1112,YTITLE)IYTITLE
1111  FORMAT(19A2)
1112  FORMAT(19A2)
C
C     LOAD THE VARIABLES CHOSEN INTO THE X AND Y VECTORS
C
C     ASSIGNING VALUES TO THE X ARRAY
C
      DO 999 KKK=1,NPTS
      IF(NANSX.EQ.1)  X(KKK)=(P(KKK))/144.
      IF(NANSX.EQ.2)  X(KKK)=TMASS(KKK)
      IF(NANSX.EQ.3)  X(KKK)=RHO(KKK)
      IF(NANSX.EQ.4)  X(KKK)=DMDT(KKK)
      IF(NANSX.EQ.5)  X(KKK)=FGAUSS(KKK)
      IF(NANSX.EQ.6)  X(KKK)=FTHRUST(KKK)
      IF(NANSX.EQ.7)  X(KKK)=FGAUSS(KKK)-FTHRUST(KKK)
      IF(NANSX.EQ.8)  X(KKK)=DELTA(KKK)*1000.
      IF(NANSX.EQ.9)  X(KKK)=DELTASTACK(KKK)*1000.
      IF(NANSX.EQ.10) X(KKK)=VDAC(KKK)
      IF(NANSX.EQ.11) X(KKK)=T(KKK)
C
C     ASSIGNING VALUES TO THE Y ARRAY
C
      IF(NANSY.EQ.1)  Y(KKK)=(P(KKK))/144.
      IF(NANSY.EQ.2)  Y(KKK)=TMASS(KKK)
      IF(NANSY.EQ.3)  Y(KKK)=RHO(KKK)
      IF(NANSY.EQ.4)  Y(KKK)=DMDT(KKK)
      IF(NANSY.EQ.5)  Y(KKK)=FGAUSS(KKK)
      IF(NANSY.EQ.6)  Y(KKK)=FTHRUST(KKK)
      IF(NANSY.EQ.7)  Y(KKK)=FGAUSS(KKK)-FTHRUST(KKK)
      IF(NANSY.EQ.8)  Y(KKK)=DELTA(KKK)*1000.
      IF(NANSY.EQ.9)  Y(KKK)=DELTASTACK(KKK)*1000.
      IF(NANSY.EQ.10) Y(KKK)=VDAC(KKK)
      IF(NANSY.EQ.11) Y(KKK)=T(KKK)
C
999   CONTINUE
C
C     GET THE TITLE FOR THE PLOT
```

```
C
      TYPE *,'PLEASE ENTER A TITLE FOR THE PLOT '
      READ (5,1)TITLE
1      FORMAT(40A2)
C
C      CONVERT TO INTEGER FORMAT
C
C      SORT THE X,Y PAIRS AND DETERMINE MAXIMA AND MINIMA
C
      DO 104 I=1,NPTS
      X1(I)=X(I)
      Y1(I)=Y(I)
104    CONTINUE
      CALL SORT(X,Y,NPTS)
      XMAX=X(NPTS)
      XMIN=X(1)
      CALL SORT(Y1,X1,NPTS)
      YMAX=Y1(NPTS)
      YMIN=Y1(1)
      RETURN
      END
```

```
C*********************************************************************
C
C  SUBROUTINE SORT
C
C  PURPOSE: TO SORT THE EXPERIMENTAL X,Y DATA PAIRS IN ASCENDING
C          ORDER PRIOR TO STATISTICAL FIT CALCULATIONS.  PROGRAM
C
C  IMPLEMENTATION: PROGRAM KEYS ON X DATA AND SWAPS Y VALUES
C                  ACCORDINGLY THEN RETURNS SORTED X AND Y VECTORS
C                TO THE MAIN PROGRAM
C
C  VARIABLES:
C
C      X: VECTOR CONTAINING INDEPENDENT VARIABLE EXPERIMENTAL
C         DATA OBSERVATIONS
C      Y: VECTOR CONTAINING DEPENDENT VARIABLE EXPERIMENTAL
C         DATA OBSERVATIONS
C      N: NUMBER OF EXPERIMENTAL OBSERVATIONS (X,Y) PAIRS.
C
C        VERSION: 5-16-89
C
C        WRITTEN BY: B. STONE
C
C*********************************************************************
C
      SUBROUTINE SORT(X,Y,N)
      DIMENSION  X(N),Y(N)
C      TYPE *,'IN SORT, NPTS=',N
C
      DO 735 I=1,N-1
      J=I+1
      DO 736 K=J,N
      IF(X(I).LE.X(K)) GOTO 736
      PARAMX=X(I)
      PARAMY=Y(I)
      X(I)=X(K)
      Y(I)=Y(K)
      X(K)=PARAMX
      Y(K)=PARAMY
736    CONTINUE
735    CONTINUE
C
      RETURN
      END
```

```
C*******************************************************
C
      SUBROUTINE SETUP_GRAPHICS(NPTS,XMAX,YMAX,XMIN,YMIN,X,Y,TITLE,
     1 OFFSETX,OFFSETY,XFACTOR,YFACTOR,OFFX,OFFY,XTITLE,YTITLE)
C
C      PURPOSE: INITIALIZE GRAPHICS DEVICE AND PLOT
C               THE RAW DATA POINTS
C
C      VERSION:    10-31-88; 5-16-89
C      WRITTEN BY: STONE
C
C         GRAPHICS FIRMWARE CALLS: (RASTER TECH ONE/80 SPECIFIC)
C
C         RTSET()        SPECIFY "WARM START"
C         RTINIT()       SPECIFY DEVICE DRIVER
C         ENTGRA         ENTER GRAPHICS MODE
C         LUT8() \       DEFINE COLOR LOOK-UP TABLE ENTRY
C         VAL8()         SPECIFY CURRENT ACTIVE COLOR
C         FLOOD          FLOOD BACKGROUND WITH CURRENT ACTIVE COLOR
C         MOVABS()       MOVE ABSOLUTE (PEN)
C         PRMFIL()       POLYGON FILL FLAG (1=FILL;0=LINE PLOT)
C         RECTAN()       DRAW A RECTANGLE FROM CURRENT LOC. TO THESE ENDPOINTS
C
C*******************************************************
C
      REAL*4 MDATA
C      CHARACTER DATATITLE*10,DATFILE*14
      COMMON TMASS(5000),P(5000),RHO(5000),VEL(5000),DMDT(5000)
      COMMON T(5000),CMASS(5000),CRHO(5000),CPRES(5000)
      COMMON THRUST(5000),AT(5000),DISP(5000)
      COMMON FGAUSS(5000),DELTA(5000),FTHRUST(5000)
      COMMON DELTASTACK(5000),VDAC(5000),ATEFF(5000)
      DIMENSION X(5000),Y(5000),MDATA(12000)
      DIMENSION XBAR(11),STD(11),D(11,11),SUMSQ(11),ISAVE(11),RX(11,11)
      DIMENSION RY(10),B(10),SB(10),ANS(10),SUMX(1001)
      DIMENSION XP(1001),YP(1001)
      INTEGER*2 TITLE(40),XTITLE(19),YTITLE(19)
C      INTEGER*2 IDX(16),IDY(16)
      INTEGER*2 IX,IY,IRED,IGRN,IBLU,OFFSETX,OFFSETY,ACTRY,ACTRX
      REAL*8 XMAX,XMIN,YMAX,YMIN
C      DATA IDX/0,1,0,-1,0,1,1,0,0,1,0,0,0,-1,-1,-1/,
C      1    IDY/0,0,1,0,1,0,0,-1,-1,0,1,1,1,0,0,0/
C
      IF(NPTS.LE.5000) GOTO 125
      TYPE *,'MAXIMUM DATA POINTS EXCEEDED'
      TYPE *,'SEE CODE COMMENTARY FOR ARRAY DIMENSION CHANGES'
      GOTO 10011
125    CONTINUE
C
C      THE PROGRAM NOW ALLOWS MULTIPLE PLOTS TO BE OVERLAYED
C      ON ONE WINDOW.  PROMPT THE USER TO SPECIFY WHETHER THE
C      CURRENT PLOT IS TO BE OVERLAYED ON THE PREVIOUS SCREEN.
C
      TYPE *,' '
      TYPE *,'..............................................'
      TYPE *,' '
      TYPE *,' DO YOU WANT THIS PLOT SUPERIMPOSED ON THE'
      TYPE *,' PREVIOUS SCREEN IMAGE?  [ANSWER NO IF THIS'
      TYPE *,' IS THE FIRST PLOT IN A SERIES] 1=YES; 2=NO'
      TYPE *,' '
      TYPE *,'..............................................'
      TYPE *,' '
      READ *,MULTPLOT
      IF(MULTPLOT.EQ.1) GOTO 10012
C
C  NEW PLOT.  THEREFORE,
C  INITIALIZE THE GRAPHICS DEVICE
C
      CALL RTSET(1,180)
      CALL RTINIT('GDA0:',5)
      CALL ENTGRA
C
C  LOAD THE COLOR MAP
C
      CALL LUT8(0,0,0,0)! BLACK
      CALL LUT8(1,255,150,255)        ! LIGHT PURPLE
      CALL LUT8(2,0,0,190)        ! BLUE (DULL)
      CALL LUT8(3,0,0,255)    ! BRIGHT BLUE
      CALL LUT8(4,0,255,255)  ! LIGHT BLUE
      CALL LUT8(5,150,255,255)!  "      "
      CALL LUT8(6,200,255,255)!  "      "
      CALL LUT8(7,0,200,200)  ! BLUE GREEN
      CALL LUT8(8,0,130,0)    ! OLIVE GREEN
      CALL LUT8(9,0,255,0)   ! BRIGHT GREEN
      CALL LUT8(10,130,255,130)! LIGHT GREEN
      CALL LUT8(11,255,255,200)! YELLOW
      CALL LUT8(12,255,255,100)! LIGHT YELLOW
      CALL LUT8(13,255,120,0)  ! ORANGE
      CALL LUT8(15,255,0,0)     ! RED
      CALL LUT8(14,255,150,150)! PINK
      CALL LUT8(16,255,200,200)! LIGHT PINK
      CALL LUT8(17,200,200,200)! LIGHT GRAY
      CALL LUT8(18,150,150,150)! GRAY
      CALL LUT8(19,50,50,50)   ! DARK GRAY
      CALL LUT8(30,255,255,255)! WHITE
      CALL LUT8(31,0,0,0)       ! BLACK
      CALL LUT8(32,IRED,IGRN,IBLU)
C
C     SET UP THE SCREEN WORKSPACE
C
C     FIRST FLOOD THE BACKGROUND
C
      CALL VAL8(3)        !BLUE
      CALL FLOOD
C
C     PORTION OFF THE GRAPH WORKSPACE IN BLACK
C
      CALL MOVABS(-600,480)
      CALL PRMFIL(1)
      CALL VAL8(0)        !BLACK
      CALL RECTAN(600,-320)
      CALL PRMFIL(0)
C
C      DRAW THE AXES AND PLOT THE DATA POINTS
C
10012  CONTINUE
      CALL POINTPLOT(NPTS,XMAX,YMAX,XMIN,YMIN,X,Y,TITLE,
     1 OFFSETX,OFFSETY,XFACTOR,YFACTOR,OFFX,OFFY,XTITLE,YTITLE,MULTPLOT)
C      TYPE *,'XFACTOR=',XFACTOR,'  YFACTOR=',YFACTOR,'IN MAIN'
C
10011  CONTINUE
      RETURN
      END
```

```
C *****************************************************************
C
C     SUBROUTINE POINTPLOT
C
C     PURPOSE: SCALES EXPERIMENTAL DATA TO SCREEN COORDINATES,
C            DRAWS AND LABELS THE AXES, AND PLOTS THE POINTS.
C
C     IMPLEMENTATION: CALL POINTPLOT(NPTS,XMAX,YMAX,XMIN,YMIN,X,Y
C                TITLE,OFFSETX,OFFSETY,XFACTOR,YFACTOR,
C                OFFX,OFFY)
C
C      DESCRIPTION OF PARAMETERS:
C
C      NPTS    - NUMBER OF X,Y DATA PAIRS (POINTS)
C      XMAX    - MAXIMUM VALUE OF INDEPENDENT VARIABLE
C      YMAX    - MAXIMUM VALUE OF DEPENDENT VARIABLE
C      XMIN    - MINIMUM VALUE OF INDEPENDENT VARIABLE
C      YMIN    - MINIMUM VALUE OF DEPENDENT VARIABLE
C      X       - VECTOR OF LENGTH (NPTS) CONTAINING INDEPENDENT
C               VARIABLE DATA
C      Y       - VECTOR OF LENGTH (NPTS) CONTAINING DEPENDENT
C               VARIABLE DATA
C      IDX     - VECTOR CONTAINING 16 RELATIVE OFFSET VALUES
C               (DEFINED IN MAIN PROGRAM) USED FOR DETERMINING
C               THE DEGREE OF 'BOLDNESS' OF A TITLE OR LABEL.
C               PROGRAM ITERATES ON THESE OFFSETS TO SMEAR A LABEL
C               GIVING THE APPEARANCE OF BOLDER TEXT.
C      IDY     - SAME AS IDX, BUT Y OFFSETS
C      TITLE   - OVERALL PLOT TITLE READ FROM THE DATA FILE
C      OFFSETX - THE PIXEL OFFSET APPLIED TO SCALED DATA, AXES
C               ETC IN THE X DIRECTION SO THAT THE ENTIRE IMAGE
C               WILL BE PROPERLY LOCATED IN THE SCREEN WINDOW.
C      OFFSETY - SAME AS OFFSETX, BUT USED FOR Y POSITIONING
C               WITHIN THE WINDOW.  THESE ARE PASSED BACK TO THE
C               MAIN PROGRAM AND PICKED UP BY SUBROUTINE LINEPLOT
C               FOR MOVING THE REGRESSION LINE TO THE CORRECT
C               LOCATION IN THE WINDOW.
C      XFACTOR - SCALE FACTOR FOR INDEPENDENT DATA.  CALCULATED
C               SO AS TO MAKE THE DATA RANGE FILL A 1200 PIXEL
C               WIDE WINDOW.
C      YFACTOR - SCALE FACTOR FOR DEPENDENT DATA.  Y DATA RANGE
C               WILL PRESENTLY FIT IN WINDOW 800 PIXELS HIGH.
C      OFFX    - SAME AS OFFSETX
C      OFFY    - SAME AS OFFSETY
C
C
C      SUBROUTINES OR SUBFUNCTIONS CALLED: NONE
C
C      VERSION: 5-16-89 (LATEST UPDATE)
C
C      WRITTEN BY: STONE
C
C *****************************************************************
C
```

```
C
      SUBROUTINE POINTPLOT(NPTS,XMAX,YMAX,XMIN,YMIN,X,Y,TITLE,
     1 OFFSETX,OFFSETY,XFACTOR,YFACTOR,OFFX,OFFY,XTITLE,YTITLE,MULTPLOT)
      DIMENSION X(NPTS),Y(NPTS)
C
      COMMON TMASS(5000),P(5000),RHO(5000),VEL(5000),DMDT(5000)
      COMMON T(5000),CMASS(5000),CRHO(5000),CPRES(5000)
      COMMON THRUST(5000),AT(5000),DISP(5000)
      COMMON FGAUSS(5000),DELTA(5000),FTHRUST(5000)
      COMMON DELTASTACK(5000),VDAC(5000),ATEFF(5000)
C
      INTEGER*2 LEY,REY,OFFSETX,OFFSETY,OFFX1,OFFX2,SIGNX,SIGNY
      INTEGER*2 OFFY1,OFFY2,XSIZE,YSIZE,OFFX,OFFY,XTLOC,YTLOC,YTLOCX
      CHARACTER*2 XDIV(3),XDIVN(3),YDIV(3),YDIVN(3)
      INTEGER*2 TITLE(40),XTITLE(19),YTITLE(19)
      INTEGER*2 INTXDIV(3),INTXDIVN(3),INTYDIV(3),INTYDIVN(3)
      INTEGER*2 SX(1001),SY(1001),JX,JL,JXN,JLN,JY,JYN
      INTEGER*2 SHORTX,SHORTY
      INTEGER*2 IDX(16),IDY(16)
      REAL*8 XMAX,XMIN,YMAX,YMIN,XRANGE,YRANGE,XFACTOR,YFACTOR
      DATA IDX/0,1,0,-1,0,1,1,0,0,1,0,0,0,-1,-1,-1/,
     1      IDY/0,0,1,0,1,0,0,-1,-1,0,1,1,1,0,0,0/
      EQUIVALENCE (XDIV,INTXDIV)
      EQUIVALENCE (XDIVN,INTXDIVN)
      EQUIVALENCE (YDIV,INTYDIV)
      EQUIVALENCE (YDIVN,INTYDIVN)
C
C  JUMP OVER THIS INITIAL SCALING IF THE CURRENT PLOT IS TO BE
C  OVERLAYED ON A PREVIOUSLY GENERATED SCREEN IMAGE
C
      IF(MULTPLOT.EQ.1) GOTO 6742
C
C  NEW PLOT, RE-INIIALIZE THE PLOT PARAMETERS
C
C  SET PROGRAM VARIABLES
C
      XSIZE = 1200        ! USEABLE PIXEL COLUMNS FOR PLOT WINDOW
      YSIZE = 800     ! USEABLE PIXEL ROWS FOR PLOT WINDOW
      RATIO = 1.0
C
C  SCALE THE DATA AND LOAD IT INTO INTEGER VECTORS
C
      Q=0.0
      IF(YMIN.GT.Q.AND.YMAX.GT.Q) YMIN=0.0 !ENSURE THE ORIGIN IS INCLUDED
      IF(XMIN.GT.Q.AND.XMAX.GT.Q) XMIN=0.0
      IF(XMIN.LT.Q.AND.XMAX.LT.Q) XMAX=0.0
      IF(YMIN.LT.Q.AND.YMAX.LT.Q) YMAX=0.0
C
      XRANGE=(XMAX-XMIN)      ! RANGE OF EXPERIMENTAL DATA: REAL
      TYPE *,’XMAX=’,XMAX
      TYPE *,’XMIN=’,XMIN
      TYPE *,’XRANGE=’,XRANGE
      YRANGE=(YMAX-YMIN)
      TYPE *,’YMAX=’,YMAX
      TYPE *,’YMIN=’,YMIN
      type *,’YRANGE=’,YRANGE
C
C      CALCULATE CONVERSION FACTORS FROM REAL WORLD UNITS TO SCREEN
C      COORDINATES (PIXELS)
C      UNITS = PIXELS PER REAL WORLD UNIT E.G. 200 PIXELS/POUND ETC.
C
      XFACTOR=XSIZE/(RATIO*XRANGE) !SCALING FACTOR FROM WORLD TO
      YFACTOR=YSIZE/(RATIO*YRANGE) !SCREEN COORDINATES
C
C  SET OFFSET VALUES
C
C      OFFSET = THE NUMBER OF PIXELS A PARTICULAR AXIS IS TO BE
C              MOVED AWAY FROM DEAD CENTER OF THE SCREEN.
C              SIGN CONVENTION:
C                  OFFSETX (POSITIVE): MOVE Y-AXIS LEFT OF CENTER SCREEN
C                  OFFSETX (NEGATIVE): MOVE Y-AXIS RIGHT OF CENTER SCREEN
C                  OFFSETY (POSITIVE): MOVE X-AXIS DOWN FROM CENTER SCREEN
C                  OFFSETY (NEGATIVE): MOVE X-AXIS UP FROM CENTER SCREEN
C
C      NOTE: IIDNNT (REAL*8 TO INTEGER*2); WAS IFIX PREVIOUSLY
      OFFSETX = IIDNNT((XMIN+XRANGE/2.)*XFACTOR)
      OFFSETY = IIDNNT((YMIN+YRANGE/2.)*YFACTOR)-80 !Y VALUES SHIFTED UP
C
6742   CONTINUE
C
C  CONVERT REAL-WORLD DATA POINTS TO SCREEN COORDINATES (INTEGER PIXEL NO.)
C
      DO 200 I=1,NPTS
      SX(I)=IIDNNT(X(I)*XFACTOR)-OFFSETX
      SY(I)=IIDNNT(Y(I)*YFACTOR)-OFFSETY
C      SX(I)=IFIX(X(I)*XFACTOR)-OFFSETX
C      SY(I)=IFIX(Y(I)*YFACTOR)-OFFSETY
200    CONTINUE
C
C  JUMP OVER AXIS PLOTTING IF THE CURRENT PLOT IS TO BE
C  OVERLAYED ON A PREVIOUSLY GENERATED SCREEN IMAGE
C
      IF(MULTPLOT.EQ.1) GOTO 6743
C
C  DRAW AXES
C
      CALL VAL8(30) !WHITE
      OFFX=OFFSETX       !Y-AXIS SCREEN OFFSET POSITION
      OFFY=OFFSETY    !X-AXIS SCREEN POSITION
C      TYPE *,’OFFX=’,OFFX,’ OFFY=’,OFFY
      CALL MOVABS(-OFFX,-YSIZE/2+80)       !MOVE POINTER TO BOTTOM OF Y-AXIS
      CALL DRWABS(-OFFX,YSIZE/2+80)        !DRAW Y-AXIS
      CALL MOVABS(-XSIZE/2,-OFFY)       !MOVE POINTER TO LEFT SIDE X-AXIS
      CALL DRWABS(XSIZE/2,-OFFY)        !DRAW X-AXIS
C
6743   CONTINUE
C
C  PLOT DATA POINTS AS CONNECTED LINES (WAS PREVIOUSLY DONE AS CIRCLES ONLY)
```

```
C
      CALL VAL8(15)               !RED LINES AND CIRCLES
      CALL PRMFIL(1)              !TURN THE PERIMETER FILL ROUTINE ON (RASTECH)
      CALL MOVABS(SX(1),SY(1))    !JUMP TO FIRST DATA POINT
      DO 127 I=1,NPTS-1
C      CALL MOVABS(OFFSETX,OFFSETY) !MOVE TO ORIGIN [OBSOLETE]
      CALL DRWABS(SX(I+1),SY(I+1)) !DRAW THE NEXT LINE SEGMENT
C      CALL MOVABS(SX(I),SY(I))     !NOW MOVE TO LOCATION OF EACH DATA POINT
      CALL CIRCLE(1)    !DRAW A CIRCLE (RADIUS 1 PIXELS) AT THE DATA POINT
127    CONTINUE
C
C  JUMP OVER LABELLING ROUTINES IF THE CURRENT PLOT IS TO BE
C  OVERLAYED ON A PREVIOUSLY GENERATED SCREEN IMAGE
C
      IF(MULTPLOT.EQ.1) GOTO 851          !DUMP THE PLOT BUFFER AND RETURN
C
C  LABLE LEGEND (FROM EXPERIMENTAL DATA FILE MASTHEAD)
C
      CALL MOVABS(-350,488)        !MOVE TO
      CALL TEXTC(40,0)       !SET THE LETTER SIZE (PIXELS)
      CALL VAL8(11)               !SET THE COLOR
C
C  BOLD TYPE
C
      DO 400 I = 1,9                    !BEGIN BOLD LOOP
      CALL MOVREL(IDX(I),IDY(I))       !SHIFTS THE PIXELS TO BLUR THE LETTERS
      CALL TEXT1(80,TITLE)              !RASTECH 1/80 WRITES THE MESSAGE OUT
 400   CONTINUE                   !TO SCREEN (50 CHARACTERS MAX)
C
C      PLOT X-AXIS TITLE
C
C      TYPE *,'X-AXIS TITLE?'
C      READ(5,220)XTITLE       !GET THE X-AXIS TITLE FROM THE USER
C220   FORMAT(19A2)
C
C      ACCOUNT FOR NEGATIVE OFFSET IN X DIRECTION
C
      XTLOC=400-OFFX                    !X LOCATION, IF OFFSET POS.
      IF(OFFX.LT.0) XTLOC=-OFFX-600   !X LOCATION, IF OFFSET NEG.
      CALL MOVABS(XTLOC,(-OFFY-50))       !MOVE THERE (50 PIXELS BELOW THE LINE)
      CALL TEXTC(30,0)               !TEXT WILL BE 30 PIXELS HIGH
      CALL VAL8(11)                    !SET THE COLOR
      DO 401 I=1,4                   !BEGIN BOLD LOOP
      CALL MOVREL(IDX(I),IDY(I))       !SMEAR THE TEXT BY MOVING RELATIVE
      CALL TEXT1(38,XTITLE)             !PRINT THE TITLE ON THE SCREEN
401    CONTINUE
C
C      PLOT Y-AXIS TITLE
C
C      TYPE *,'Y-AXIS TITLE?'
C      READ(5,220)YTITLE              !GET TITLE FROM USER
C
C      CALCULATE VERTICAL LOCATION OF Y-AXIS TITLE
C
```

```
      YTLOC=300-OFFY                    !MOVE 300 PIXELS UP THE AXIS TO
C                          CENTER TITLE (IF OFFSETY IS POS)
      IF(OFFY.LT.0) YTLOC=-OFFY-600   !ACCOUNT FOR OFFY BEING NEGATIVE
C
C      CALCULATE HORIZONTAL LOCATION OF Y-AXIS TITLE
C
      YTLOCX=-OFFX-26               !MOVE 26 PIXELS TO RIGHT OF AXIS
      IF(OFFX.LT.0) YTLOCX=30-OFFX    !IF NEGATIVE, MOVE 30 PIXELS LEFT
      CALL MOVABS(YTLOCX,YTLOC)       !MOVE THERE
      CALL TEXTC(30,90)              !SET TEXT SIZE IN PIXELS
      CALL VAL8(11)                  !SET THE COLOR
      DO 6003 N=1,4                  !BEGIN BOLD LOOP
      CALL MOVREL(IDX(N),IDY(N))      !MOVE RELATIVE TO PRESENT LOCATION
      CALL TEXT1(38,YTITLE)            !PRINT THE TITLE ON THE SCREEN
6003   CONTINUE
C
C      CALCULATE NUMBER OF TICK MARKS (SPACED AT 100 PIXELS) THAT
C      WILL GO ON EACH AXIS (LONG AND SHORT SECTIONS ARE INCLUDED IF
C      IT EXTENDS INTO MORE THAN ONE QUADRANT)
C
      SHIFT=-80
      IF(OFFY.LT.0) SHIFT=+80
      LONGX=(XSIZE/2)+ABS(OFFX)       !NO. PIXELS ON LONG SIDE X-AXIS
      SHORTX=(XSIZE/2)-ABS(OFFX)      !NO. PIXELS ON SHORT SIDE X-AXIS
      LONGY=(YSIZE/2)+ABS(OFFY)-SHIFT !NO. PIXELS ON LONG SIDE Y-AXIS
      SHORTY=YSIZE/2-ABS(OFFY)+SHIFT  !NO. PIXELS ON SHORT SIDE Y-AXIS
C
C      FIND THE NUMBER OF TICK MARKS TO BE INCLUDED
C
      NLONGX=(LONGX/100)       !TICKS ARE SPACED AT 100 PIXELS
      NSHORTX=(SHORTX/100)
      NLONGY=(LONGY/100)
      NSHORTY=(SHORTY/100)
C
C      DRAW TICK MARKS FOR X-AXIS (LONG SIDE)
C
      CALL TEXTC(25,0)       !TELL THE RASTECH TO READY THE DISPLAY
      DO 350 L=1,NLONGX       !LOOP ON THE NUMBER OF TICKMARKS
      SIGNX=1                 !INITIALIZE THE SIGN FLAG (IF NEGATIVE,
C                        !WE MOVE LEFT OF ORIGIN & LABELS HAVE
C                        !NEGATIVE SIGN
      IF(OFFX.LE.0) SIGNX=-1
C
C      NOTE: IF BOTH SIDES HAVE IDENTICAL RANGE, PLOT THE LEFT (NEGATIVE)
C            SIDE FIRST (HENCE THE .LE. CHECK)
C
      JX=SIGNX*L*100-OFFX     !DETERMINE LOCATION OF NEXT TICK
      CALL MOVABS(JX,-8-OFFY) !MOVE THERE
      CALL DRWABS(JX,-OFFY)   !DRAW THE TICK
C
C      LABEL TICK VALUES FOR X-AXIS (LONG SIDE)
C
      XSCALE=XFACTOR             !REDUNDANT LABEL: CONVERSION FACTOR
C                          FROM REAL WORLD TO SCREEN COORDINATES
```

```
      YSCALE=YFACTOR              !REDUNDANT LABEL:
      XL=L                        !REM: L IS NUMBER OF TICK MARKS LONG SIDE
      XTIC=(100./XSCALE)*XL*SIGNX  !GET REAL VALUE AT NEXT TICK LOCATION
      ENCODE(11,351,XDIV)XTIC       !ENCODE IT FOR THE  RASTERTECH
351   FORMAT(G11.4)
      JL=-15. +SIGNX*L*100-OFFX    !FIND LOCATION OF NEXT LABEL
      CALL MOVABS(JL,-20-OFFY)     !MOVE THERE
      DO 353 N=1,4                 !BEGIN BOLD SMEARING LOOP
      CALL MOVREL(IDX(N),IDY(N))   !MOVE RELATIVE TO PRESENT LOCATION
      CALL TEXT1(11,INTXDIV)       !PRINT THE TEXT ON THE SCREEN
353   CONTINUE
350   CONTINUE
C
C     DRAW TICK MARKS FOR X-AXIS (SHORT SIDE)
C
      DO 850 L=1,NSHORTX      !L IS NOW NUMBER OF TICKS SHORT SIDE
      JXN=-100*SIGNX*L-OFFXC  !DETERMINE THE LOCATION OF THE NEXT TICK
      CALL MOVABS(JXN,-8-OFFY)!MOVE THERE
      CALL DRWABS(JXN,-OFFY)  !DRAW IT
C
C     LABEL TICK VALUES FOR X-AXIS (SHORT SIDE)
C
      XL=L                         !REM: L IS LOOP INDEX FOR NUMBER TICKS
      XTICN=-1.*(100./XSCALE)*XL*SIGNX  !DETERMINE THE VALUE AT NEXT TICK
      ENCODE(11,351,XDIVN)XTICN       !ENCODE IT FOR DISPLAY (RASTECH)
C                               !LOCATE VALUE 15 PIXELS LEFT OF TICK
      JLN=-1.*SIGNX*(L*100)-15-OFFX
      CALL MOVABS(JLN,-20-OFFY)       !MOVE THERE
      DO 6001 N=1,4                 !BEGIN BOLD SMEARING LOOP
      CALL MOVREL(IDX(N),IDY(N))       !MOVE LOCATION RELATIVE TO CURRENT POS
      CALL TEXT1(11,INTXDIVN)            !PRINT THE TEXT STRING
6001  CONTINUE
850   CONTINUE
C
C     DRAW TICK MARKS FOR Y-AXIS (LONG SIDE)
C
      CALL TEXTC(25,90)             !SET THE CHARACTER HEIGHT 25 PIXELS
      DO 360 L=1,NLONGY               !LOOP FOR NUMBER OF TICK MARKS
      SIGNY=1                       !INITIALIZE SIGN FLAG
      IF(OFFY.LE.0) SIGNY=-1             !MAKE IT NEGATIVE OF OFFSET IS NEG.
      JY=SIGNY*L*100-OFFY             !CALCULATE LOCATION OF NEXT TICK
      CALL MOVABS(-OFFX,JY)             !MOVE THERE
      CALL DRWABS(8-OFFX,JY)              !DRAW IT
C
C     LABEL TIC VALUES FOR Y-AXIS (LONG SIDE)
C
      YL=L                         !INDEX NUMBER FOR CURRENT TICK
      YTIC=(100/YSCALE)*YL*SIGNY       !CALCULATE THE VALUE AT THE TICK
      ENCODE(11,352,YDIV)YTIC          !ENCODE IT FOR RASTERTECH
352   FORMAT(G11.4)
      JL=-40+ SIGNY*L*100-OFFY         !LOCATION IS 40 PIXELS RIGHT OF AXIS
C     JLN=-1*SIGNY*(L*100+15)-OFFY              !CALCULATE LOCATION OF LABEL
      CALL MOVABS(-OFFX-10,JL)       !MOVE THERE
      DO 354 N=1,4                 !BEGIN BOLD LOOP
      CALL MOVREL(IDX(N),IDY(N))       !SMEAR RELATIVE TO PRESENT LOCATION
      CALL TEXT1(11,INTYDIV)          !PRINT THE TEXT TO THE SCREEN
354   CONTINUE
360   CONTINUE
C
C     DRAW TICK MARKS FOR Y-AXIS (SHORT SIDE)
C
      DO 851 L=1,NSHORTY              !LOOP ON NUMBER OF TICK MARKS
      JYN=-100*SIGNY*L-OFFY             !CALCULATE CURRENT TICK LOCATION
      CALL MOVABS(-OFFX,JYN)              !MOVE THERE
      CALL DRWABS(8-OFFX,JYN)              !DRAW IT
C
C     LABEL TICK VALUES FOR Y-AXIS (SHORT SIDE)
C
      YL=L                            !CURRENT TICK INDEX NO.
      YTICN=-1.*(100./YSCALE)*YL*SIGNY       !CALCULATE VALUE THERE
      ENCODE(11,352,YDIVN)YTICN            !ENCODE IT FOR RASTERTECH
      JLN= -40-1*SIGNY*(L*100)-OFFY           !CALCULATE LOCATION OF LABEL
C     JLN=-1*SIGNY*(L*100+15)-OFFY            !CALCULATE LOCATION OF LABEL
      CALL MOVABS(-OFFX-10,JLN)            !MOVE THERE
      DO 6002 K=1,4                   !BEGIN BOLD LOOP
      CALL MOVREL(IDX(K),IDY(K))            !SMEAR THE LABEL RELATIVE
      CALL TEXT1(11,INTYDIVN)               !PRINT LABEL TO SCREEN
6002  CONTINUE
851   CONTINUE
      CALL EMPTYB
C
C     OPTIONAL: WRITE THIS PLOT DATA TO DISK FILE FOR SUBSEQUENT
C             ANALYSIS (E.G. FFT FREQUENCY ANALYSI VIA EQPROCESSOR)
C
      TYPE *,'DO YOU WANT TO WRITE THIS PLOT FILE TO DISK?'
      TYPE *,'[1=YES; 2=NO]'
      READ *,NHNS
      IF(NHNS.EQ.1)THEN
        CALL OUTPUT(NPTS,X,Y)
        ENDIF
      RETURN
      END
```

```
C**********************************************************************
C
      SUBROUTINE OUTPUT(NPTS,X,Y)
C
C     PURPOSE: WRITES PLOT DATA TO AN EXTERNAL FILE
C              FOR SUBSEQUENT FREQUENCY ANALYSIS BY THE PROGRAM
C              EQPROCESSOR.
C
C     IMPLEMENTATION:  CALL OUTPUT(NPTS,X,Y)
C
C     OTHER ROUTINES CALLED: SORT (FOR SORTING DATA PRIOR TO DETERMINING
C                            MAXIMA AND MINIMA
C
C     WRITTEN BY: STONE
C     VERSION   : 7-07-89
C
C************************************************************************
C
      CHARACTER*20 DATATITLE
      CHARACTER*25 NEWFILE
      CHARACTER*80 TITLE
      DIMENSION X(10000),Y(10000),X1(10000),Y1(10000)
      INTEGER*2 NPTS
C
C    GET TITLES FROM USER
C
      TYPE *,’ENTER A FILENAME FOR DISK STORAGE’
    TYPE *,’[NO EXTENSIONS, 20 CHARACTERS MAX]’
      READ (5,19)DATATITLE
19    FORMAT(A20)
      TYPE *,’TITLE FOR DATA SET’
    TYPE *,’[80 CHARACTERS MAX]’
      READ (5,1)TITLE
1     FORMAT(A80)
      TYPE *,’TOTAL NUMBER OF DATA POINTS =’,NPTS
    TYPE *,’IS THIS CORRECT? 1=YES;2=NO ’
    READ *,NTNS
    IF(NTNS.EQ.2)THEN
     TYPE *,’ENTER CORRECT VALUE’
     READ *,NPTS
     ENDIF
C
C     SORT X,Y PAIRS FOR DETERMINATION OF MAXIMA AND MINIMA
C
      DO 104 I=1,NPTS
      X1(I)=X(I)
      Y1(I)=Y(I)
104   CONTINUE
      CALL SORT(X,Y,NPTS)
      XMAX=X(NPTS)
      XMIN=X(1)
      CALL SORT(Y1,X1,NPTS)
      YMAX=Y1(NPTS)
      YMIN=Y1(1)
```

```
C       YMAX=Y(NPTS)
C       YMIN=Y(1)
C
C       WRITE DATA TO DISK FILE
C
        NEWFILE=DATATITLE//’.NBS’
        OPEN (UNIT=3,FILE=NEWFILE,ACCESS=’SEQUENTIAL’,FORM=’FORMATTED’,
     1 STATUS=’NEW’)
        REWIND 3
        WRITE(3,9)TITLE
9       FORMAT(A80)
        WRITE(3,5)XMIN,XMAX,YMIN,YMAX,NPTS
5       FORMAT(4(E12.5,3X),I5)
        DO 105 I=1,NPTS
        WRITE(3,6) X(I),Y(I)
6       FORMAT(2(E12.5,3X))
105     CONTINUE
        CLOSE(UNIT=3)
        RETURN
        END
```

Appendix B
Program: HDYN

Transient Response of HSILS Subjected
to Random Voltage-Time Histories

```
c*******************************************************
c
c  program: HDYN
c
c  purpose: allows real-time graphics solution of generalized
c           transient dynamic analysis of single degree of freedom
c           systems with or without damping and/or time varying
c           forcing function. The program is written for a Silicon
c           Graphics workstation and allows for real-time user
c           modification of key parameters such as stiffness and
c           damping by means of a dial box.  Dial box assignments
c           are listed below.
c
c           The capability also exists for superimposing
c           the results of experimental data over a theoretical
c           response to the same force-time history and allows for
c           real-time y-ordinate scaling, x-axis shifting (time)
c           and examination of individual data point values.
c
c           HDYN was developed for real time analysis of the
c           transient reponse of the NIST High Speed Intelligent
c           Loading System (HSILS).
c
c  revision history:
c
c  a) 11-14-91: version 1.0 written by B.Stone
c
c  external subroutines called: dderkf.f
c
c  linkage requirements:
c
c  "cc stone1.f -lfgl -lgl_s -lm -o stone1 dderkf.o" to compile and link
c
c   Notes: a) Press the "Esc"[ape] key to exit.
c          b) Dial0 is the lower left knob on the dial box
c             and is used to vary damping ratio (%)
c          c) Dial1 is the lower right knob on the dial box
c             and is used to vary stiffness.
c
c******************************************************************
c
      IMPLICIT REAL*8 (A-H,O-Z)
      real*8 DATA1(2500),DATA2(2500),DATA3(2500)
      real*8 DATA4(2500),ACCEL(4000),data6(4000)
      real*8 olddata1(2500),olddata2(2500)
      real*8 lastdamp,laststiff
      real*8 signal(4000)

      integer*4 size(2)
      integer*2 mdev(8), mval(8), lastval(8),npointer,npointer_last
      integer dev,init,min,max

      integer*2 time(2500),displacement(2500),velocity(2500)
      integer*2 time2(2500),signal_pix(2500)
      integer*2 oldtime(2500),olddisplacement(2500),oldvelocity(2500)
      character*2 fxchar_last(12),fychar_last(12)

      COMMON /SYSTEMP/ X0,X0PRIME,BETA,CEE,OMEGAN,SMASS,FREQ
      COMMON /CONTROL/ NPTS,DT,TTIME,NTOTAL,IPRINT,IRESP,M
      COMMON /EARTHQ/ ACCEL,NACCEL,ADT,TIMEREC,AG
      common /plotstuff/ data1,data2,data3

      common/sensordata/ signal,nsignalpts,sig_time_step,data6

      EXTERNAL FCN

#include <gl/fgl.h>                ! fortran graphics library constants
#include <gl/fdevice.h>             ! fortran device constants
c
c-----------------------------------------------------------------------------
c
c     Initialize transient dynamic problem parameters:
c
      PI = 4.0*ATAN(1.0)

      CALL XSETF(-1)                                 !disable error interupts
      CALL XSETUN(10)                              !dump error messages to file

      M=2500                                         !initial dial maxima
      call initialize
      damp      = cee               !assign the initial damping ratio (C(system)/C(critical), decimal value)
      stiffness = ((2*pi*freq)**2)*smass  !assign initial stiffness
      ichange = 0                             !reset the replot flag
c
c
c-----------------------------------------------------------------------------
c
c     initialize and open the window:
c
      call prefpo(10,XMAXSC-10,300,YMAXSC-30)
      winid = winope('SDOF1-SGI',9)            ! give it the name "SDOF1-DGI"
      call qdevic(ESCKEY)                     ! queue the escape key for later hit
                                         ! detection
      call color(BLUE)                        ! set the background window color to blue
      call clear                              ! flood the window with the background
                                         !color
      call getsiz(size(1),size(2))              ! read the window size (its fixed here,
                                         ! but
                                         ! in later programs it may be variable)
c
c-----------------------------------------------------------------------------
c
c     graphics setups before entering the plotting routine:
c
      toffset = 0
      npointer = 1                  !initialize the data feedback crosshairs at point number 1

      init =  0                    !initial value of dials
      min  = -5000                   !highest negative ticks allowed for dials
      max  =  5000                   !highest positive tick number allowed for dials
      call setval(DIAL0,init,min,max)  !initialize dial0
      call setval(DIAL1,init,min,max)  !initialize dial1
      call setval(DIAL2,init,min,max)  !initialize dial2
      call setval(DIAL3,init,min,max)  !initialize dial3
      call setval(DIAL4,init,min,max)  !initialize dial4

      mdev(1) = DIAL0                  ! damping adjust
      mdev(2) = DIAL1                  ! stiffness adjust
      mdev(3) = DIAL2                  ! display cursor movement (numeric feedback from current position)
      mdev(4) = DIAL3                  ! vertical scale factor for sensor data
      mdev(5) = DIAL4                  ! horizontal shift for sensor data (allows for overlapping and fitting)
      mdev(6) = DIAL5                  ! unused
      mdev(7) = DIAL6                  ! unused
      mdev(8) = DIAL7                  ! unused

      call getdev(8,mdev,mval)         ! read the current values for DIAL0, DIAL1, DIAL2, etc.

      lastval(1) = mval(1)            ! save current value for later comparison
      lastval(2) = mval(2)            !
      lastval(3) = mval(3)            !
      lastval(4) = mval(4)            !
      lastval(5) = mval(5)            !
                                  !
      call legend(size)              !indicate on screen the type of problem being solved
                                  !(either free vibration or forced)
c
c-----------------------------------------------------------------------
c
c     solve the initial problem:
c
        call sdof1
c
c       scale the new data to fit within window bounds:
c
        call rescale(npts,data1,data2,time,displacement,
     1                   size,xfactor,yfactor)
c
c       plot the theoretical displacement-time data:
c
        ncolor = 7                              ! outline white
        call plotpdelta(npts,time,displacement,ncolor,size,
     1                 npointer,npointer_last)
c
c-------------
c
c       overlay the experimental acoustic sensor data
c
        yfactor_sensor = yfactor   !temporary assignment
        call scale(npts,data6,signal,time2,signal_pix,
     1              size,xfactor,yfactor_sensor)

c
c       overlay the acoustic experimental signal on the predicted displacement record
c
```

```
      ncolor=3
      call plotpdelta(npts,time2,signal_pix,ncolor,size,
     1                npointer,npointer_last)

c
c-------------
c
c     update the displayed values of stiffness and damping:
c
      call drwtxt(size,npointer)
c
c     store current values:
c
      do i=1,npts
        oldtime(i) = time(i)                    ! new time series
        olddisplacement(i) = displacement(i)        ! new displacements
      enddo
      lastdamp  = damp                          ! save current value of damping ratio
      laststiff = stiffness                   ! save current value of stiffness
      lastval(1) = mval(1)                      ! damping
      lastval(2) = mval(2)                      ! stiffness
      lastval(3) = mval(3)                      ! feedback cursor
      lastval(4) = mval(4)                      ! y-scaling of sensor data
      lastval(5) = mval(5)                      ! x-translation of sensor data

      ichange = 0                             ! reset the replot flag

c-------------------------------------------------------------------------------
c
c
c    window manager check loop
c
25    continue

c
c    Read devices:
c
     dev = qtest()
     if (dev.ne.0) then
        dev = qread(val)
     endif
     if(dev .eq. ESCKEY) go to 1090

     call reshap
     call getdev(8,mdev,mval)                ! read the current values of
c
c-------------------------------------------------------------------------
c
c    Adjust Damping via Dial 0:
c
     if(mval(1).eq.lastval(1)) then
        goto 999                                ! no change in damping; leave value displayed
     else
        damp = damp + (mval(1)-lastval(1))*0.001           ! assign new damping value
        cee  = damp                               ! reload common parameter with current damping value
```

```
          ichange = 1                                    ! flag to plot routine to update transient reponse plot
        endif
999   continue
c
c--------------------------------
c
c     Adjust Stiffness via Dial 1:
c
        if(mval(2).eq.lastval(2)) then                    ! no change in Dial 1
          goto 998
        else
          stiffness = stiffness + (mval(2) - lastval(2))*10.   ! assign new stiffness value
          omegan = sqrt(stiffness/smass)           ! reload common parameter with current natural frequency
          freq = omegan/(2*pi)
          ichange = 1                                    ! flag to plot routine to update transient response plot
        endif
998   continue
c
c--------------------------------
c
c     Adjust location of cursor (for feedback of numerical quantities) via Dial 2
c
        if(mval(3).eq.lastval(3)) then                    ! no change in Dial 1
          goto 997
        else
          npointer = npointer + (mval(3)-lastval(3))           ! assign new pointer
          ichange = 1                                    ! flag to plot routine to update transient response plot
        endif
        if(npointer.lt.1) then
          npointer = 1
        elseif(npointer.gt.npts) then
          npointer = npts
        endif
997   continue
c
c--------------------------------
c
c     Adjust vertical scaling of sensor data via Dial 3
c
        if(mval(4).eq.lastval(4)) then                    ! no change in Dial 1
          goto 996
        else
          yfactor_sensor = yfactor_sensor*(1+ (mval(4)-lastval(4))*0.01)
          ichange = 1                                    ! flag to plot routine to update transient response plot
        endif
996   continue
c
c--------------------------------
c
c
c     Adjust horizontal position of sensor data via dial 4
c
        if(mval(5).eq.lastval(5)) then                    ! no change in Dial 1
          goto 995
        else
        toffset =  (mval(5)-lastval(5))          ! assign new pointer
        do kkl = 1,4000
          data6(kkl) = data6(kkl) + toffset*0.000020     !offset in 20 microsecond increments
        enddo

        ichange = 1                                  ! flag to plot routine to update transient response plot
      endif
995   continue

c
c--------------------------------------------------------------------------------
c
c     If a change has occurred in either Damping or Stiffness has occurred, then
c     re-solve the transient dynamic problem and update the plot.
c
      if(ichange.eq.1) then
c
c       resolve the transient problem
c
        call sdof1
c
c       erase the old plot
c
        ncolor = 4                               ! erase, background blue
        call plotpdelta(npts,time,displacement,ncolor,size,
     1                npointer,npointer_last)
c
c       scale the new data to the original scaling factors
c       (need to do it this way to indicate changes in amplitude):
c
        call scale(npts,data1,data2,time,displacement,
     1                 size,xfactor,yfactor)
c
c       plot the new data:
c
        ncolor = 7                               ! outline white
        call plotpdelta(npts,time,displacement,ncolor,size,
     1                npointer,npointer_last)
c
c------------
c
c       overlay the experimental acoustic/pressure sensor data
c
c       erase the last plot

        ncolor=4
        call plotpdelta(npts,time2,signal_pix,ncolor,size,
     1                npointer,npointer_last)
c
c       determine the current vertical scaling factor for sensor data:
c
c       note that we will use the same x-scaling factor as the displacement data to keep
c       the time bases in sync.

        call scale(npts,data6,signal,time2,signal_pix,
     1              size,xfactor,yfactor_sensor)
c
c       overlay the acoustic experimental signal on the predicted displacement record
c
        ncolor=2
        call plotpdelta(npts,time2,signal_pix,ncolor,size,
     1                 npointer,npointer_last)
c
c------------
c       update the displayed values of stiffness and damping:
c
        call drwtxt(size,npointer)
c
c       store current values:
c
        do i=1,npts
          oldtime(i) = time(i)                        ! new time series
          olddisplacement(i) = displacement(i)          ! new displacements
        enddo
        lastdamp  = damp                    ! save current value of damping ratio
        laststiff = stiffness             ! save current value of stiffness
        lastval(1) = mval(1)                ! save current value of DIAL0 (damping)
        lastval(2) = mval(2)                ! save current value of DIAL1 (stiffness)
        lastval(3) = mval(3)                ! save current value of DIAL2 (cursor position)
        lastval(4) = mval(4)                ! save current value of DIAL3 (y-scaling of sensor data)
        lastval(5) = mval(5)                ! save current value of DIAL3 (x-translation of sensor data)
        ichange = 0                        ! reset the replot flag
        npointer_last = npointer

      endif
      go to 25
1090  continue

      call gexit
      stop
      end
c------------------------------end main graphics loop--------------------
```

```
c------------------------------------------
c
      subroutine drwtxt(size,npointer)
c
c     purpose: plots screen text; updates pointer values
c
c     version: 12/91  B.Stone
c     subroutines called: none
c
c------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      real*8 DATA1(2500),DATA2(2500),DATA3(2500)
      real*8 DATA4(2500),ACCEL(4000)
      real*8 olddata1(2500),olddata2(2500)
      real*8 lastdamp,laststiff

      character*2 fxchar(12),fxchar_last(12),fychar(12),fychar_last(12)
      character*2 fzchar(12),fwchar(12),gxchar(12),gychar(12)
      integer*2 npointer
      integer*4 xstart,ystart,xend,yend
      integer*4 xtext,vspc,ytext1,ytext2,ytext3,ytext4
      integer*4 size(2),ystart2,yend2,y2txt1,y2txt2

      COMMON /SYSTEMP/ X0,X0PRIME,BETA,CEE,OMEGAN,SMASS,FREQ
      COMMON /CONTROL/ NPTS,DT,TTIME,NTOTAL,IPRINT,IRESP,M
      COMMON /EARTHQ/ ACCEL,NACCEL,ADT,TIMEREC,AG
      common /plotstuff/ data1,data2,data3
      EXTERNAL FCN

#include <gl/fgl.h>
c
c------------------------------------------------------
c
c     erase the old values of damping, stiffness, frequency, and mass
c
      xtext  = size(1)-305              !beginning x-coord for titles
      vspc   = 15                       !vertical line spacing for titles
      ytext1 = 5                        !starting y-position for first line of text
      ytext2 = ytext1+vspc                !second line of text etc.
      ytext3 = ytext2+vspc
      ytext4 = ytext3+vspc

      xstart = xtext+195                 !x-coordinate lower left corner of box to erase values
      ystart = ytext1                   !y-coordinate of lower left corner of box to erase values
      xend   = xstart+110                 !x-coordinate of top right corner
      yend   = 4*vspc+ytext1               !y-coordinate of top right corner

      ystart2=yend + 20
      yend2  =ystart2+50
      y2txt1 =ystart2+10
      y2txt2 = y2txt1+vspc

      call color(BLUE)                   !background color, erase
      call rectfi(xstart,ystart,xend,yend)   !draw the box, fill with blue
c
```

```
c     draw an outlined box around the legend area
c
      call color(YELLOW)
      call rectfi(xtext-5,ytext1-5,xend,yend)
      call color(BLACK)
      call recti(xtext-5,ytext1-5,xend,yend)

      call color(BLUE)                        !background color, erase
      call rectfi(xstart,ystart2,xend,yend2)  !draw the box, fill with blue
c
c     draw an outlined box around the legend area
c
      call color(YELLOW)
      call rectfi(xtext-5,y2txt1-5,xend,yend2)
      call color(BLACK)
      call recti(xtext-5,y2txt1-5,xend,yend2)
c
c-------------------------------------------------------------------------------------------
c
c     begin routine to display in the window the current values of DIAL0 and DIAL1.
c     Note that this procedure allows any mathematical modification of the
c     values read from the devices prior to writing the value the screen.  Thus, in IDENT,
c     the dials could be used to scale the parameters alpha, beta, and gamma, pass those
c     values to the analytical program, plot the results, and simultaneously plot the
c     current values of alpha, beta, and gamma on the screen.
c
c     encode the current dial positions (in screen coordinates, not raw values)
c
      PI = 4.0*ATAN(1.0)
      stiffness = (omegan**2)*smass  !assign initial stiffness

      encode(12,300,fxchar)cee               ! convert damping value to character string
      encode(12,300,fychar)stiffness         ! convert stiffness value to character string
      encode(12,300,fzchar)smass              ! convert system mass value to character string
      encode(12,300,fwchar)omegan/(2*PI)      ! convert nat.frequency (hertz) into character string
      encode(12,300,gxchar)data1(npointer)   ! time at cursor position
      encode(12,300,gychar)data2(npointer)   ! displacement at cursor position

300   format(g12.5)
c
c     now plot the x and y positions with appropriate information text
c
      call cmov2i(xtext,ytext2)                         ! move to the location to  print out the x-value
      call charst('Damping Ratio      = ',20)    ! x-title
      call cmov2i(xstart,ytext2)                ! move to the fixed position for plotting x-data
      call charst(fxchar,12)                   ! write out the x-position
      call cmov2i(xtext,ytext1)                 ! move the location to print out the y-value
      call charst('Stiffness (lb/in) = ',20)    ! y-title
      call cmov2i(xstart,ytext1)                ! move to the fixed position for plotting y-data
      call charst(fychar,12)                   ! write out the y-position

      call cmov2i(xtext,ytext3)                 ! move the location to print out the y-value
      call charst('Mass (lb-s^2/in)  = ',20)     ! y-title
      call cmov2i(xstart,ytext3)                ! move to the fixed position for plotting y-data
      call charst(fzchar,12)                   ! write out the y-position
      call cmov2i(xtext,ytext4)             ! move the location to print out the y-value
      call charst('Nat. Freq. (Hz)   = ',20)   ! y-title
      call cmov2i(xstart,ytext4)            ! move to the fixed position for plotting y-data
      call charst(fwchar,12)                ! write out the y-position

      call cmov2i(xtext,y2txt1)             ! move the location to print out the y-value
      call charst('Time (seconds)    = ',20)   ! y-title
      call cmov2i(xstart,y2txt1)            ! move to the fixed position for plotting y-data
      call charst(gxchar,12)                ! write out the y-position

      call cmov2i(xtext,y2txt2)             ! move the location to print out the y-value
      call charst('Displacement (in) = ',20)   ! y-title
      call cmov2i(xstart,y2txt2)            ! move to the fixed position for plotting y-data
      call charst(gychar,12)                ! write out the y-position
c
      return
      end
```

```
c-----------------------------------------
c
      subroutine legend(size)
c
c     purpose: plots the legend box containing the type of
c              dynamic system being presently solved.
c
c     version: 12/91 B. Stone
c
c     subroutines called: none
c
c-----------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      real*8 DATA1(2500),DATA2(2500),DATA3(2500)
      real*8 DATA4(2500),ACCEL(4000)
      real*8 olddata1(2500),olddata2(2500)
      real*8 lastdamp,laststiff

      character*2 fxchar(12),fxchar_last(12),fychar(12),fychar_last(12)
      character*2 fzchar(12),fwchar(12)
      integer*4 xstart,ystart,xend,yend
      integer*4 xtext,ytext
      integer*4 size(2)

      COMMON /SYSTEMP/ X0,X0PRIME,BETA,CEE,OMEGAN,SMASS,FREQ
      COMMON /CONTROL/ NPTS,DT,TTIME,NTOTAL,IPRINT,IRESP,M
      COMMON /EARTHQ/ ACCEL,NACCEL,ADT,TIMEREC,AG
      common /plotstuff/ data1,data2,data3
      EXTERNAL FCN

#include <gl/fgl.h>
c
c-------------------------------------------------------
c
c     erase the old values of damping, stiffness, frequency, and mass
c
      xstart = size(1)-320          !x-coordinate lower left corner of box to erase values
      ystart = size(2)-60           !y-coordinate of lower left corner of box to erase values
      xend   = size(1)-10           !x-coordinate of top right corner
      yend   = size(2)-30           !y-coordinate of top right corner
      xtext  = size(1)-310          !beginning x-coord for titles
      ytext  = ystart+10             !starting y-position for first line of text
c
c     draw an outlined box around the legend area
c
      call color(YELLOW)
      call rectfi(xstart,ystart,xend,yend)
      call color(BLACK)
      call recti(xstart,ystart,xend,yend)

c---------------------------------------------------------------------------------
c     now plot the x and y positions with appropriate information text
c
      call cmov2i(xtext,ytext)              ! move to the location to  print out the x-value
      if(iresp.eq.1) then
```

```
        call charst('Solving:  Mx** + Cx* + Kx = 0',29)       ! x-title
      elseif(iresp.eq.2) then
        call charst('Solving:  Mx** + Cx* + Kx = P(t)',32)
      endif
c
      return
      end
```

```
c-----------------------------------------------------
c
      subroutine plotpdelta(kptstop,delta,p,ncolor,size,
     1                    npointer,npointer_last)
c
c     Purpose: plots values of two vectors (displacement, load) between the low index "kptsbottom"
c              and the high index "kptstop" on
c              the screen using color "ncolor".  If the data goes outside of screen coordinates
c              an advisory message is plotted in the message box and the offending vector entry
c              is not plotted; the advisory indicates that the rescale button should be picked.
c
c
c     Revision History:  1) 10-25-90  Initial code development       B. Stone
c                        2) 11-13-90  Code documentation added       B. Stone
c
c     Subroutines called: none
c
c-------------------------------------------------------------------------

#include <gl/fgl.h>                ! fortran graphics library constants
#include <gl/fdevice.h>            ! fortran device constants

      integer*2 p(2500),delta(2500),ptrad1,ptrad2
      integer*2 ptrad3
      integer*2 npointer_last,npointer
      integer*2 newpos(2),oldpos(2),beginpos(2),cursorpos(2)
      integer*4 kptstop,xsize,ysize,kptsbottom
      integer*4 xcenter,ycenter,size(2)
      integer*4 xaxis,yaxis,nwid
      integer flag
      character*80 message
c
c         constant definitions:
c
      ptrad1 = 5
      ptrad2 = 3
      ptrad3 = 7
      kptsbottom = 1
      xsize = size(1)
      ysize = size(2)
      xaxis = xsize-100
      yaxis = (ysize/2)-100
c
c     plot the axes
c
      nwid=4
      call linewi(nwid)
      call color(YELLOW)
      xcenter= size(1)/20.          !(xsize-200)/2            !screen x-center
      ycenter= size(2)/2.           !(ysize-200)/2 + 200       !screeb y-center

      call move2i(xcenter,ycenter)
      call rdr2i(xaxis,0)             !(xsize-200)/2-100,0)     !positive x-axis
      call move2i(xcenter,ycenter)
      call rdr2i(0,yaxis)             !(ysize-200)/2 -100)      !positive y-axis
      call move2i(xcenter,ycenter)
      call rdr2i(0,-yaxis)              ! 100 - (ysize-200)/2)    !negative y-axis
c
c     draw the line using the current value of kptstop
c
      oldpos(1)=delta(kptsbottom)         !set the initial "from" x-coordinate
      oldpos(2)=p(kptsbottom)             !set the initial "from" y-coordinate

      call color(ncolor)          !set the line color to that passed from the calling routine
c
c     plot the hysteresis loop: each loop cycle plots one line between successive experimental
c     data points

      nwid = 2
      call linewi(nwid)
      do 832 i=kptsbottom,kptstop-1          !notice that only the data between kptsbottom
                                             !and kptstop is plotted.
        newpos(1)=delta(i+1)        !set the "to" x-coordinate
        newpos(2)=p(i+1)            !set the "to" y-coordinate

        call bgnlin              !begin a line entity
        call v2s(oldpos)          !move to the "from" vertex
        call v2s(newpos)           !and draw a line to the "to" vertex
        call endlin              !close this graphics line entity

        oldpos(1)=newpos(1)        !set the current "from" x-coordinate to the previous "to" x-coordinate
        oldpos(2)=newpos(2)        !set the current "from" y-coordinate to the previous "to" y-coordinate

832   continue
c -------------------------------
c
c     plot the data feedback cursor:
c
      if(ncolor.eq.4) then
        call color(BLUE)
        cursorpos(1) = delta(npointer_last)
        cursorpos(2) = p(npointer_last)
c
c         filled white circle, full diameter, if ncolor =7; otherwise blue
c
          call circfs(cursorpos(1),cursorpos(2),ptrad3)
      elseif (ncolor.ne.4) then
        cursorpos(1) = delta(npointer)
        cursorpos(2) = p(npointer)
        call color(YELLOW)                  !outline black circle, full diameter
        call circs(cursorpos(1),cursorpos(2),ptrad1)
        call color(RED)
        call circfs(cursorpos(1),cursorpos(2),ptrad2)   !red center dot
      endif
      return
      end
```

```
c----------------------------------------------------------------
c
      subroutine rescale(kptstop,floatdelta,floatp,delta,p,
     1                    size,xfactor,yfactor)
c
c      Purpose: rescales  load & displacement (either theoretical or experimental) data to fit the
c              the full screen dimensions.  The active variable is KPTS, the number of active points
c              as set by DIAL0.  If KPTS = NPTS (original number of experimental data points, then
c              no change takes place; otherwise, the maxima and minima for the reduced number of
c              active points are determined and the displacements and loads are scaled to fit the screen.
c
c
c    Revision History:  1) 11-06-90  Initial code development       B. Stone
c                       2) 11-13-90  Code documentation added       B. Stone
c
c    Subroutines called: none
c
c
c---------------------------------------------------------------------------
        IMPLICIT REAL*8 (A-H,O-Z)
      real*8 floatdelta(2500),floatp(2500)
        integer*2 p(2500),delta(2500)
      integer*4 size(2),xcenter,ycenter
        integer xsize, ysize, npts,mxsize,mysize,kptstop
      COMMON /SYSTEMP/ X0,X0PRIME,BETA,CEE,OMEGAN,SMASS,FREQ

#include <gl/fgl.h>                   ! fortran graphics library constants
#include <gl/fdevice.h>               ! fortran device constants

C-----------------------------------------------------------
c
c      set the screen size (must fit into the fixed plotting window)
c
      xcenter= size(1)/20.            !(xsize-200)/2              !screen x-center
      ycenter= size(2)/2.             !(ysize-200)/2 + 200          !screeb y-center
C
C       ZERO THE MAXIMA AND MINIMA COMPARITORS
C
      XMAX=0.0
      XMIN=0.0
      YMAX=0.0
      YMIN=0.0

      XMAX1=0.0
      XMIN1=0.0
      YMAX1=0.0
      YMIN1=0.0
C
C        DETERMINE MAXIMA AND MINIMA
C
        DO 1001 I=1,kptstop

        IF(floatdelta(I).GT.XMAX) XMAX=floatdelta(I)
        IF(floatdelta(I).LT.XMIN) XMIN=floatdelta(I)
        IF(floatp(I).GT.YMAX) YMAX=floatp(I)
        IF(floatp(I).LT.YMIN) YMIN=floatp(I)

1001   CONTINUE
c
c      compute real data range
c
        xrange = amax1(abs(xmin),abs(xmax))      !take larger of xmax or xmin; range equals twice this
        yrange = amax1(abs(ymin),abs(ymax))*2.   !since we assume xmin to be negative; assymetric graphs
c
c      determine screen scaling factors
c
        xfactor = (real(size(1))/xrange)*.8       !will therefore always get displayed in all quadrants
        yfactor = (real(size(2))/yrange)*.8       !scaling down factor (arbitrary)
c
c      re-compute the screen coordinates of the p, and delta vectors
c      using the new values of xfactor and yfactor
c
        do 100 i = 1, kptstop                     !kptstop = npts

        delta(i) = floatdelta(i)*xfactor + xcenter     !offset to the center of the plot window
        p(i) = floatp(i)*yfactor + ycenter            !offset to the center of the plot window
100     continue

      return
        end
```

```
c-------------------------------------------------------
c
      subroutine scale(kptstop,floatdelta,floatp,delta,p,
     1                 size,xfactor,yfactor)
c
c      Purpose: rescales  load & displacement (either theoretical or experimental) data to fit the
c              the full screen dimensions.  The active variable is KPTS, the number of active points
c              as set by DIAL0.  If KPTS = NPTS (original number of experimental data points, then
c              no change takes place; otherwise, the maxima and minima for the reduced number of
c              active points are determined and the displacements and loads are scaled to fit the screen.
c
c
c     Revision History:  1) 11-06-90  Initial code development      B. Stone
c                        2) 11-13-90  Code documentation added      B. Stone
c
c     Subroutines called: none
c
c
c--------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      real*8 floatdelta(2500),floatp(2500)
        integer*2 p(2500),delta(2500)
      integer*4 size(2),xcenter,ycenter
        integer xsize, ysize, npts,mxsize,mysize,kptstop
      COMMON /SYSTEMP/ X0,X0PRIME,BETA,CEE,OMEGAN,SMASS,FREQ

#include <gl/fgl.h>                ! fortran graphics library constants
#include <gl/fdevice.h>             ! fortran device constants

C---------------------------------------------------------
c
c      set the screen size (must fit into the fixed plotting window)
c
      xcenter= size(1)/20.            !(xsize-200)/2            !screen x-center
      ycenter= size(2)/2.             !(ysize-200)/2 + 200        !screeb y-center
c
c      scale the theoretical data using the current window scaling factors xfactor, and yfactor
c
c
c      re-compute the screen coordinates of the p, and delta vectors
c      using the new values of xfactor and yfactor
c
        do 100 i = 1, kptstop                        !kptstop = npts

        delta(i) = floatdelta(i)*xfactor + xcenter      !offset to the center of the plot window
        p(i) = floatp(i)*yfactor + ycenter            !offset to the center of the plot window
100     continue

      return
        end
c
```

```
c---------------------------------------------------------------
c
   Subroutine sdof1
c
c  Response of an SDOF to various forcing functions, solved using
c  the DEPAC subroutine DDERKF (NIST - out of Sandia National
c  Laboratory)
c
c  Original Development: 87-88  T.Shenton
c  Solution basedon IMSL subroutine DVERK- restricted to VAX use
c
c  Program Re-written : 11-7-91  B.Stone
c  Solution based on DEPAC subroutine DDERKF - FORTRAN source available
c  suitable for many machines (uncomment "IMACH" and "DIMACH" subroutine
c  lines for machine dependent constants). Should be portable to PC.
c
c*****************************************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 DATA1(2500),DATA2(2500),DATA3(2500),DATA4(2500),
     *ACCEL(4000)
      COMMON /SYSTEMP/ X0,X0PRIME,BETA,CEE,OMEGAN,SMASS,FREQ
      COMMON /CONTROL/ NPTS,DT,TTIME,NTOTAL,IPRINT,IRESP,M
      COMMON /EARTHQ/ ACCEL,NACCEL,ADT,TIMEREC,AG
      EXTERNAL FCN
C
      OPEN(UNIT=5,STATUS='UNKNOWN')
      OPEN(UNIT=6,STATUS='UNKNOWN')
C
C      M = ARRAY DIMENSION OF MAIN DATA ARRAYS, MUST CORRESPOND TO ACTUAL
C      DIMENSION OF ARRAYS AND CONTROLS MAX TIME OF CALCULATION
      M    = 2500
C
      CALL STEPPER              !solve the transient second order ODE

      return
      END
```

```
c-----------------------------------------------------------------
c
      SUBROUTINE STEPPER
c
c         Purpose: time step solution of differential equations of motion
c
c         Version:          1988 original code by T.Shenton
c                           1991 re-written by B.Stone
c
c       subroutines called: dderkf
c
c--------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 DATA1(2500),DATA2(2500),DATA3(2500),DATA4(2500)
      REAL*8 T,TEND,X(2),XOLD(2),RWORK(50),RTOL,RPAR(1)
      INTEGER N,INFO(15),IDID,LRW,LIW,IWORK(50),IPAR(1)

      COMMON /SYSTEMP/ X0,X0PRIME,BETA,CEE,OMEGAN,SMASS,FREQ
      COMMON /CONTROL/ NPTS,DT,TTIME,NTOTAL,IPRINT,IRESP,M
      common /plotstuff/ data1,data2,data3

      EXTERNAL FCN
c
c     Assign Constant Values:
c
      PI      = 4.0*ATAN(1.0)
      N       = 2
      T       = 0.0
      TEND    = 0.0
      X(1)    = X0
      X(2)    = X0PRIME
      INFO(1) = 0
      INFO(2) = 0
      INFO(3) = 0
      LRW     = 50
      LIW     = 50
      RTOL    = 0.001        !how many digits of accuracy you want
      ATOL    = 0.001        !this defines zero resolution; pick carefully depending
      DATA1(1) = T           !on the specific problem, anything computed smaller than this
      DATA2(1) = X0          !is considered to be zero.
      DATA3(1) = X0PRIME
C
C STEP THROUGH CALCULATION
c
       DO 100 I = 1,NTOTAL
       TEND = TEND + DT
       CALL DDERKF(FCN,N,T,X,TEND,INFO,RTOL,ATOL,IDID,
     * RWORK,LRW,IWORK,LIW,RPAR,IPAR)
       IF(IDID.LT.0) THEN

         IF(IDID .EQ. -1) THEN
           INFO(1) = 1
         ELSEIF(IDID .EQ. -5) THEN
           INFO(1) = 1
         ELSEIF(IDID .EQ. -4) THEN
```

```
          INFO(1) = 1
        ELSE
          WRITE(6,45)IDID
        ENDIF

      ENDIF
c      IF(IPRINT  .EQ. 1)THEN
c         WRITE(6,40)T,X(1),X(2)
c      ENDIF
C
C VECTORS OF TIME, DISP AND VELOCITY
      DATA1(I+1) = T                   !TIME (SECONDS)
      DATA2(I+1) = X(1)                 !DISPLACEMENT (INCH)
      DATA3(I+1) = X(2)                 !VELOCITY (INCH/SECOND)
100   CONTINUE
C
C WRITE OUT VECTORS TO DISK

c*****
c*****
c
c Deactivated 11-13-91 Stone
c
c      CALL WDATA(DATA1,DATA2,DATA3,DATA4,M,NTOTAL+1,3)
c
c*****
c*****
40    FORMAT(3(5X,G12.5))
45    FORMAT(/,5X,'*** ERROR *** IDID = ',I5)
50    FORMAT(/,5X,'**ERROR INVALID RESPONSE TYPE *** ')
     END
```

```
c----------------------------------------
c
      SUBROUTINE INITIALIZE
c
c
c     Purpose: reads in initial constants for problem setup
c
c     Version: 1) original            T.Shenton
c              2) re-written 11-14-91  B.Stone
c
c     Subroutines called: none
c
c     Parameters:
c
c     iresp      == solution type; 0 = free vibration; 1 = forced
c     x0         == initial displacement
c     x0prime    == initial velocity
c     freq       == system natural frequency (hertz) = omegan/(2*pi) == (sqrt(k/m))/(2*pi)
c     smass      == system mass
c     cee        == damping ratio
c     dt         == time step (seconds) for solution
c     ttime      == total solution time (seconds)
c     timerec    == total length (seconds) of forcing function (iresp=1 only)
c     ntotal     == total number of points in solution
c     m          == limiting vector dimension for solution (presently set to 2500 points)
c
c-------------------------------------------------------------------------------
c
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER ANS*1

      REAL*8 ACCEL(4000)
      real*8 signal(4000),data6(4000)

      COMMON /SYSTEMP/ X0,X0PRIME,BETA,CEE,OMEGAN,SMASS,FREQ
      COMMON /CONTROL/ NPTS,DT,TTIME,NTOTAL,IPRINT,IRESP,M
      COMMON /EARTHQ/ ACCEL,NACCEL,ADT,TIMEREC,AG
      common/sensordata/ signal,nsignalpts,sig_time_step,data6
      PI = 4.0*ATAN(1.0)
      OMEGAN = 0.0
      BETA = 0.0
C
      WRITE(6,5)
      READ(5,*)IRESP
C
      IF(IRESP .EQ. 1) THEN
C READ IN INITIAL CONDITIONS
        WRITE(6,10)
        READ(5,*)X0,X0PRIME
C READ IN NATURAL FREQ IN HERTZ
        WRITE(6,22)
        READ(5,*)FREQ
        OMEGAN = 2.*PI*FREQ
C READ IN SYSTEM MASS
        WRITE(6,21)
        READ(5,*)SMASS
C READ IN DAMPING RATIO
        WRITE(6,15)
        READ(5,*)CEE
C READ IN CALCULATION TIME STEP DELTA T
        WRITE(6,26)
        READ(5,*)DT
C READ IN TOTAL TIME OF CALCULATION
        WRITE(6,30)
        READ(5,*)TTIME
      ELSEIF (IRESP .EQ. 2) THEN
        WRITE(6,10)
        READ(5,*)X0,X0PRIME
        WRITE(6,22)
        READ(5,*)FREQ
        OMEGAN = 2.*PI*FREQ
        WRITE(6,21)
        READ(5,*)SMASS
        WRITE(6,15)
        READ(5,*)CEE
        CALL FORCEREAD          !get the voltage-time history
        call responseread       !get the acoustic feedback signal
        WRITE(6,50)TIMEREC
        WRITE(6,26)
        READ(5,*)DT
        WRITE(6,30)
        READ(5,*)TTIME
      ENDIF
      WRITE(6,35)
      READ(5,36)ANS
      IF(ANS .EQ. ’Y’ .OR. ANS .EQ. ’y’)THEN
        IPRINT = 1
      ELSE
        IPRINT = 0
      ENDIF
      NTOTAL = INT(TTIME/DT)+2
      npts=ntotal+1
      IF(NTOTAL .GT. M-1)THEN
        WRITE(6,55)(M-1)*DT
        NTOTAL = M-1
      ENDIF
      RETURN
5     FORMAT(/,5X,’RESPONSE MENU: ’,/,
     *7X,’FREE VIBRATION                    (1)’,/,
     *7X,’ARBITRARY FORCED RESPONSE         (2)’,/,
     *5X,’ENTER CHOICE : ’,$)
10    FORMAT(/,5X,’ENTER INITIAL CONDITIONS, X0 AND X0PRIME  : ’,$)
15    FORMAT(5X,  ’ENTER DAMPING RATIO                       : ’,$)
20    FORMAT(5X,  ’ENTER EXCITATION FREQUENCY                : ’,$)
21    FORMAT(5X,  ’ENTER SYSTEM MASS                         : ’,$)
22    FORMAT(5X,  ’ENTER NATURAL FREQUENCY (HERTZ)           : ’,$)
23    FORMAT(5X,  ’ENTER ACCELERATION SCALE FACTOR           : ’,$)
25    FORMAT(5X,  ’ENTER # PTS IN CYCLE OF HIGHEST FREQ.     : ’,$)
26    FORMAT(5X,  ’ENTER DELTA T                             : ’,$)
30    FORMAT(5X,  ’ENTER TOTAL TIME TO COMPUTE RESPONSE      : ’,$)
```

```
35   FORMAT(5X,  'VIEW THE RESULTS (Y/N) [N]                  : ',$)
36   FORMAT(A1)
50   FORMAT(  5X,'TOTAL TAU OF RECORD           = ',F10.3)
55   FORMAT(/,5X,'ARRAY SPACE HAS LIMITED TIME TO : ',F10.5)
     END
```

```
c------------------------------------------------
c
      SUBROUTINE FCN(T,X,XPRIME,RPAR,IPAR)
c
c     Purpose: Defines a system of FIRST ORDER ordinary differential
c              equations (ODE's) which will be used to evaluate the
c              displacement and velocity as a function of time
c
c     Version:    original    1988     T.Shenton
c                 re-written  11/15/91 B.Stone
c                 [includes embedded piezoelectric characteristic equation]
c
c     Parameters:
c
c     T         == current time (seconds)
c     X         == vector of independent variables
c                  (in this case the displacement and velocity)
c     XPRIME    == vector of first derivatives of the independent variables
c                  (in this case: velocity and acceleration)
c     CEE       == damping ratio (constant, set in INITIALIZE)
c     OMEGAN    == natural frequency of the system (radians/second) = sqrt(K/M) = 2*pi*freq ;
c                  freq =  hertz
c     SMASS     == system mass
c     FREQ      == natural frequency of the system (hertz)
c     F         == applied force
c                  *********
c                    note that in the special case of the piezoelectric stack this has been modified
c                    to read the VOLTAGE at time T used to drive the stack
c                  *********
c     X0        == initial displacement
c     X0PRIME   == initial velocity
c
c     Subroutines Called: Force(t,f)
c
c---------------------------------------------------------------------------------------
c
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER IPAR(1)
      REAL*8 X(2),XPRIME(2),T,RPAR(1)
      COMMON /SYSTEMP/ X0,X0PRIME,BETA,CEE,OMEGAN,SMASS,FREQ
c
c     determine the current force through interpolation of the
c     force-time history read from disk file.  In this case, force
c     is actually the drive voltage to the piezoelectric
c     stack;  The characteristic equation for the stack is used
c     (see below) to determine the force applied at a given voltage.
c
      CALL FORCE(T,F)
c
c     set up the equations of motion (note that DDERKF solves FIRST ORDER
c     systems of equations.  A second order ODE can be solved by two simultaneous
c     first order equations.
c
c     For our general case, Mx" + Cx' + Kx = F(t)
c
```

```
c     this can be broken down into two simultaneous first order ODE's as follows:
c
c     we let x1  = x
c        let x2  = x1' = x'
c
c       then x2' = x1" = x" = -(K/M)x - (C/M)x' + F(t)/M
c
c     we then substitude the definitions for x1 and x2:
c
c            x2' = -(K/M)x1 - (C/M)x2 + F(t)/M
c
c     but, in this analysis we ask the user for damping ratio, which is given by
c
c     cee = C/(2*M*omegan) ; where omegan**2 = K/M and M = the variable SMASS
C
C      solving:
c
c     C = 2*M*omegan*CEE ;   and (K/M) = omegan**2;  substitution of these yields:
c
c*********
c*********
c
c     11-15-91: UN-comment the following two equations to solve the general
c               SDOF dynamics problem.
c
c      XPRIME(1) = X(2)
c      XPRIME(2) = -OMEGAN**2*X(1) - 2.*CEE*OMEGAN*X(2) + F/SMASS
c
c
c********
c********
c
c     For the specific problem of the Piezoelectrically driven
c     system, the forcing function is given by:
c
c     F(t) = -375000.*x + 35.94*E(t)^0.6597
c
c     this is determined on by the blocked Force/Displacement/Voltage
c     characteristics of the specific piezoelectric stack and will change
c     for each type of material.  If we choose to read in the VOLTAGE - TIME
c     history of the drive pulse sent to the piezoelectric stack in the
c     variable name "F", the solution then becomes:
c
      xprime(1) = x(2)
      xprime(2) = -omegan**2*x(1)                       !stiffness contribution
     1            -2.*cee*omegan*x(2)              !damping contribution
     2            +(-375000.*x(1)+35.94*F**0.6597)/SMASS  !forcing function (F = drive voltage)
c                                            !x = piezoelectric stack expansion (displacement)
c
      RETURN
      END
      SUBROUTINE FORCE(T,F)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 ACCEL(4000)
      COMMON /CONTROL/ NPTS,DT,TTIME,NTOTAL,IPRINT,IRESP,M
      COMMON /SYSTEMP/ X0,X0PRIME,BETA,CEE,OMEGAN,SMASS,FREQ
      COMMON /EARTHQ/ ACCEL,NACCEL,ADT,TIMEREC,AG
      IF (IRESP .EQ. 1) THEN
c
c        free vibration response has no forcing function:
c
         F = 0.0
      ELSEIF (IRESP .EQ. 2) THEN
c
c        determine applied force for this time step.  Note that
c        if the solution time step does not coincide with the time
c        step in the forcing function record then an interpolated
c        value of the current force is needed:
c
         IN = INT(T/ADT) + 1
         INP1 = IN + 1
         AN = ACCEL(IN)
         ANP1 = ACCEL(INP1)
         TN = (IN-1)*ADT
         F = (ANP1-AN)*(T-TN)/ADT + AN     !interpolated current force
      ENDIF
      RETURN
      END
```

```
c-----------------------------------------------------
c
      SUBROUTINE FORCEREAD
c
c      purpose: reads force (acceleration) vector from a
c             disk file containing a header line which
c             describes the number of data points and the
c             time step (assumed constant) in seconds
c             between the data points.
c
c      version: 12/91 B. Stone
c
c      subroutines called: none
c
c------------------------------

      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER FILENAME*80,REC*80
      REAL*8 ACCEL(4000)
      COMMON /EARTHQ/ ACCEL,NACCEL,ADT,TIMEREC,AG
C
c      initialize the force vector
c
      do i=1,4000
        accel(i) = 0.0
      enddo
c
c      get the input filename
c
      WRITE(6,10)
      READ(5,15)FILENAME
      OPEN(UNIT=1,FILE=FILENAME,STATUS='UNKNOWN')
C
C READ IN NUMBER OF DATA POINTS IN TIME HISTORY "NACCEL"
C (MAXIMUM OF 4000 - MAY BE INCREASED)
C AND THE TIME INTERVAL OF FORCE TIME HISTORY "ADT"
C
      READ(1,*)NACCEL,ADT
C
C READ OR COMPUTE TOTAL TIME OF FORCE TIME HISTORY "TIMEREC"
C
      TIMEREC = NACCEL*ADT
C
C READ IN FORCE TIME HISTORY INTO VECTOR ACCEL "ACCEL"
C
      DO 100 I=1,NACCEL
100      READ(1,*)ACCEL(I)
C
10     FORMAT(/,5X,'ENTER FILE WITH FORCE TIME HISTORY : ',$)
15     FORMAT(A80)
      CLOSE(UNIT=1)
      RETURN
      END
```

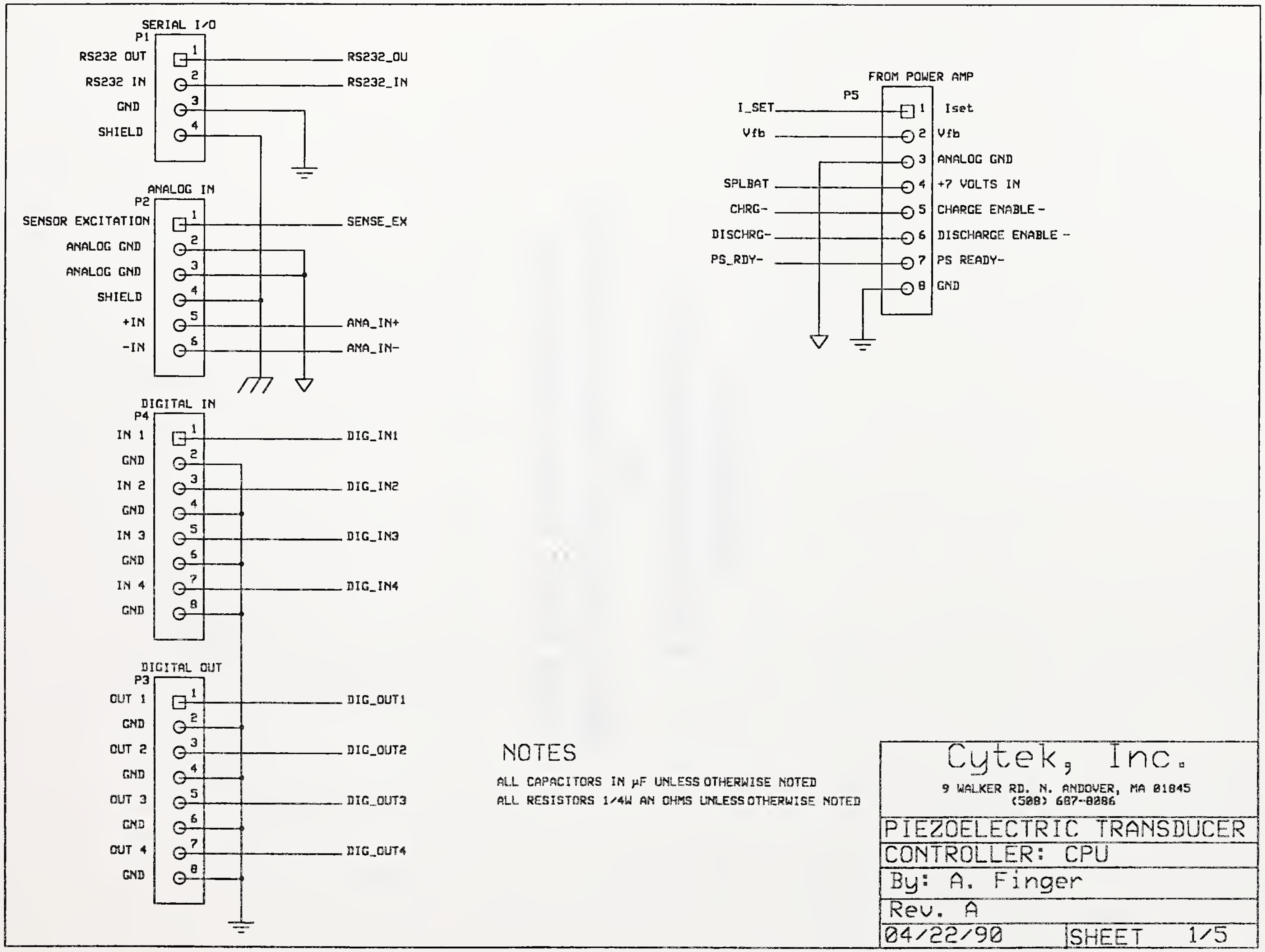
```
c--------------------------------------------------
c
      subroutine responseread
c
c     purpose: reads sensor test data representing HSILS
c              response to a drive voltage pulse.  This
c              must be contained in a disk file containing
c              a header line which
c              describes the number of data points and the
c              time step (assumed constant) in seconds
c              between the data points.
c
c     version: 12/91 B. Stone
c
c     subroutines called: none
c
c------------------------------

      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER FILENAME*80,REC*80
      REAL*8 ACCEL(4000),signal(4000),data6(4000)
      COMMON /EARTHQ/ ACCEL,NACCEL,ADT,TIMEREC,AG

      common/sensordata/ signal,nsignalpts,sig_time_step,data6
C
c     initialize the sensor data vector:
c
      do i=1,4000
        signal(i) = 0.0
      enddo
c
c     get the sensor data filename
c
      type *,'Please enter the filename containing'
      type *,'the acoustic sensor data:'
      READ(5,15)FILENAME
      OPEN(UNIT=1,FILE=FILENAME,STATUS='UNKNOWN')
C
C READ IN NUMBER OF DATA POINTS IN TIME HISTORY "NACCEL"
C (MAXIMUM OF 4000 - MAY BE INCREASED)
C AND THE TIME INTERVAL OF FORCE TIME HISTORY "ADT"
C
      READ(1,*)nsignalpts,sig_time_step
C
C READ OR COMPUTE TOTAL TIME OF FORCE TIME HISTORY "TIMEREC"
C
c      TIMEREC = NACCEL*ADT
C
C Read in the sensor data:
C
      DO  I=1,nsignalpts

        READ(1,*)signal(i)
        signal(i) = -1.*signal(i)          !invert the sensor data
        data6(i) = sig_time_step*(i-1)     !elapsed time (seconds)
      enddo
C
15     FORMAT(A80)
      CLOSE(UNIT=1)
      RETURN
      END
```

```
c----------------------------------------------
c
      SUBROUTINE WDATA (DATA1,DATA2,DATA3,DATA4,M,NPTS,NVEC)
c
c     purpose: optional hardcopy printout of results
c
c     version: original: T. Shenton
c
c     subroutines called: none
c--------------------------------
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 DATA1(M),DATA2(M),DATA3(M),DATA4(M)
      CHARACTER ANS*1,FNMOUT*40,SCALES
C
      WRITE(6,5)
      READ(5,10)ANS
      IF (ANS .EQ. 'Y' .OR. ANS .EQ. 'y'.OR. ANS.EQ. ' ') THEN
       WRITE(6,15)
       READ(5,20)FNMOUT
       OPEN(UNIT=3,FILE=FNMOUT,FORM='FORMATTED',STATUS='NEW')
       IF (NVEC.EQ.1)THEN
         DO 100 I=1,NPTS
         WRITE(3,50)DATA1(I)
100        CONTINUE
       ELSE IF (NVEC.EQ.2) THEN
         DO 101 I=1,NPTS
         WRITE(3,51)DATA1(I),DATA2 (I)
101        CONTINUE
       ELSE IF (NVEC.EQ.3) THEN
         DO 102 I=1,NPTS
         WRITE(3,52)DATA1(I),DATA2(I),DATA3(I)
102        CONTINUE
       ELSE IF (NVEC.EQ.4) THEN
         DO 103 I=1,NPTS
         WRITE(3,53)DATA1(I),DATA2 (I),DATA3(I),DATA4(I)
103        CONTINUE
       ENDIF
       CLOSE(UNIT=3)
      ENDIF
C       WRITE(6,25)
C       READ(5,10)SCALES
C       IF (SCALES .EQ. 'Y' .OR. SCALES .EQ. 'y') THEN
C         CALL SCALE(DATA1,M,NVEC,NPTS,4)
C       ENDIF
      RETURN
5     FORMAT(/,5X,'WISH TO WRITE DATA TO FILE ?(Y/N) [Y]: ',$)
10     FORMAT(A1)
15     FORMAT(5X,'ENTER DATA FILE NAME > ',$)
20     FORMAT(A40)
25     FORMAT(5X,'EVALUATE MAX AND MIN (Y/N) [N] : ',$)
50     FORMAT(1(G12.5,1X))
51     FORMAT(2(G12.5,1X))
52     FORMAT(3(G12.5,1X))
53     FORMAT(4(G12.5,1X))
      END
```

```
c-------------------------------
      SUBROUTINE XSETF (KONTRL)
c
c     purpose: sets error message level for messages
c              returned from DDERKF during execution
c              of solution to the ODE.
c
C***BEGIN PROLOGUE  XSETF
C***PURPOSE  Set the error control flag.
C***LIBRARY   SLATEC (XERROR)
C***CATEGORY  R3A
C***TYPE      ALL (XSETF-A)
C***KEYWORDS  ERROR, XERROR
C***AUTHOR  Jones, R. E., (SNLA)
C***DESCRIPTION
C
C     Abstract
C        XSETF sets the error control flag value to KONTRL.
C        (KONTRL is an input parameter only.)
C        The following table shows how each message is treated,
C        depending on the values of KONTRL and LEVEL.  (See XERMSG
C        for description of LEVEL.)
C
C        If KONTRL is zero or negative, no information other than the
C        message itself (including numeric values, if any) will be
C        printed.  If KONTRL is positive, introductory messages,
C        trace-backs, etc., will be printed in addition to the message.
C
C              ABS(KONTRL)
C        LEVEL        0              1              2
C        value
C          2        fatal          fatal          fatal
C
C          1     not printed      printed         fatal
C
C          0     not printed      printed        printed
C
C         -1     not printed      printed        printed
C                                  only           only
C                                  once           once
C
C***REFERENCES  JONES R.E., KAHANER D.K., 'XERROR, THE SLATEC ERROR-
C                 HANDLING PACKAGE', SAND82-0800, SANDIA LABORATORIES,
C                 1982.
C***ROUTINES CALLED  J4SAVE, XERMSG
C***REVISION HISTORY  (YYMMDD)
C   790801  DATE WRITTEN
C   890531  Changed all specific intrinsics to generic.  (WRB)
C   890531  REVISION DATE from Version 3.2
C   891214  Prologue converted to Version 4.0 format.  (BAB)
C   900510  Change call to XERRWV to XERMSG.  (RWC)
C***END PROLOGUE  XSETF
      CHARACTER *8 XERN1
C***FIRST EXECUTABLE STATEMENT  XSETF
      IF (ABS(KONTRL) .GT. 2) THEN
```

```
         WRITE (XERN1, '(I8)') KONTRL
         CALL XERMSG ('SLATEC', 'XSETF',
     *      'INVALID ARGUMENT = ' // XERN1, 1, 2)
         RETURN
      ENDIF
C
      JUNK = J4SAVE(2,KONTRL,.TRUE.)
      RETURN
      END
*DECK XSETUN
      SUBROUTINE XSETUN (IUNIT)
C***BEGIN PROLOGUE  XSETUN
C***PURPOSE  Set output file to which error messages are to be sent.
C***LIBRARY   SLATEC (XERROR)
C***CATEGORY  R3B
C***TYPE      ALL (XSETUN-A)
C***KEYWORDS  ERROR, XERROR
C***AUTHOR  Jones, R. E., (SNLA)
C***DESCRIPTION
C
C     Abstract
C        XSETUN sets the output file to which error messages are to
C        be sent.  Only one file will be used.  See XSETUA for
C        how to declare more than one file.
C
C     Description of Parameter
C      --Input--
C        IUNIT - an input parameter giving the logical unit number
C                to which error messages are to be sent.
C
C***REFERENCES  JONES R.E., KAHANER D.K., 'XERROR, THE SLATEC ERROR-
C                 HANDLING PACKAGE', SAND82-0800, SANDIA LABORATORIES,
C                 1982.
C***ROUTINES CALLED  J4SAVE
C***REVISION HISTORY  (YYMMDD)
C   790801  DATE WRITTEN
C   861211  REVISION DATE from Version 3.2
C   891214  Prologue converted to Version 4.0 format.  (BAB)
C***END PROLOGUE  XSETUN
C***FIRST EXECUTABLE STATEMENT  XSETUN
      JUNK = J4SAVE(3,IUNIT,.TRUE.)
      JUNK = J4SAVE(5,1,.TRUE.)
      RETURN
      END
```

Appendix C
Electrical Schematics
for
HSILS Onboard Microcontroller
and Pulse Power Subsystems

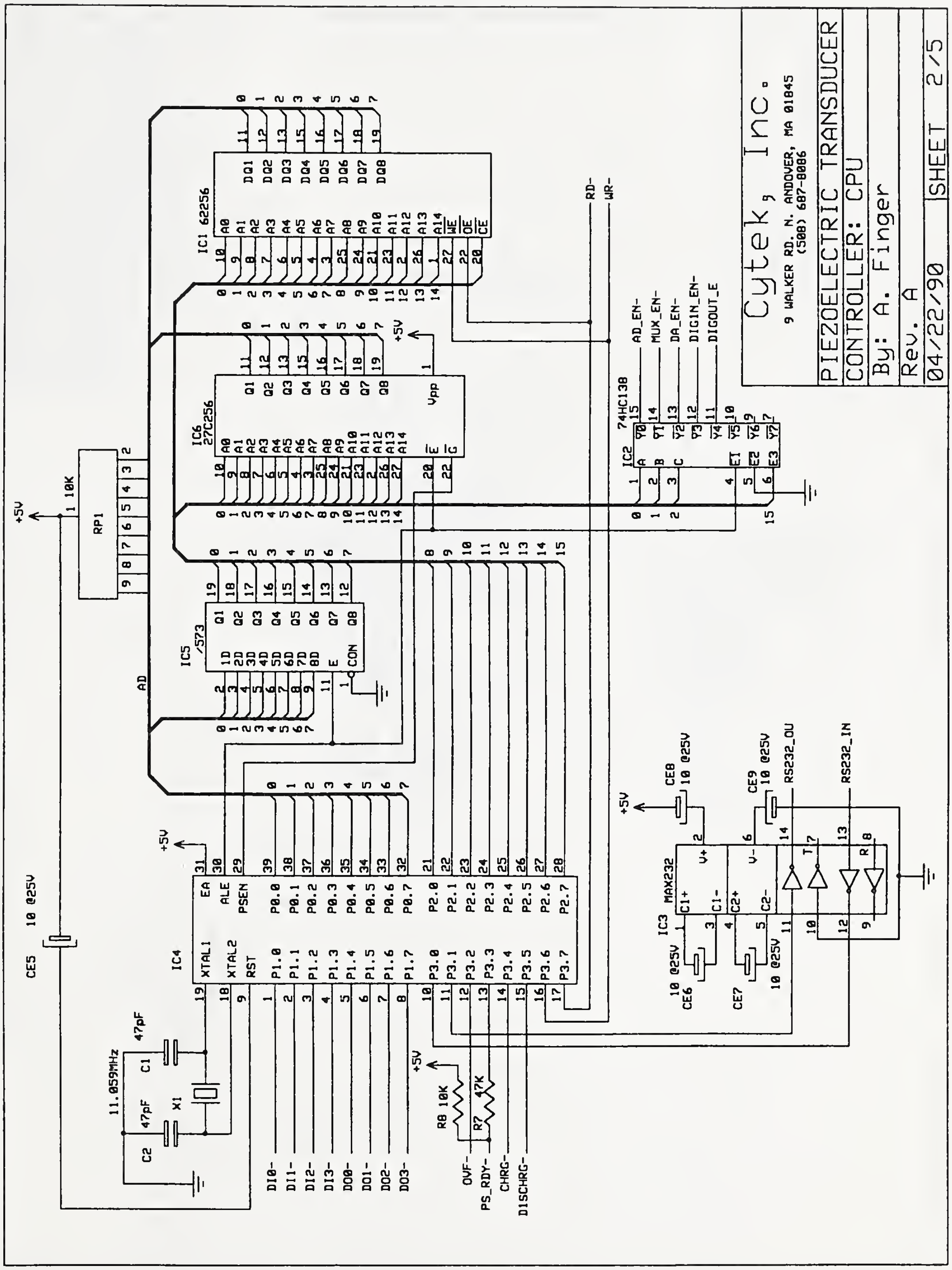


NOTES

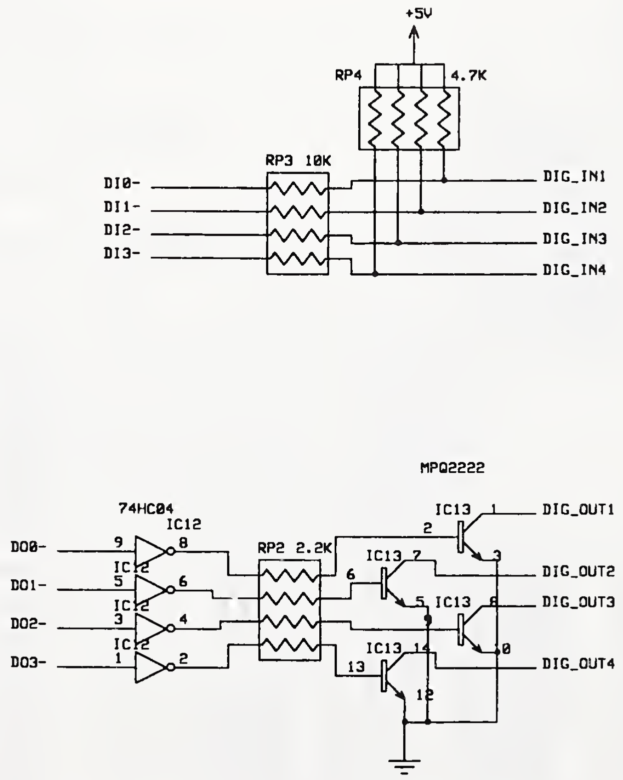
ALL CAPACITORS IN µF UNLESS OTHERWISE NOTED
ALL RESISTORS 1/4W AN OHMS UNLESS OTHERWISE NOTED

Cytek, Inc.
9 WALKER RD. N. ANDOVER, MA 01845
(508) 687-0086

PIEZOELECTRIC TRANSDUCER
CONTROLLER: CPU
By: A. Finger
Rev. A
04/22/90 SHEET 1/5

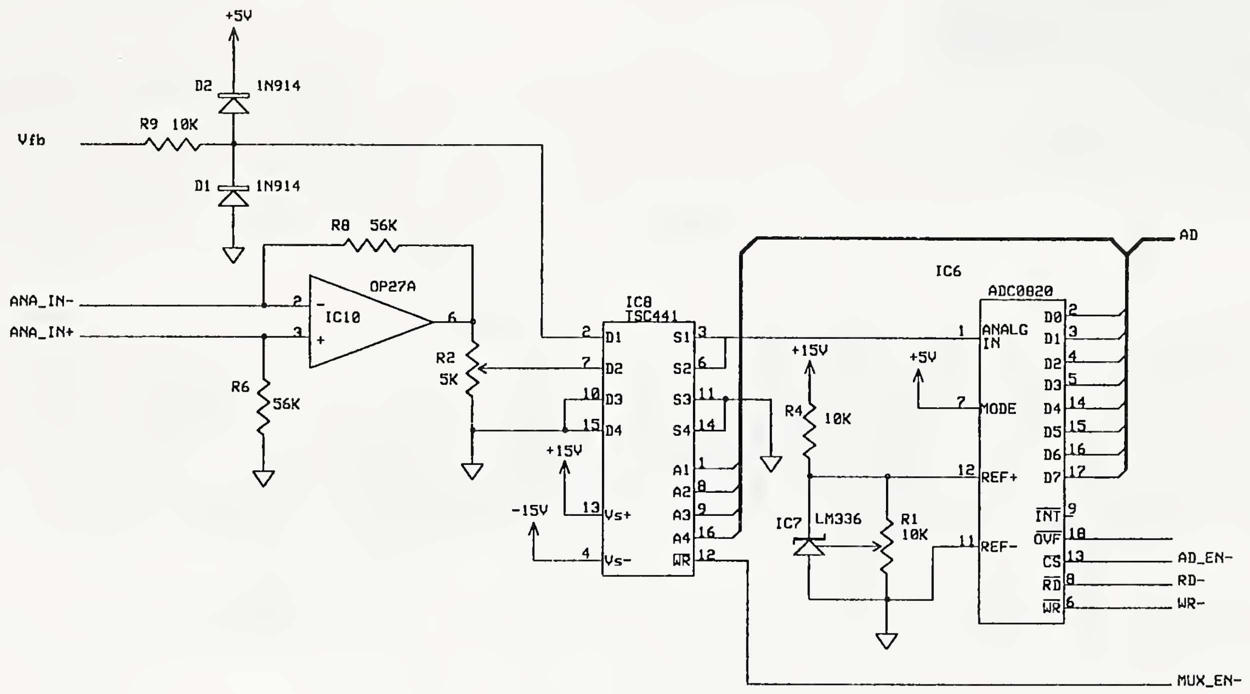


Cytek, Inc.
9 WALKER RD. N. ANDOVER, MA 01845
(508) 687-8086
PIEZOELECTRIC TRANSDUCER
CONTROLLER: CPU
By: A. Finger
Rev. A
04/22/90
SHEET 2/5
IC1 62256
IC6 27C256
IC5 /573
IC4
IC2 74HC138
IC3 MAX232
RP1 10K
CE5 10 @25V
CE6 10 @25V
CE7 10 @25V
CE8 10 @25V
CE9 10 @25V
C1 47pF
C2 47pF
X1 11.059MHz
R8 10K
R7 47K
+5V
AD
RD-
WR-
AD_EN-
MUX_EN-
DA_EN-
DIGIN_EN-
DIGOUT_E
RS232_OU
RS232_IN
DI0-
DI1-
DI2-
DI3-
DO0-
DO1-
DO2-
DO3-
OVF-
PS_RDY-
CHRG-
DISCHRG-
XTAL1
XTAL2
RST
EA
ALE
PSEN
Vpp



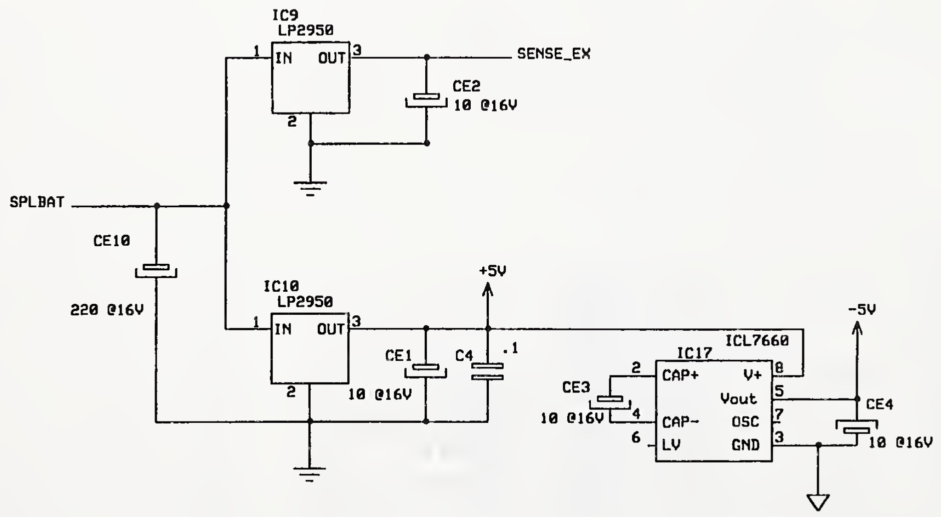
Cytek, Inc.
9 WALKER RD. N. ANDOVER, MA 01845
(508) 687-8086

PIEZOELECTRIC TRANSDUCER
CONTROLLER: CPU
By: A. Finger
Rev. A
04/22/90 SHEET 3/5



Cytek, Inc.
65 MYSTIC ST. METHUEN, MA 01844
(508) 687-8086

PIEZOELECTRIC TRANSDUCER
CONTROLLER: CPU
By: A. Finger
Rev. A
04/22/90 SHEET 4/5



Cytek, Inc.

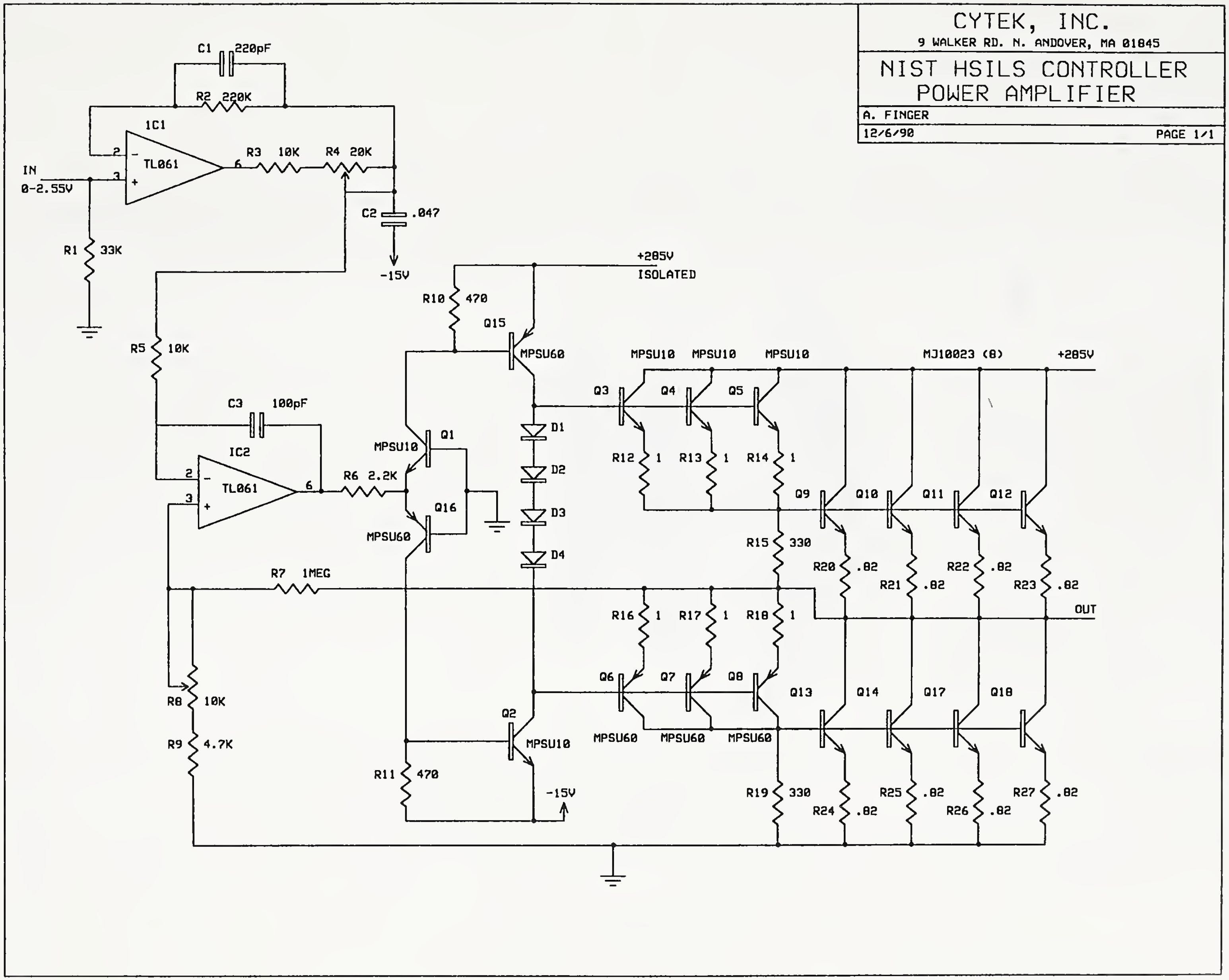
9 WALKER RD. N. ANDOVER, MA 01845
(508) 687-8086

PIEZOELECTRIC TRANSDUCER
CONTROLLER: CPU

By: A. Finger

Rev. A

04/22/90 SHEET 5/5



CYTEK, INC.
9 WALKER RD. N. ANDOVER, MA 01845

NIST HSILS CONTROLLER
POWER AMPLIFIER

A. FINGER
12/6/90 PAGE 1/1

Appendix D
HSILS Driver Software

Developed Under Contract by:

Cytek, Inc.
9 Walker Road
North Andover, MA 01845

```
/***************************************************************
**
**              FILE: main.c -- Piezoelectric Actuator Controller main module
**
**              The unit comes up monitoring the serial port for a profile download
**              and two pusbuttons connected to digital inputs bit 0 and 1.
**
**              A profile is an ascii list of time and voltage points to be followed
**              during operation. The first point is assumed to be 0,0 and all points
**              must be in order of increasing time. The list is terminated with the
**              keyword "END". The word "READY" is sent when the processor has
**              completed preprocessing. It then waits for switch 0 to be pressed
**              to initiate        execution of the profile with an initial 5 second delay.
**
**
**
**
**
**       $Log:   E:/CUSTS/NBS/CODE/VCS/NBS.C_V  $
 *
 *    Rev 1.1   01 Mar 1991  9:05:12   Alan Finger
 * Release 1.0
 *
 *    Rev 1.0   15 Jan 1991 21:45:50   Alan Finger
 * Initial revision.
**
**/
#include "nbs.h"
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>

xdata char lineBuf[20];

extern void InitIO();
extern BOOLEAN KeyStat();

extern BOOLEAN profileReady;
extern BOOLEAN driveReady;

void GetProfile();
void GetPulse();
void ProcessKey();
char strccmp(char * s1, char * s2);
BOOLEAN SwitchState(WORD b);

void ListCommands();
BOOLEAN GetPromptedString(char *s, int n);

main()
        {
        SetDA(0);
        InitIO();
        driveReady=FALSE;
        P1=0xff;

        puts("\r\nNIST HSILS Controller V1.0\r");
        putchar('>');

        for (;;)
                {
                if ( KeyStat() )
                        {
                        ProcessKey();
                        putchar('>');
                        }

                if ( SwitchState(0) )
                        {
                        if (profileReady)
                                {
                                if (driveReady)
                                        {
                                        RunProfile();
                                        puts("Profile Run Complete\r");
                                        }
                                else
                                        puts("Power Supply Not Charged.\r");
                                }
                        else
                                puts("ERROR: Profile not loaded\r");

                        while ( SwitchState(0) );
                        putchar('>');
                        }
                }
        }

void ProcessKey()
        {
        char c;

        c=getchar();
        switch (tolower(c))
                {
                case 'u':
                        {
                        GetProfile();
                        break;
                        }
                case 'p':
                        {
                        GetPulse();
                        break;
                        }
                default:
                        ListCommands();
                }
        }
```

```
void GetProfile()
    {
    long x;
    unsigned char v;
    BYTE error=FALSE;
    char *p;

    InitProfile();
    puts("\r");


    for (;;)
        {
        if ( !GetPromptedString(lineBuf, sizeof(lineBuf)) )
            return ;

        if ( strlen(lineBuf) > 0 )
            {
            if (lineBuf[0] == ';')
                continue;
            if ( !strccmp(lineBuf,"END") )
                {
                CompleteProfile();
                puts("Profile successfully loaded.\r");
                return ;
                }
            else
                {
                if ( (p=strtok(lineBuf,", \t")) != NULL)
                    {
                    if ( isSignedFpNumber(p) )
                        {
                        x=lrnd(atof(p));
                        if ( (p=strtok(NULL,", \t")) != NULL)
                            {
                            if ( !isSignedFpNumber(p) || !AddPoint(x,
v=(unsigned char) rnd(atof(p))) )
                                {
BadLine:
                                puts("Invalid line syntax. Rest of load
ignored\r");
                                do
                                    {
                                    if ( !GetPromptedString(lineBuf,
sizeof(lineBuf)) )
                                        return ;
                                    }
                                while ( strccmp(lineBuf,"END") );
                                puts("WARNING: Profile load
unsuccessful\r");
                                return ;
                                }
                            }
                        }
                    else
```

```
                        goto BadLine;
                    }
                }
            }
        }
    }

char strccmp(char * s1, char * s2)
    {
    char c1,c2;

    while ((c1=*s1) && (c2=*s2))
        {
        if (toupper(c1) > toupper(c2) )
            return 1;
        if (toupper(c1) < toupper(c2) )
            return -1;
        s1++; s2++;
        }
    if (*s1)
        return 1;
    if (*s2)
        return -1;
    return 0;
    }


BOOLEAN isnumber(char *s, BOOLEAN negAllowed, BOOLEAN fpAllowed)
    {
    BOOLEAN dpFound=FALSE;

    if ( (*s == ’-’ && negAllowed) || *s != ’-’  && *s != ’+’ && !isdigit(*s) )
        return FALSE;

    s++;
    for (;;)
        {
        switch (*s)
            {
            case ’\0’:
                return TRUE;

            case ’.’:
                {
                if ( !dpFound && fpAllowed)
                    dpFound=TRUE;
                else
                    return FALSE;
                }
            case ’0’:
            case ’1’:
            case ’2’:
            case ’3’:
            case ’4’:
            case ’5’:
```

```
                        case ’6’:
                        case ’7’:
                        case ’8’:
                        case ’9’:
                                {
                                s++;
                                break;
                                }
                        default:
                                {
                                return FALSE;
                                }
                        }
                }
        }


void ListCommands()
        {
        puts("Valid Commands Are:\r");
        puts("U                   - Load Profile\r");
        puts("P width,height     - issue a pulse\r");
        }

BOOLEAN GetPromptedString(char *s, int n)
        {
        int i;
        char tries;

        for (tries=0; tries < 5; tries++)
                {
                putchar(’:’);
                for (i=0; i < 20000; i++)
                        {
                        if (KeyStat())
                                {
                                gets(s,n);
                                putchar(’\r’);
                                return TRUE;
                                }
                        }
                }
        puts("TIMEOUT: Aborting Load\r");
        return FALSE;
        }
```

```
/*****************************************************************
**
**              FILE: act.c -- Piezoelectric Actuator Controller actuator control
**                                                          module.
**
**              A 'profile' is entered as a series of time-voltage pairs. The first
**              is always assumed to be 0,0. Each subsequent pair causes the
**              required current to reach that point to be calculated.  An empty line
**              terminates entry. The last point must have a zero voltage.
**
**
**
**      $Log:   E:/CUSTS/NBS/CODE/VCS/ACT.C_V  $
 *
 *    Rev 1.1   01 Mar 1991  9:05:38   Alan Finger
 * Release 1.0
 *
 *    Rev 1.0   15 Jan 1991 21:45:52   Alan Finger
 * Initial revision.
**
**/
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "nbs.h"


/*      The following are machine Cycles from profile.asm */
#define SETUP_CYCLES                    4
#define LOOP_CYCLES                     2
#define POINT_CYCLES                    19
#define MIN_CYCLES                      (SETUP_CYCLES+POINT_CYCLES+LOOP_CYCLES)

#define CYCLE_TIME              1.085                   /* microseconds/cycle */
#define SETUP_TIME              (SETUP_CYCLES*CYCLE_TIME)
#define MIN_TIME                        (MIN_CYCLES*CYCLE_TIME)
#define LOOP_TIME                       (LOOP_CYCLES*CYCLE_TIME)

#define PULSE_LOOP_CYCLES                       9
#define MIN_PULSE_CYCLES                        PULSE_LOOP_CYCLES+2
#define PULSE_LOOP_TIME                         (9*CYCLE_TIME)
#define MIN_PULSE_TIME                          (MIN_PULSE_CYCLES*CYCLE_TIME)


struct _points
        {
        BYTE loopCount;
        char Yincr;
        };

#define MAXPOINTS ( 16384/sizeof(struct _points) )

xdata struct _points     point[MAXPOINTS];

BOOLEAN profileReady;
```

```
data WORD pointCount;
data long t0;
data BYTE v0;

extern xdata char lineBuf[20];


void InitProfile()
        {
        t0=0;
        v0=0;
        profileReady=FALSE;
        pointCount=0;
        }

BOOLEAN AddPoint(long t, unsigned char v)
        {
        float slope;
        unsigned char loops;

        if (v > 250.0 || v < 0)
                {
                puts("ERROR: Requested Voltage outside 0-250 volts\r");
                return FALSE;
                }

        t/=CYCLE_TIME;                                                  /* convert t to cycles
*/

        if (t-t0 < MIN_CYCLES)
                {
                puts("WARNING: Point too close to previous, ignored\r");
                }

        if ( t < t0 )
                {
                puts("ERROR: Point time value less than previous\r");
                return FALSE;
                }

        if (v == v0 )
                {
                while ( t0+2*(MIN_CYCLES)+255*LOOP_CYCLES < t )
                        {
                        point[pointCount].loopCount=0;
                        t0+= MIN_CYCLES+255*LOOP_CYCLES;
                        v0+=point[pointCount++].Yincr=0;
                        }
                point[pointCount].loopCount=(t-t0-SETUP_CYCLES-POINT_CYCLES)/LOOP_CYCLES;
                t0+=SETUP_CYCLES+POINT_CYCLES+point[pointCount].loopCount*LOOP_CYCLES;
                v0+=point[pointCount++].Yincr=0;
                return TRUE;
                }

        slope= (float)(v-v0)/(t-t0);                    /* in v/cycle */
```

```
          if (fabs(slope) > 127 )
                    {
                    puts("ERROR: Maximum slope 127 volts/usec exceeded\r");
                    return FALSE;
                    }

          if (fabs(slope) > 1.0/MIN_CYCLES )
                    {
                    while ( t0+2*MIN_CYCLES < t )
                              {
                              point[pointCount].loopCount=1;
                              t0+= MIN_CYCLES;
                              v0+=point[pointCount++].Yincr=slope*MIN_CYCLES;
                              }
point[pointCount].loopCount=(t-t0-SETUP_CYCLES-POINT_CYCLES+1)/LOOP_CYCLES;
                    t0+=SETUP_CYCLES+POINT_CYCLES+point[pointCount].loopCount*LOOP_CYCLES;
                    v0+=point[pointCount++].Yincr=v-v0;
                    }
          else
                    {
                    loops= (unsigned char) ( (rnd(fabs(1/slope))
-SETUP_CYCLES-POINT_CYCLES+1)/LOOP_CYCLES);
                    while ( v0 != v )
                              {
                              point[pointCount].loopCount=loops;
                              t0+=SETUP_CYCLES+POINT_CYCLES+loops*LOOP_CYCLES;
                              point[pointCount++].Yincr= sgn(slope);
                              v0+=sgn(slope);
                              }
                    }
          return TRUE;
          }

void CompleteProfile()
          {
          AddPoint( t0*(long) CYCLE_TIME + (long) MIN_TIME * 2,0);
          profileReady=TRUE;
          }

data unsigned pulseWidth;
data unsigned char pulseHeight;

void GetPulse()
          {
          char *p;
          void RunPulse();

          gets(lineBuf, sizeof(lineBuf) );
          putchar('\r');
          if ( (p=strtok(lineBuf,", \t")) != NULLPTR &&
                              isUnsignedInteger(p) &&
                              (pulseWidth=(rnd(atof(p))/PULSE_LOOP_TIME)) > 0 )
                    {
                    if ( (p=strtok(NULLPTR,", \t")) != NULLPTR &&
                                    isUnsignedInteger(p)  &&
                              (pulseHeight=atoi(p)) < 255 )
                              {
                              RunPulse();
                              return ;
                              }
                       }
          puts("Syntax: width (25-71,100 usec), height (0-255 volts)\r");
          }

#ifdef MSC
PrintProfile()
          {
          int i;
          unsigned char v=0;
          long t=0;
          FILE *output;

          if ( (output=fopen("test.out","w")) == NULLPTR)
                              puts("ERROR opening test.out");

          for (i=0; i < pointCount; i++)
                    {
                    v+=point[i].Yincr;

t+=(point[i].loopCount*LOOP_CYCLES+SETUP_CYCLES+POINT_CYCLES)*CYCLE_TIME;

                    if (output != NULLPTR)
                              {
                              fprintf(output,"%lu\t%u\n",t,v);
                              }

                    printf("%5u: dx=%3u dv=%3d   t=%6lu v=%3u\n", i,
                              point[i].loopCount,
                              point[i].Yincr,
                              t,v);
                    }
                    if (output != NULLPTR)
                              fclose(output);
          }
#endif
```

```
/****************************************************************
**
**              FILE: io.c -- Piezoelectric Actuator Controller I/O and timer
**
**      $Log:   E:/CUSTS/NBS/CODE/VCS/IO.C_V  $
 *
 *    Rev 1.1   01 Mar 1991  9:05:46   Alan Finger
 * Release 1.0
 *
 *    Rev 1.0   15 Jan 1991 22:27:20   Alan Finger
 * Initial revision.
**
**/
#include "nbs.h"

xdata char buffer[20];
data char *head;
data char *tail;
data BYTE bufCount;
data WORD delayTime;
data BYTE switchCount[4];
data BOOLEAN driveReady;
data BYTE blinkCount;

extern BOOLEAN profileReady;

void InitIO()
        {
        head=tail=buffer;
        bufCount=delayTime=0;
        blinkCount=25;

        SCON = 0x52;  /* SCON */       /* setup serial port control */
        TMOD = 0x21;            /* TMOD */      /* hardware (9600 BAUD @11.059MHZ) */
        TL0=   HW_COUNTS & 0xff;
        TH0=   (WORD) HW_COUNTS >> 8;
        TCON = 0x50;  /* TCON */
        TH1 =  0xfd;    /* TH1 */
        RI=FALSE;
        EA=1;
        ET0=1;
        ES=1;
        }

BOOLEAN KeyStat()
        {
        return ( bufCount != 0 );
        }

char _getkey()
        {
        char c;
        while ( !bufCount ) ;
        ES=0;
        c=*head;
        if ( head == &buffer[19] )
                head=buffer;
        else
                head++;
        bufCount--;
        ES=1;
        return c;
        }

void putchar( char c )
        {
        TI=FALSE;
        SBUF=c;
        while (!TI);
        }

void SerialIO() interrupt 4 using 2
        {
        char c;
        if (RI)
                {
                c=SBUF;
                RI=FALSE;
                if ( bufCount < sizeof(buffer) )
                        {
                        bufCount++;
                        *tail=c;
                        if ( tail == &buffer[19] )
                                tail=buffer;
                        else
                                tail++;
                        }
                }
        }

void timer0() interrupt 1 using 1
        {
        BYTE i;

        if (!(P1&1) )
                {
                if (switchCount[0] != 4)
                        switchCount[0]++;
                }
        else
                switchCount[0]=0;

                if (!(P1&2) )
                        {
                        driveReady=TRUE;
                        }
                else
                        {
                        driveReady=FALSE;
                        }
                if (!(P1&4) )
                        {
                        if (switchCount[2] != 4)
                                switchCount[2]++;
                        }
                else
                        switchCount[2]=0;

                if (!(P1&8) )
                        {
                        if (switchCount[3] != 4)
                                switchCount[3]++;
                        }
                else
                        switchCount[3]=0;

        if (delayTime)
                delayTime--;

        if (driveReady )
                {
                if  (profileReady)
                        {
                        if (!blinkCount--)
                                {
                                blinkCount=25;
                                P1^=0x10;
                                }
                        }
                else
                        P1&=0xef;
                }
        else
                P1|=0x10;
        }

void Delay( WORD ticks)
        {
        delayTime=ticks;
        while ( delayTime ) ;
        }
BOOLEAN SwitchState(WORD s)
        {
        return (switchCount[s] == 4 );
        }
```

```
/*************************************************************
**
**              FILE: strtok.c -- string token parser
**
**              Parses a string into tokens separated by a specified set
**              of delimiters. The Franklin compiler is missing this standard
**              function.
**
**              char *strtok(char *s, char *d)
**
**              On the initial call, s points to the string to be parsed. On
**              subsequent calls, s should be a null ptr. leading whitespace
**              is skipped and the string is terminated when any of the characters
**              in string d is encountered. strtok returns a pointer to the
**              first non-whitespace character or NULL if the end of the
**              string has been reached.
**
**      $Log:   E:/CUSTS/NBS/CODE/VCS/NBS.C_V  $
 *
 *    Rev 1.0   15 Jan 1991 21:45:50   Alan Finger
 * Initial revision.
**/
#include <ctype.h>
#include <string.h>
#include "nbs.h"

char *strtok(char *s, char *d)
        {
        static char *p;
        char *t;
        char c;

        if (s != NULLPTR)
                p=s;

        while ((c=*p) && isspace(c) )
                p++;

        if (c == ’\0’)
                return NULLPTR;
        t=p;
        while ( (c=*p) && strpos(d,c) == -1 )
                p++;

        if (c != ’\0’)
                *(p++)=’\0’;

        return t;
        }
        $PAGELENGTH(60)
        $DEBUG
; !R! FONT 15; EXIT;
```

```
;****************************************************************
;
;       HSILS Controller Profile Execution Module
;
;
;
;       $include(c:\c51\include\reg51.inc)

        NAME PROFILE

DtoA    EQU 8002h

        extrn   XDATA (point)
        extrn   DATA (pointCount)
        extrn   DATA (pulseWidth, pulseHeight)

        public  RunProfile
        public  RunPulse

dtoaptr         EQU     R0
y               EQU     R1


codeseg         segment         CODE
        rseg    codeseg
;****************************************************************
;
;       RunPulse
;
;       Inputs:
;               struct _point *point
;               unsigned pointCount
;
pointsh         EQU     R2
pointsl         EQU     R3

RunProfile:
        clr     EA                              ;disable interrupts

; Clear D/A
        mov     a,#0
        mov     y,a
        mov     dtoaptr,#LOW DtoA
        mov     P2,#HIGH DtoA
        movx    @dtoaptr,a

        mov     dph,#HIGH point                 ;initialize point array pointer
        mov     dpl,#LOW point
;
; Convert the # of points to a negative #
;

        clr     c
        mov     a,#0
        subb    a,pointCount+1
        mov     pointsl,a
        mov     a,#0
        subb    a,pointCount
        mov     pointsh,a

        anl     p1,#0dfh

NextPoint:
;
; get the delay loop count and wait
;
;                                       ; Cycles
        movx    a,@dptr                         ;2
        inc     dptr                    ;2
                                        ;=4
TightLoop:
        djnz    acc,Tightloop           ;2
;
; Get the delta V and update the D/A and y

        movx    a,@dptr                         ;2
        inc     dptr                    ;2
        add     a,y                     ;1
        mov     y,a                     ;1
        mov     P2,#HIGH DtoA                   ;2
        movx    @dtoaptr,a              ;2
;
; Any more points?

        mov     a,pointsl               ;1
        add     a,#1                    ;1
        mov     pointsl,a               ;1
        mov     a,pointsh               ;1
        addc    a,#0                    ;1
        mov     pointsh,a               ;1
        orl     a,pointsl               ;1
        jnz     NextPoint               ;2
                                        ;=19

        orl     p1,#20h
        setb    EA
        ret

;*****************************************************************
;
;       RunPulse
;
;       Inputs:
;               long pulseWidth
;               unsigned char pulseHeight
;

loopsl          EQU     R2
loopsh          EQU     R3
```

```
RunPulse:
          clr       c
          mov       a,#0
          subb      a,PulseWidth+1
          mov       loopsl,a
          mov       a,#0
          subb      a,PulseWidth
          mov       loopsh,a

; Set D/A
          mov       a,#0
          mov       dtoaptr,#LOW DtoA
          mov       P2,#HIGH DtoA
          clr       EA                            ;disable interrupts

          mov       a,pulseHeight
          anl       p1,#0dfh
          movx      @dtoaptr,a

PulseLp:
          mov       a,loopsl                      ;1
          add       a,#1                          ;1
          mov       loopsl,a                      ;1
          mov       a,loopsh                      ;1
          addc      a,#0                          ;1
          mov       loopsh,a                      ;1
          orl       a,loopsl                      ;1
          jnz       PulseLp                                 ;2
                                                  ;=9
; Clear D/A

          movx      @dtoaptr,a                    ;2

          orl       p1,#20h
          setb      EA
          ret



;
          END
```

```
/***************************************************************
**
**      FILE: nbs.h -- Piezoelectric Actuator Controller include file
**
**
**
**
**      $Log:   E:/CUSTS/NBS/CODE/VCS/NBS.H_V  $

     Rev 1.1   01 Mar 1991  9:05:52   Alan Finger
  Release 1.0

     Rev 1.0   15 Jan 1991 21:45:52   Alan Finger
  Initial revision.
**
**/
#ifndef NBSH
#define NBSH
#ifndef MSC
#include <reg51.h>
#include <absacc.h>
#endif

#ifdef MSC
typedef char BOOLEAN;
#else
typedef bit BOOLEAN;
#endif
typedef unsigned char BYTE;
typedef unsigned int  WORD;
typedef unsigned long LONGWORD;

#ifdef MSC
#define xdata
#endif

#define FALSE 0
#define TRUE (!FALSE)

#ifndef NULL
#define NULL ((void *) 0 )
#endif

#ifdef __SMALL__
#define NULLPTR      0x40000L
#endif

#ifdef __LARGE__
#define NULLPTR      0x20000L
#endif

#ifndef MSC
#define OVERFLOW            P3^2
#define PS_READY            P3^3
#define CHARGE_EN           P3^4
```

```
#define DISCHG_EN          P3^5

#define SetDA(x)           XBYTE[0x8002]=x;
#endif


#define FS_STACK           256                          /* FS stack voltage */

/*
**      counter runs at 1.0850 usec/count. 9216 counts/clock tick yields
** 10.181 msec per tick.
*/
#define COUNTS_PER_TICK         9216
#define HW_COUNTS                    -9216              /*  actually  65536-count
*/


#define isSignedFpNumber(s) isnumber(s, TRUE, TRUE)
#define isUnsignedFpNumber(s) isnumber(s, FALSE, TRUE)
#define isSignedInteger(s) isnumber(s, TRUE, FALSE)
#define isUnsignedInteger(s) isnumber(s, FALSE, FALSE)
BOOLEAN isnumber(char *s, BOOLEAN negAllowed, BOOLEAN fpAllowed);

#define lrnd(x) ( (long) (x+.5) )
#define rnd(x) ( (int) (x+.5) )
#define sgn(x)  ( (x >= 0) ? 1:-1 )

char strccmp(char * s1, char * s2);
void InitProfile();
BOOLEAN AddPoint(long t, unsigned char v);
void GetProfile();
void CompleteProfile();
void RunProfile();
void Delay(WORD ticks);
char *strtok(char *s, char *d);
#endif
```

```
################################################################
####
##
## HSILS Controller Polymake Makefile
##
##

OBJS            = nbs.obj act.obj io.obj profile.obj strtok.obj

.PATH.c= .
.PATH.obj= .
.PATH.hex = .
.PATH.h = .


nbs.hex:        $(OBJS)
                l51 nbs.obj,act.obj,io.obj,profile.obj,strtok.obj PRECEDE (?DT?NBS) NOOVERLAY
                ohs51 nbs

.c.obj:
        c51 $< DB LARGE CODE DEFINE(__LARGE__)

.asm.obj:
        a51 $<

clean:
        del nbs.hex

nbs.obj:        nbs.h
act.obj: nbs.h
io.obj:         io.c nbs.h
```

Appendix E
HSILS Fabrication Drawings

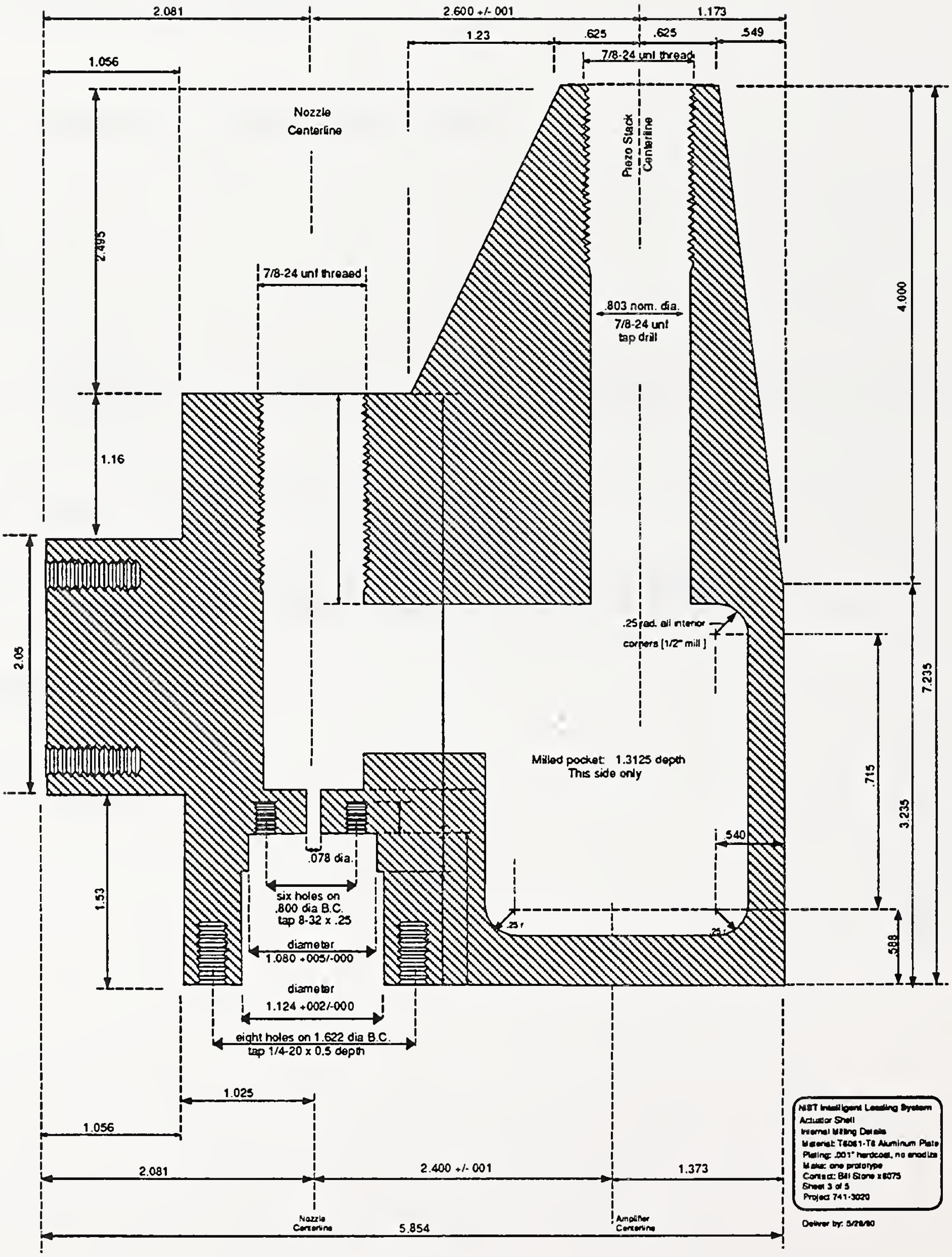


Fig. E.1: Side view cut-away of HSILS housing showing main bore holes for valve core rod, piezoelectric stack, and expansion nozzle. Units shown are inches; 1 inch = 25.4 mm.

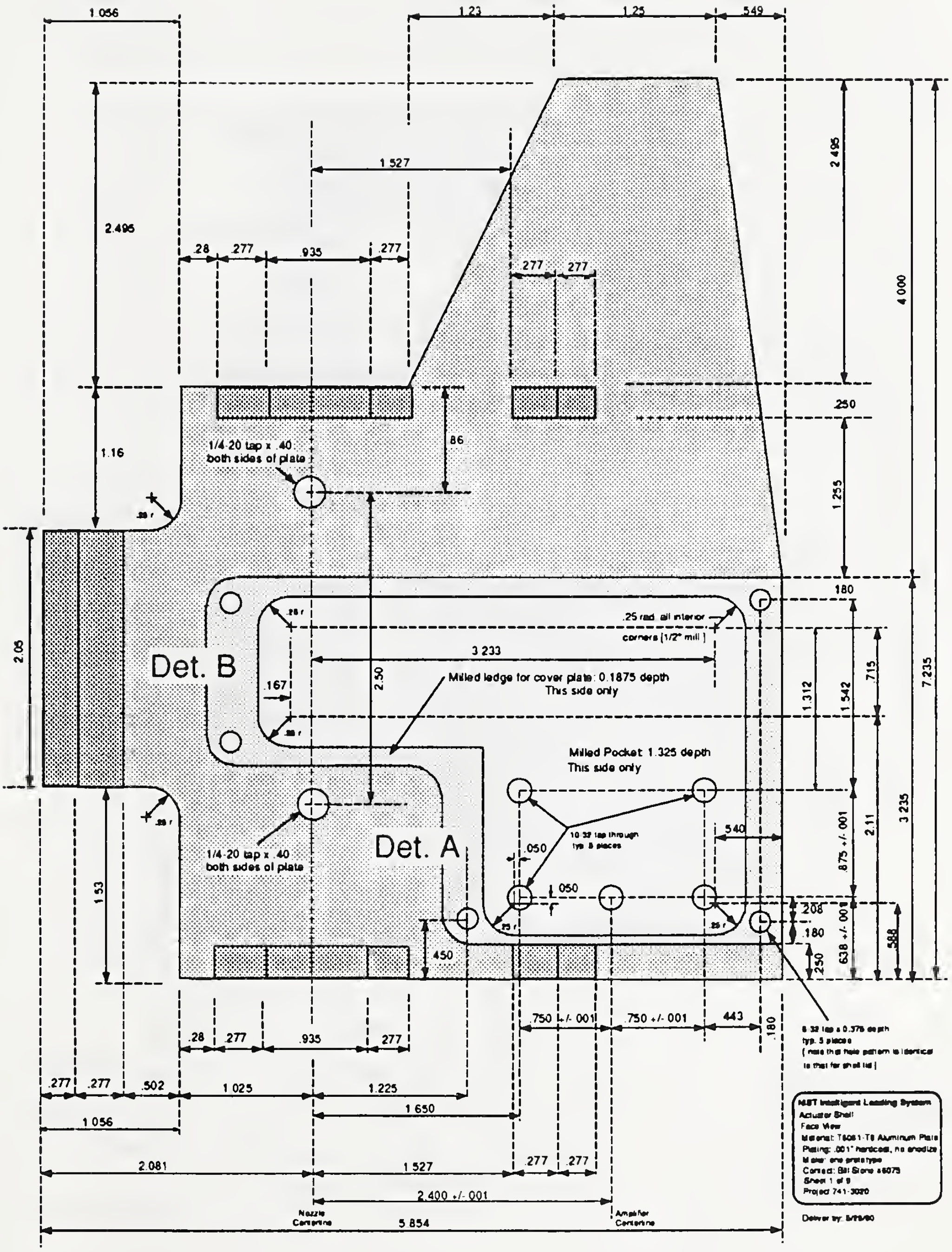


Fig. E.2: Side View of HSILS housing showing exterior milling details and mounting holes for the displacement amplifier. Units shown are inches; 1 inch = 25.4 mm.

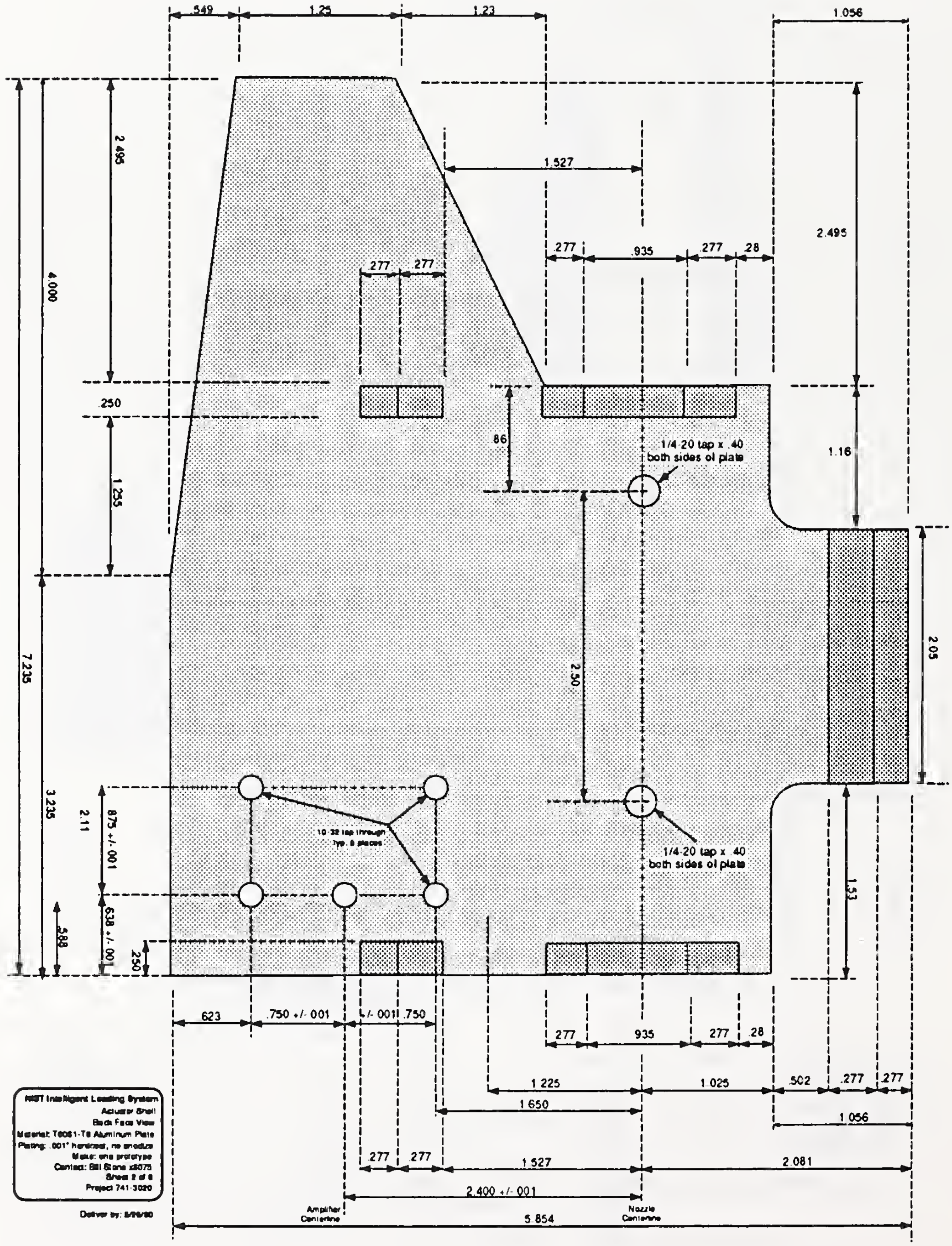
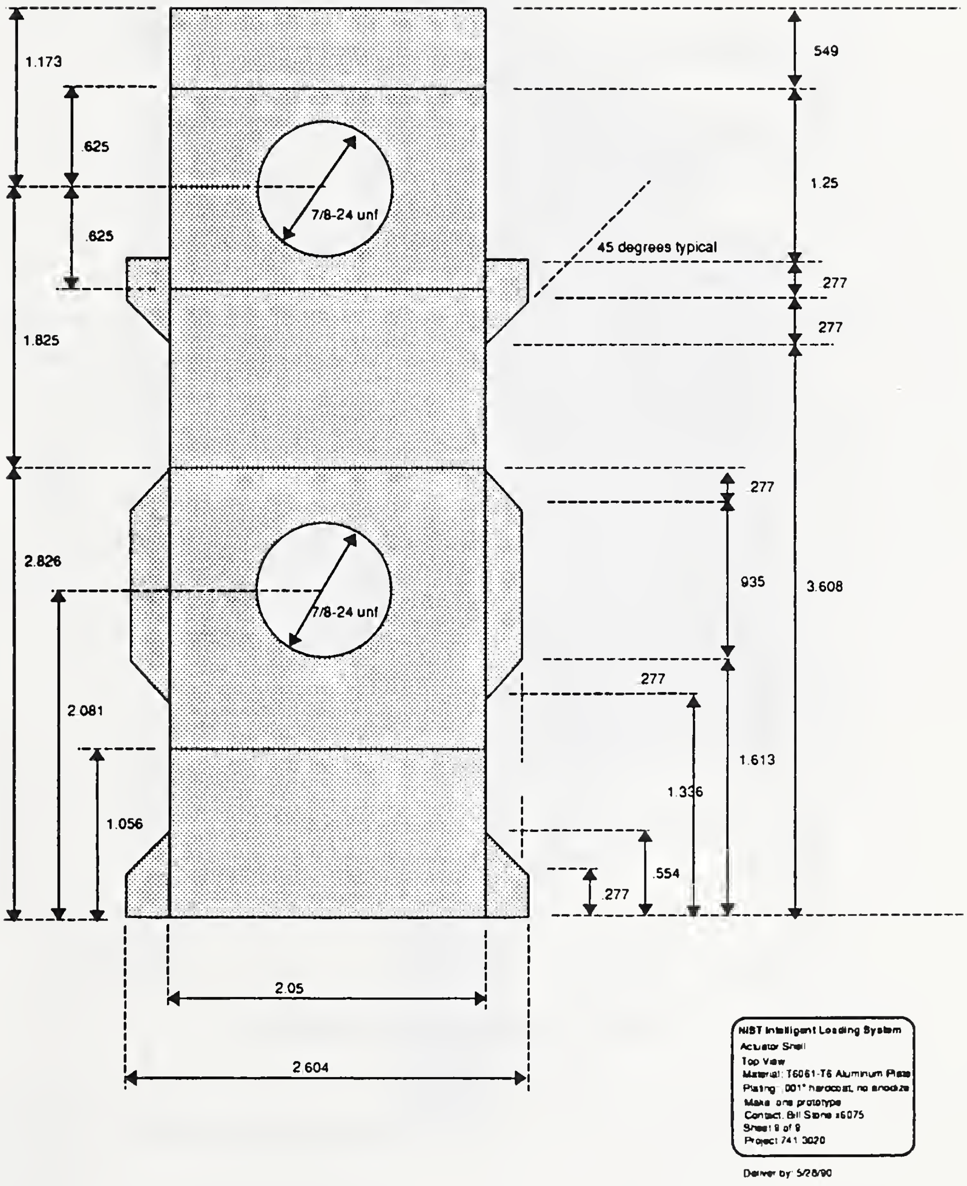


Fig. E.3: Reverse side view of HSILS housing showing accumulator mounting holes and positioning tabs. Units shown are inches; 1 inch = 25.4 mm.

Top View: Actuator Shell



NIST Intelligent Loading System
Actuator Shell
Top View
Material: T6061-T6 Aluminum Plate
Plating .001" hardcoat, no anodize
Make one prototype
Contact. Bill Stone x6075
Sheet 9 of 9
Project 741 3020

Deliver by: 5/28/90

Fig. E.4: Top external view of HSILS housing showing piezoelectric stack and core rod bore holes and mounting tabs for gas accumulators. Units shown are inches; 1 inch = 25.4 mm.

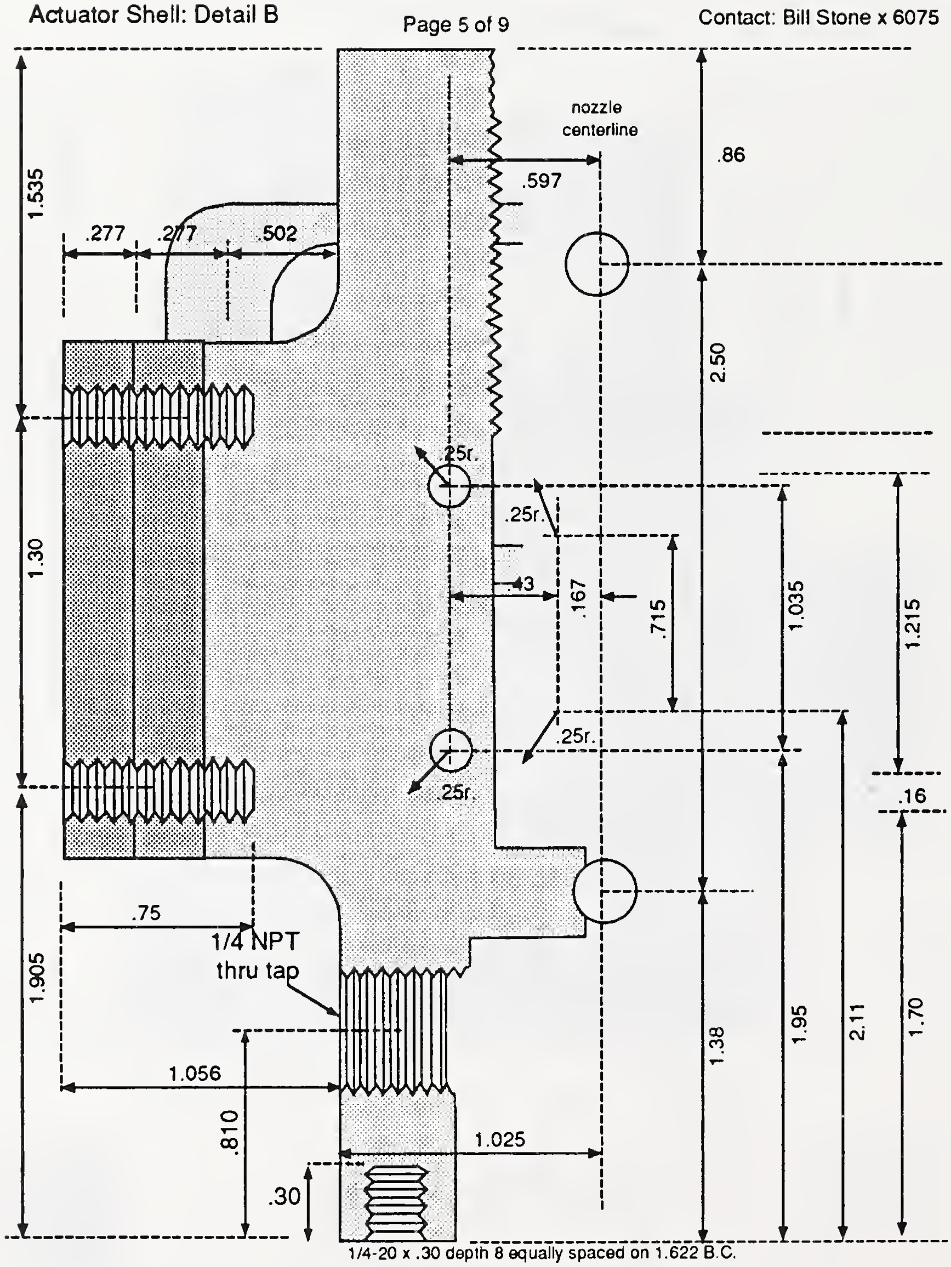
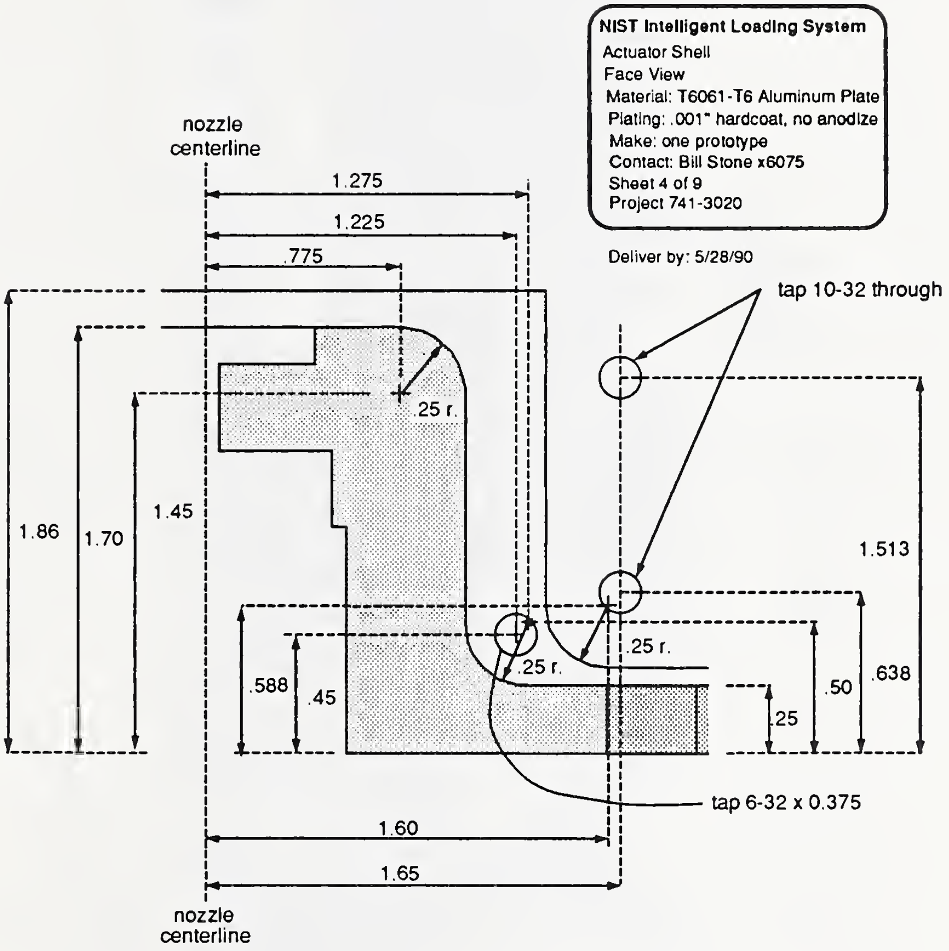


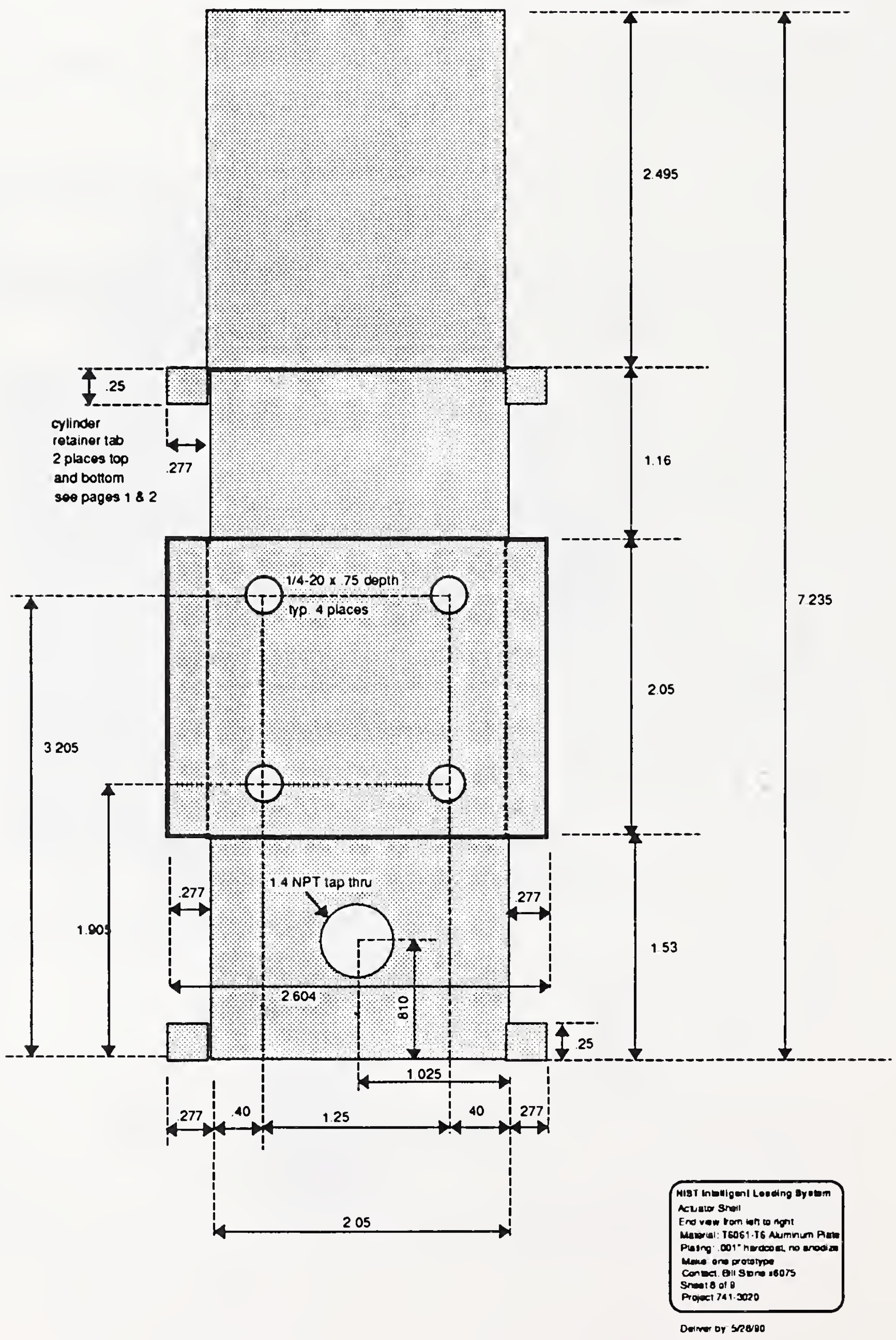
Fig. E.5: Side View detail of HSILS housing in the vicinity of the valve core seat showing threaded connections for the expansion nozzle, high pressure gas inlet from the accumulators, and external mounting bolts (left). Units shown are inches; 1 inch = 25.4 mm.



Detail A: Amplifier Shell

Fig. E.6: Side View detail of HSILS Cover plate (see additionally Fig. E.20) fitting tolerances. Units shown are inches; 1 inch = 25.4 mm.

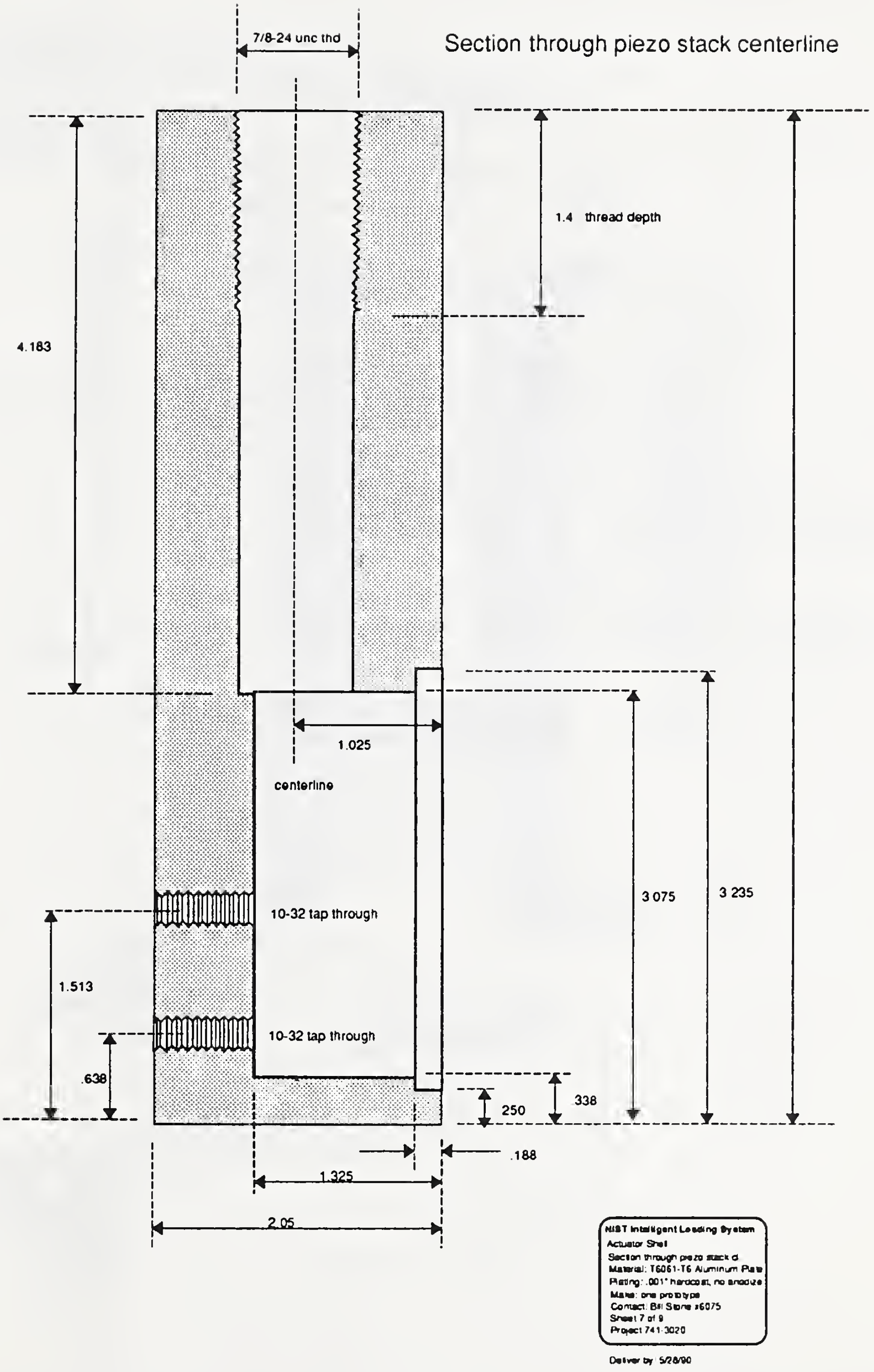
End View, Looking from left towards right, pages 1-3



NIST Intelligent Loading System
Actuator Shell
End view from left to right
Material: T6061-T6 Aluminum Plate
Plating: .001" hardcoat, no anodize
Make one prototype
Contact: Bill Stone x6075
Sheet 8 of 9
Project 741-3020

Deliver by 5/28/90

Fig. E.7: End view of HSILS housing showing mounting holes and high pressure gas inlet. Units shown are inches; 1 inch = 25.4 mm.



NIST Intelligent Loading System
Actuator Shell
Section through piezo stack cl
Material: T6061-T6 Aluminum Plate
Plating: .001" hardcoat, no anodize
Make: one prototype
Contact: Bill Stone #6075
Sheet 7 of 9
Project 741-3020

Deliver by 5/28/90

Fig. E.8: End View of HSILS housing interior pockets showing central boreholes for the core rod and stack, milled pocket for the displacement amplifier, and recessed pocket for side cover. Units shown are inches; 1 inch = 25.4 mm.

Section through nozzle centerline

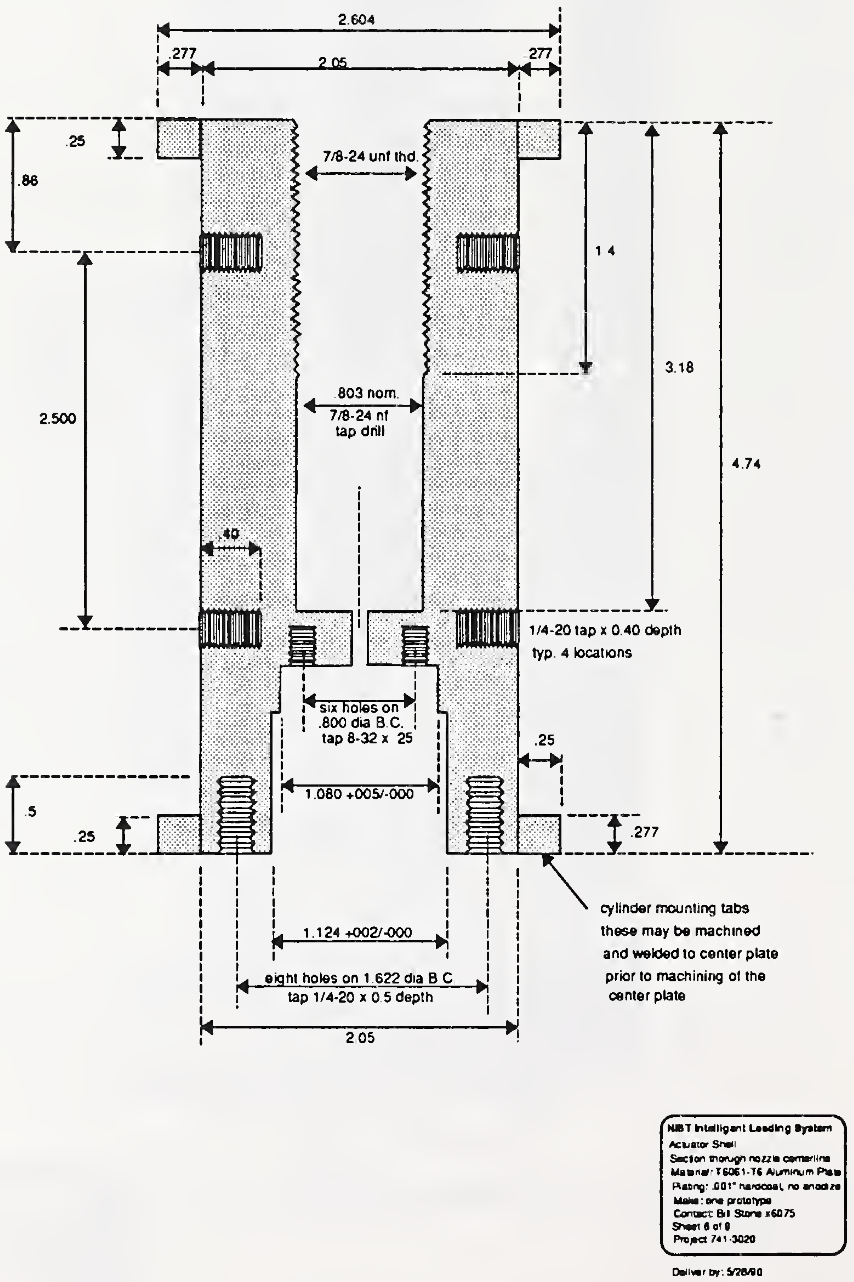


Fig. E.9: End View of HSILS housing showing interior details of high pressure valve seat cavity and alignment of valve core rod borehole. Units shown are inches; 1 inch = 25.4 mm.

NIST High Speed Intelligent Loading System

Part: piezo amplifier

Material: 120 ksi yield titanium [alloy not critical]

Version: 2/21/90
Contact: Stone x6075
Fabricate: 1 unit
Page 1/2

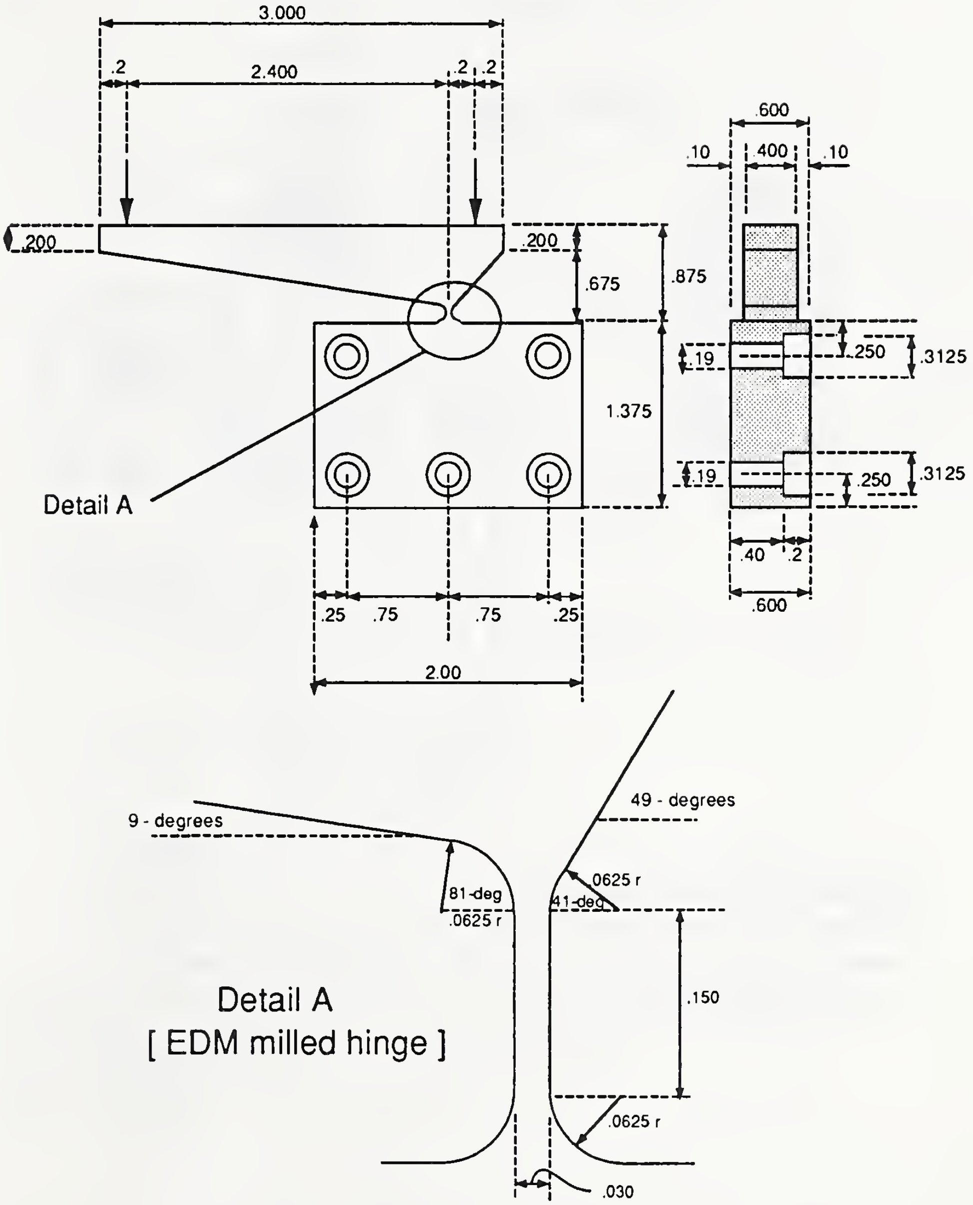
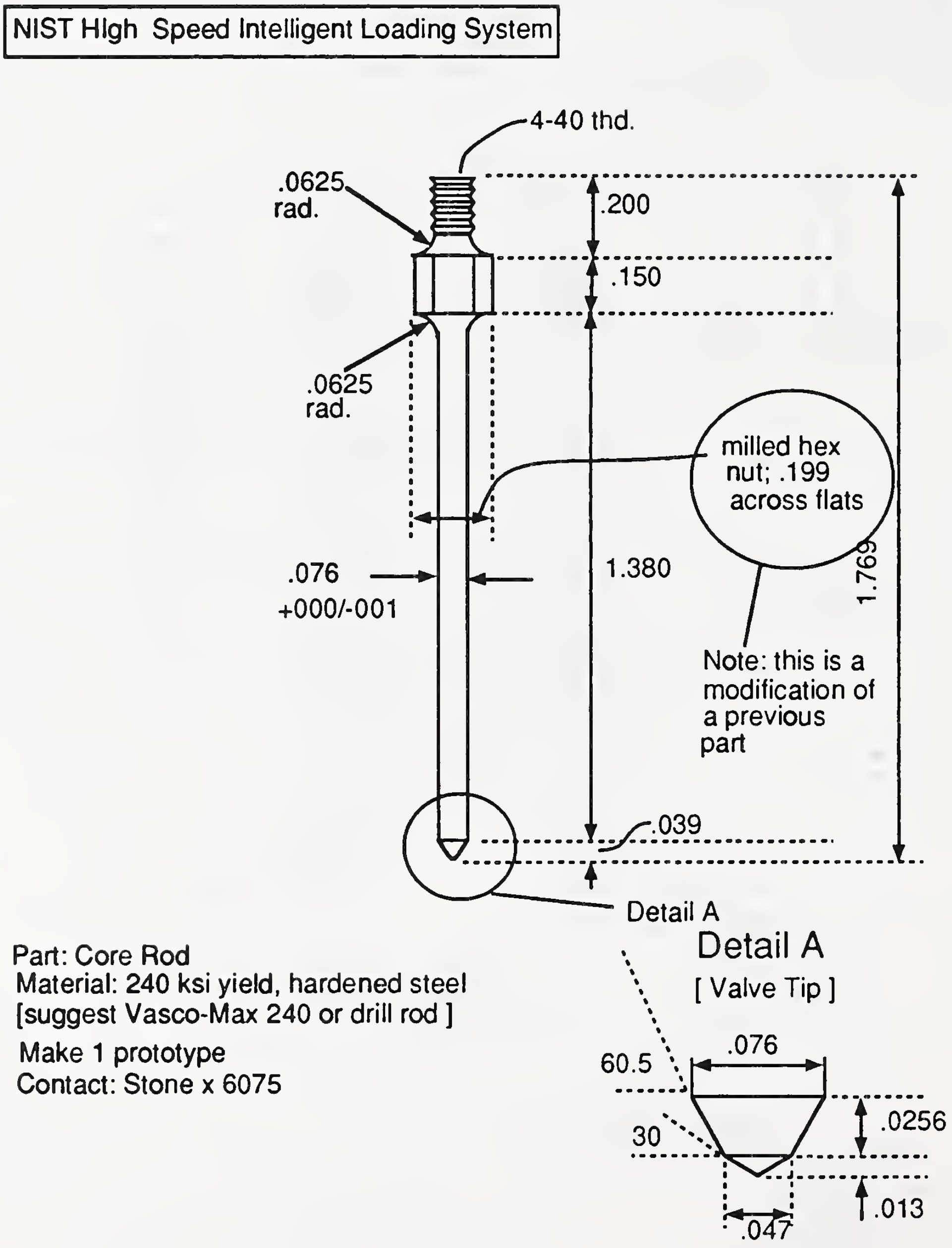


Fig. E.10: Detail of displacement amplifier. This was monolithically wire EDM milled from a solid plate of titanium. Units shown are inches; 1 inch = 25.4 mm.


NIST HIgh Speed Intelligent Loading System
4-40 thd.
.0625 rad.
.200
.150
.0625 rad.
milled hex nut; .199 across flats
.076 +000/-001
1.380
1.769
Note: this is a modification of a previous part
.039
Detail A
Detail A
[Valve Tip]
.076
60.5
.0256
30
.013
.047

Part: Core Rod
Material: 240 ksi yield, hardened steel
[suggest Vasco-Max 240 or drill rod]
Make 1 prototype
Contact: Stone x 6075

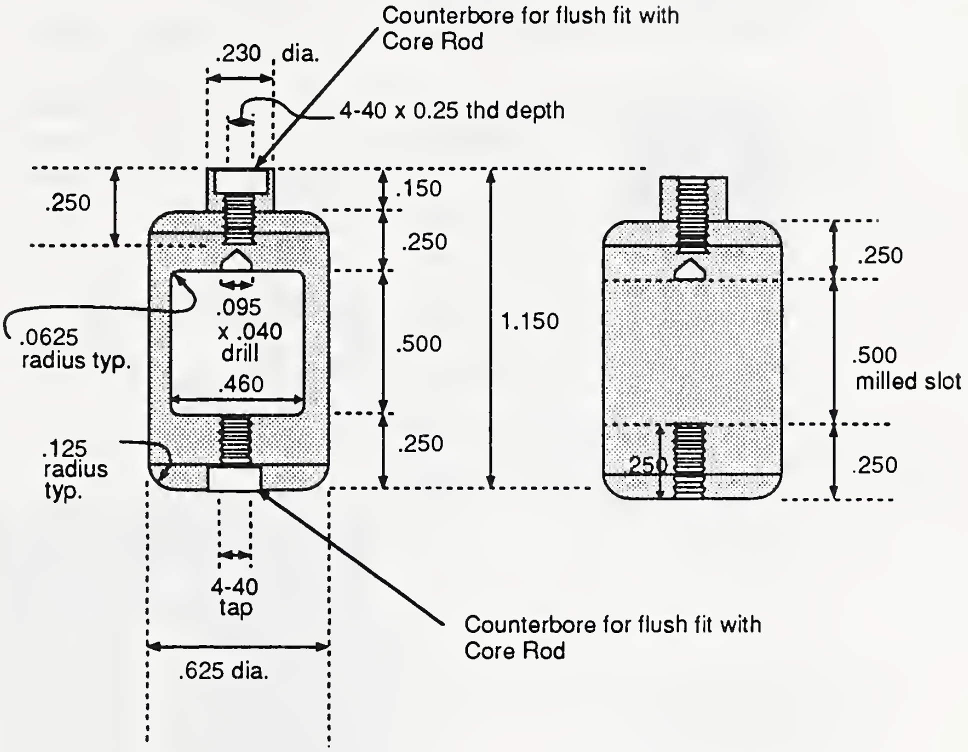
Fig. E.11: Detail of valve core rod. This was first turned from maraging steel bar stock and then heat treated. Units shown are inches; 1 inch = 25.4 mm.

NIST High Speed Intelligent Loading System

Part: Core Connector

Material: T7075 aluminum; anodize blue

Version 2/23/90
Contact: Stone x6075
Fabricate: 1 unit
Page 1/1



Notes:

1. Material: 5/8 aluminum bar stock;
2. First choice T7075; if not available, use T6061.
3. Anodize blue; allow .002 undersize dia on mating parts
4. taper at end of .095 drilled hole not critical

Fig. E.12: Valve core rod coupler. Valve core rod (Fig. E.11) attaches to the base while displacement amplifier (Fig. E.10) seats in the central milled pocket and bears in compression against the conical interior seat into which is glued a sapphire sphere prior to assembly with the displacement amplifier. Units shown are inches; 1 inch = 25.4 mm.

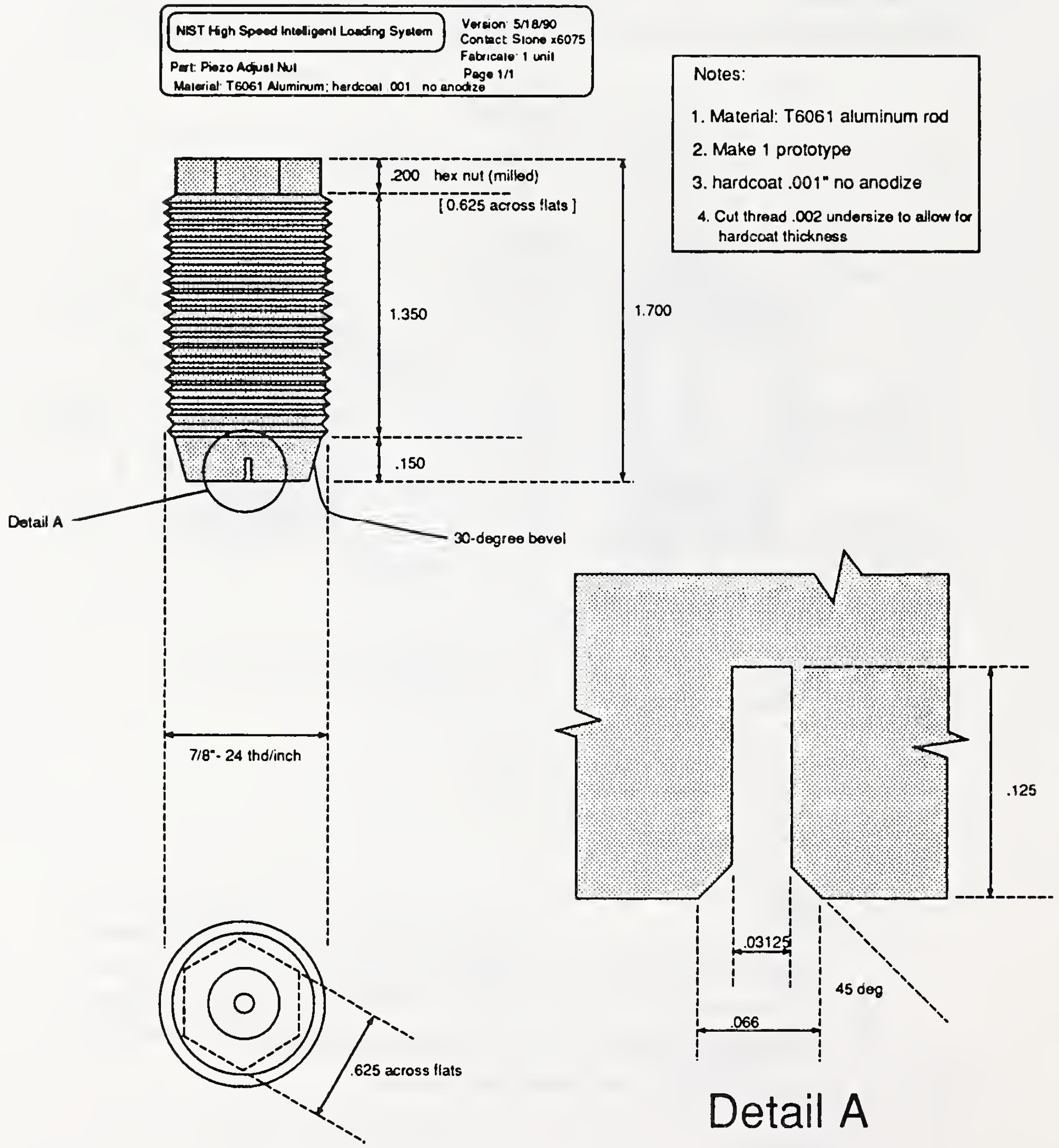
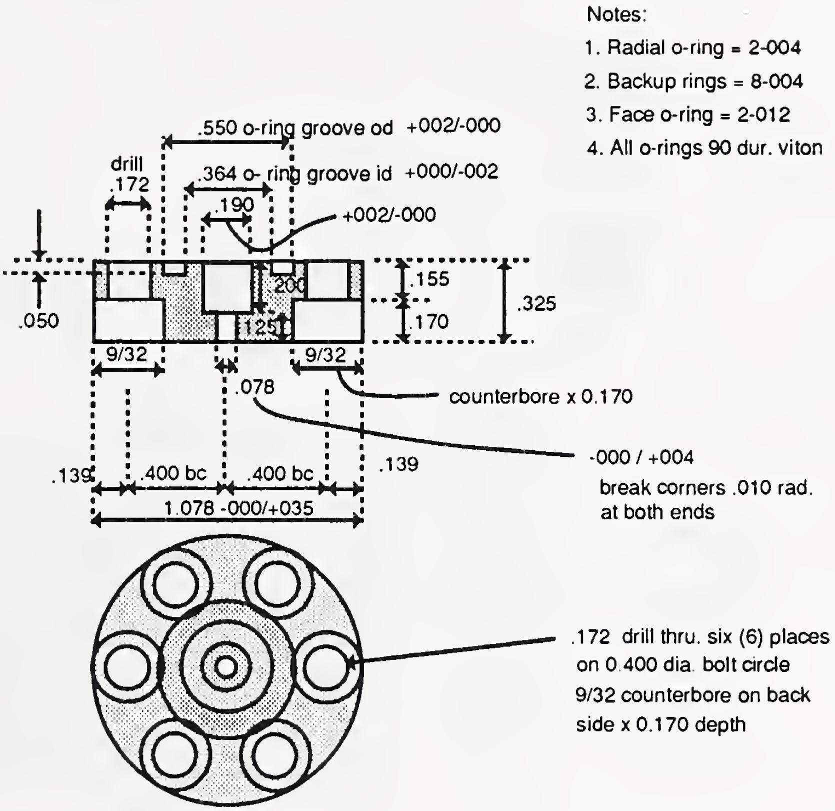


Fig. E.13: Fine thread adjustment screw. One is used to set the precompression in the return spring and a second is used to permit initial setting and loading of the piezoelectric stack. Units shown are inches; 1 inch = 25.4 mm.

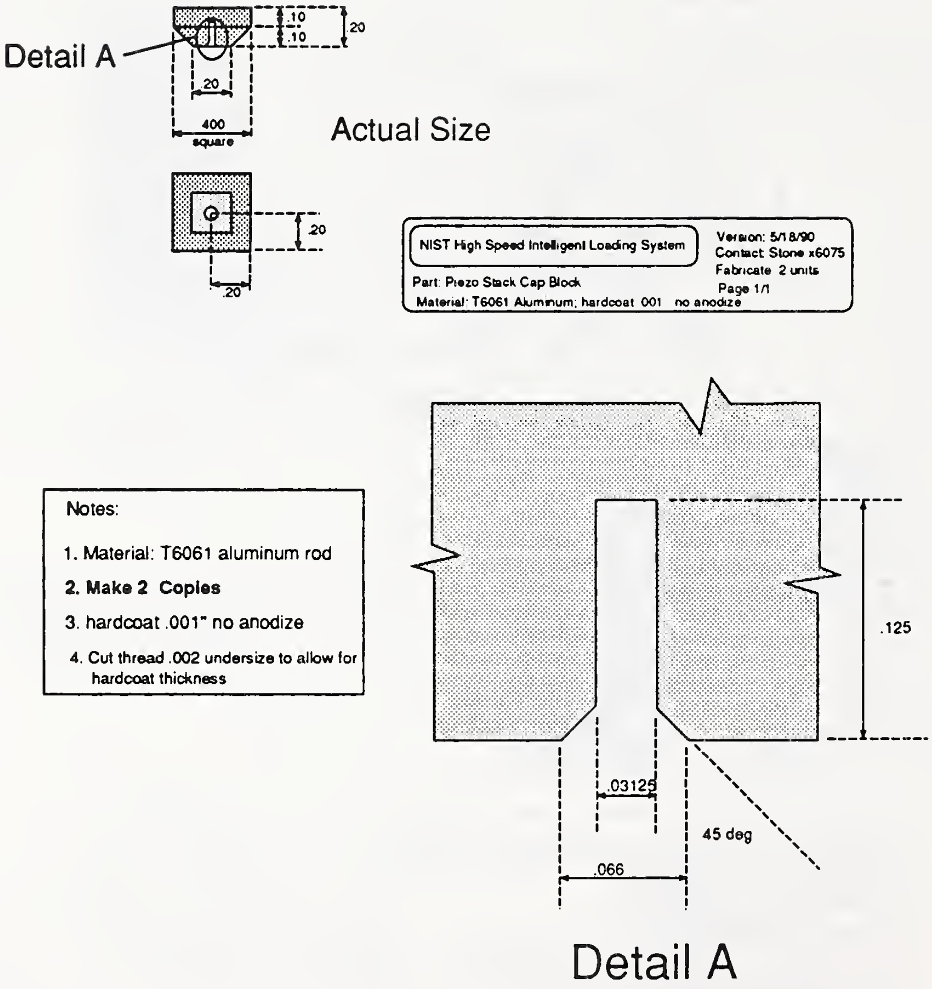
NIST High Speed Intelligent Loading System



Notes:
1. Radial o-ring = 2-004
2. Backup rings = 8-004
3. Face o-ring = 2-012
4. All o-rings 90 dur. viton

Part: HP Core Valve Seal Plate
Material: SS 304
Rev. B, 5/08/90
Contact: Bill Stone x 6075
Fabricate one (1) unit

Fig. E.14: Valve core sealing plate. Provides the necessary high pressure o-ring and backup ring seals to isolate the high pressure valve seat cavity from the ambient pressure displacement amplification and drive portions of the HSILS unit. Units shown are inches; 1 inch = 25.4 mm.



NIST High Speed Intelligent Loading System

Part: Piezo Stack Cap Block

Material: T6061 Aluminum; hardcoat .001 no anodize

Version: 5/18/90
Contact Stone x6075
Fabricate 2 units
Page 1/1

Notes:

1. Material: T6061 aluminum rod
2. **Make 2 Copies**
3. hardcoat .001" no anodize
4. Cut thread .002 undersize to allow for hardcoat thickness

Fig. E.15: Pyramid block with conical seat. These are glued to each end of the piezoelectic stack using strain gage glue. Sapphire bearings are subsequently glued into the conical seats prior to final assembly. Units shown are inches; 1 inch = 25.4 mm.

NIST High Speed Intelligent Loading System

Part: HP Nozzle
Material: SS 303 or SS 302
Rev. B: 6/29/90

Contact: Stone x6075
Fabricate: 1 unit
Page 1/2

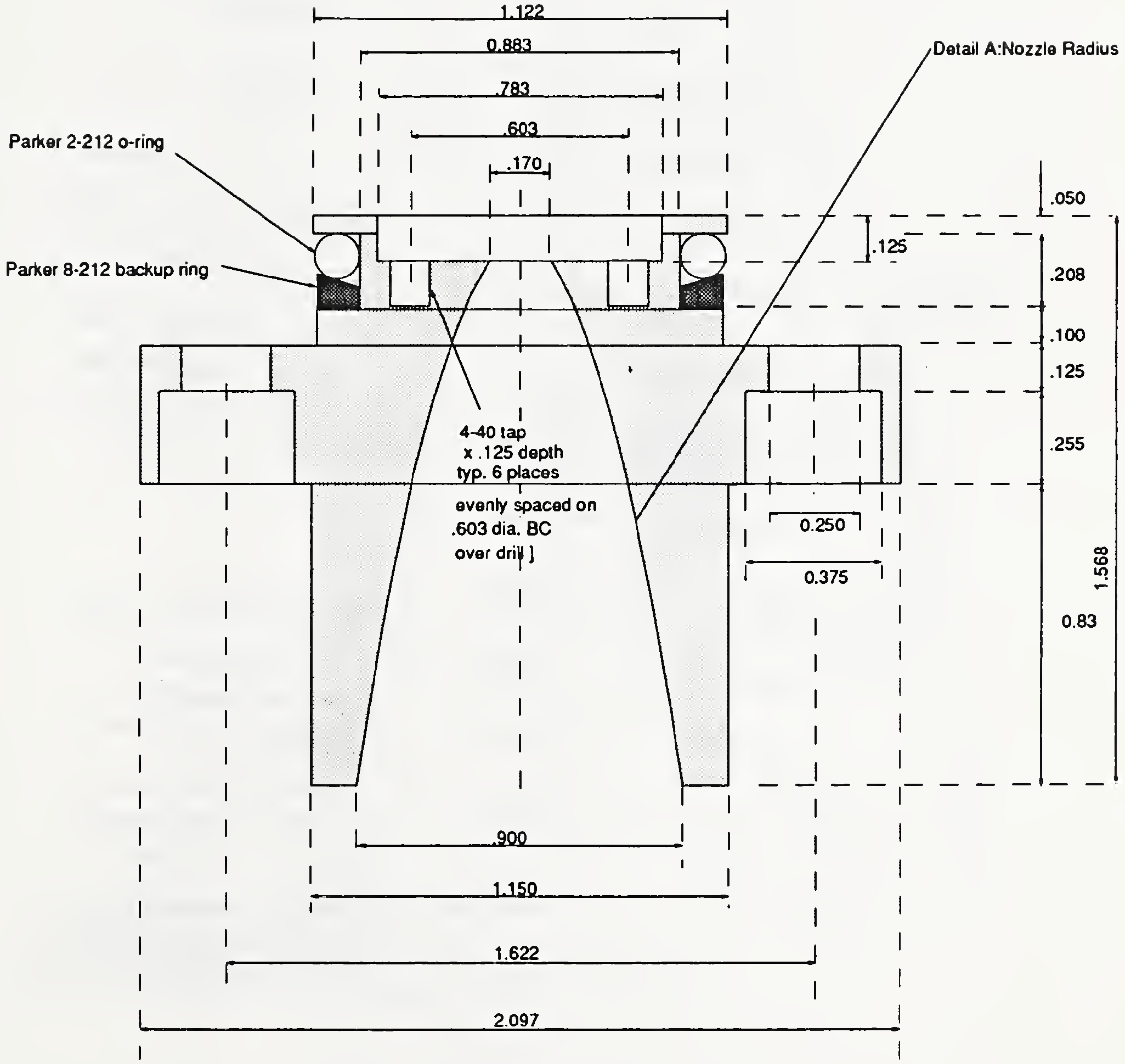
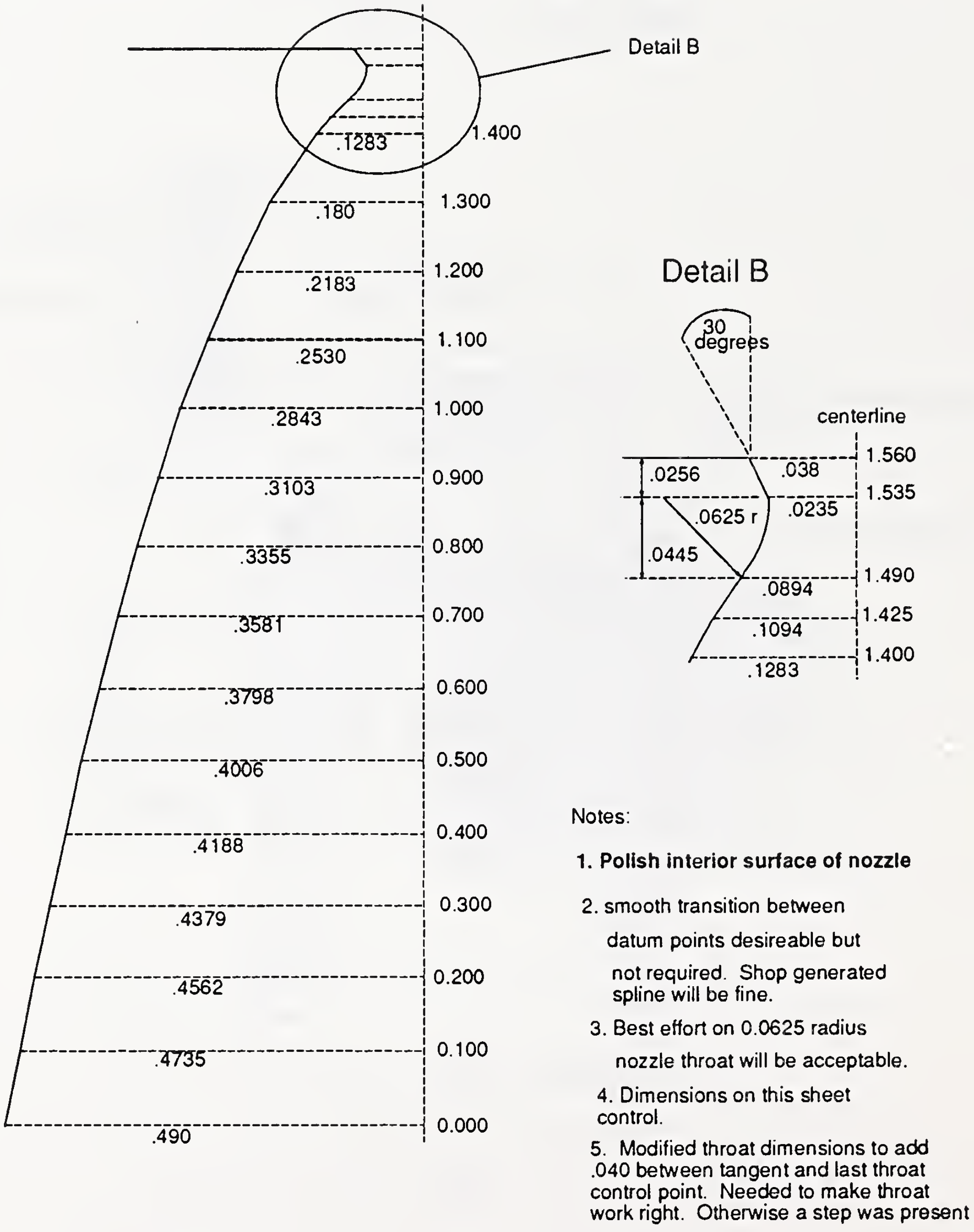


Fig. E.16: Removeable expansion nozzle with recepticle for replacement valve seats (which also comprise the nozzle throat). Units shown are inches; 1 inch = 25.4 mm.



Notes:

1. **Polish interior surface of nozzle**

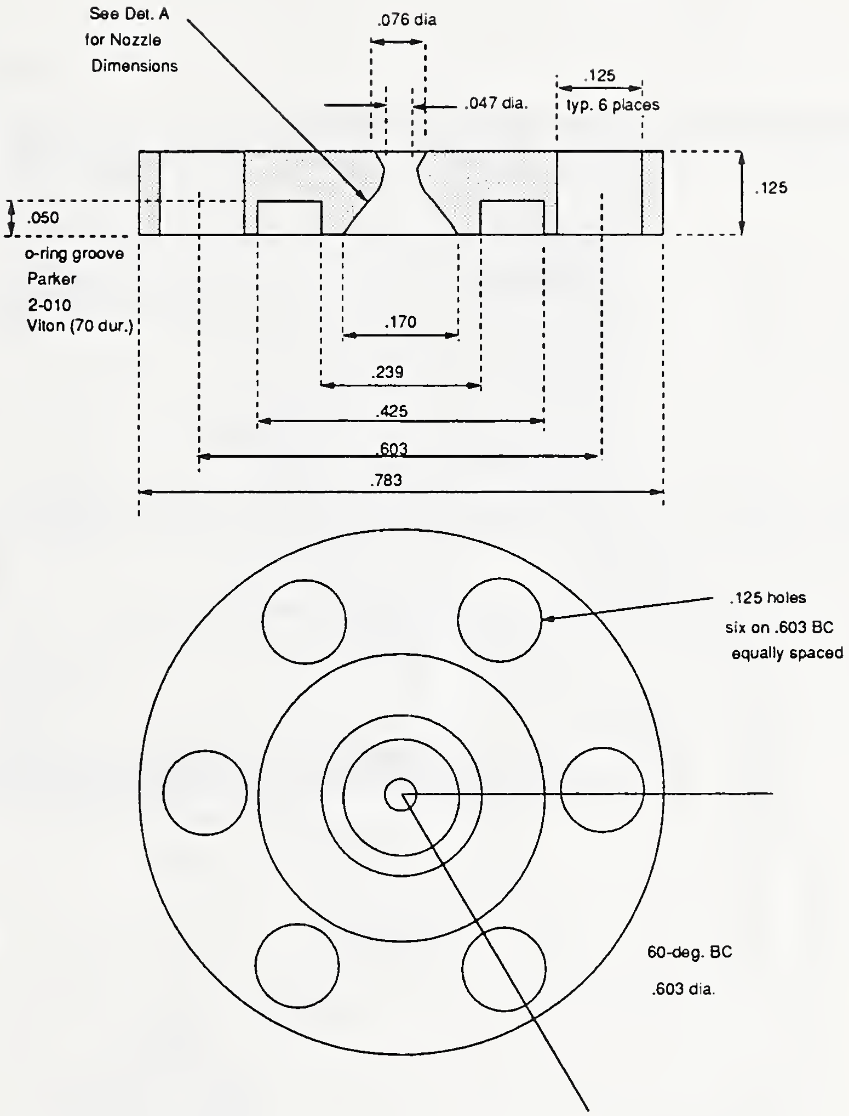
2. smooth transition between datum points desireable but not required. Shop generated spline will be fine.

3. Best effort on 0.0625 radius nozzle throat will be acceptable.

4. Dimensions on this sheet control.

5. Modified throat dimensions to add .040 between tangent and last throat control point. Needed to make throat work right. Otherwise a step was present

Fig. E.17: EDM milling profile for the interior of the expansion nozzle. Units shown are inches; 1 inch = 25.4 mm.



Notes:

1: Fabricate one copy from "Ferro-TiC" (Titanium Carbide)

2: Fabricate one copy from VascoMax 500

3: Contact Bill Stone x6075

4: See Detail sheet for nozzle dimensions; an existing EDM electrode exists

Fig. E.18: Valve seat insert. This mates to the expansion nozzle (Fig. E.17) and contains an internal o-ring seal. Two such units were made for prototype testing, one from VascoMax 500 maraging steel, a second from FerroTic, a titanium carbide alloy. Units shown are inches; 1 inch = 25.4 mm.

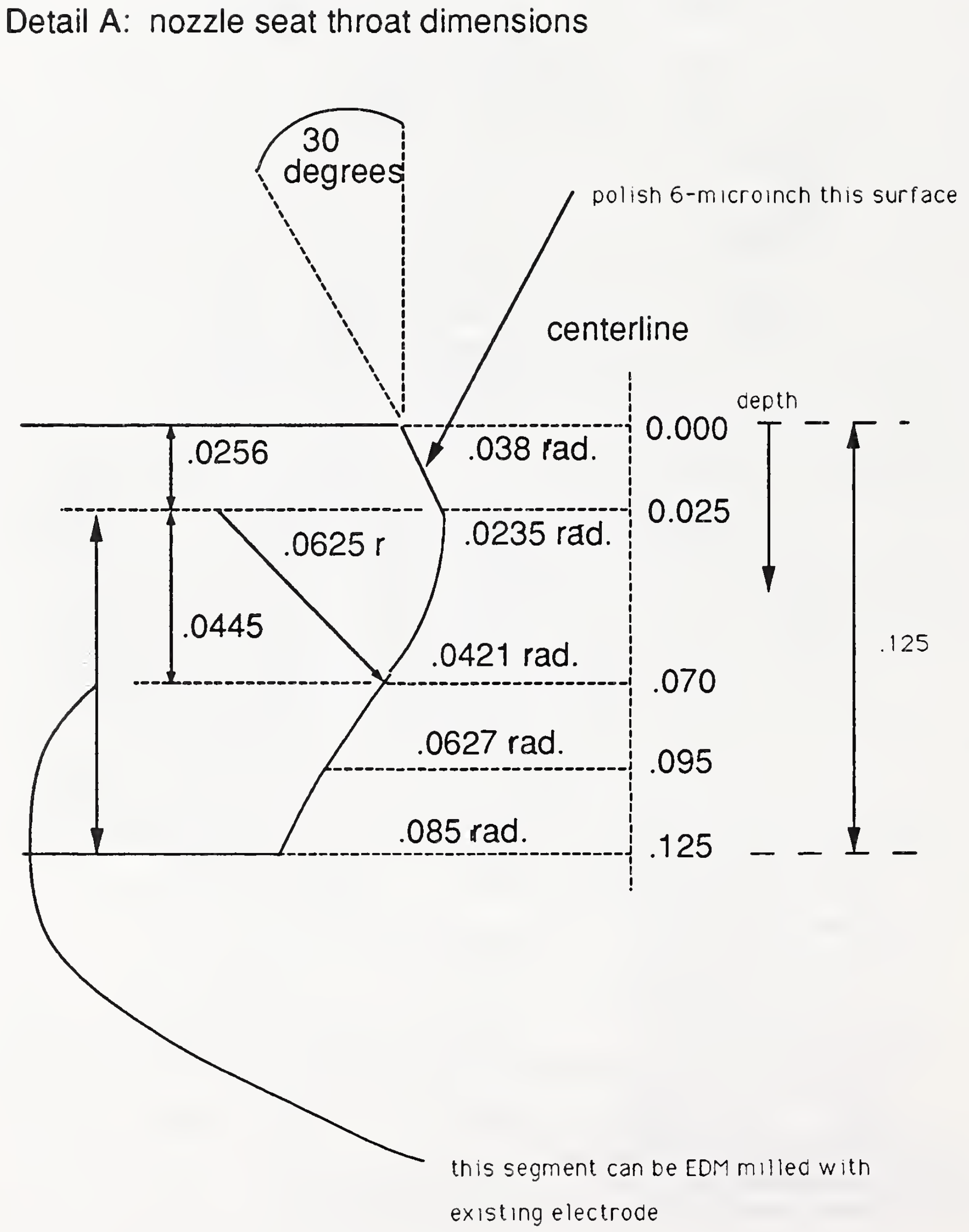

Detail A: nozzle seat throat dimensions
30 degrees
polish 6-microinch this surface
centerline
depth
0.000
.038 rad.
.0256
0.025
.0235 rad.
.0625 r
.0445
.125
.0421 rad.
.070
.0627 rad.
.095
.085 rad.
.125
this segment can be EDM milled with existing electrode

Fig. E.19: EDM milling details for the replaceable nozzle seat insert (see Fig. E.18). Units shown are inches; 1 inch = 25.4 mm.

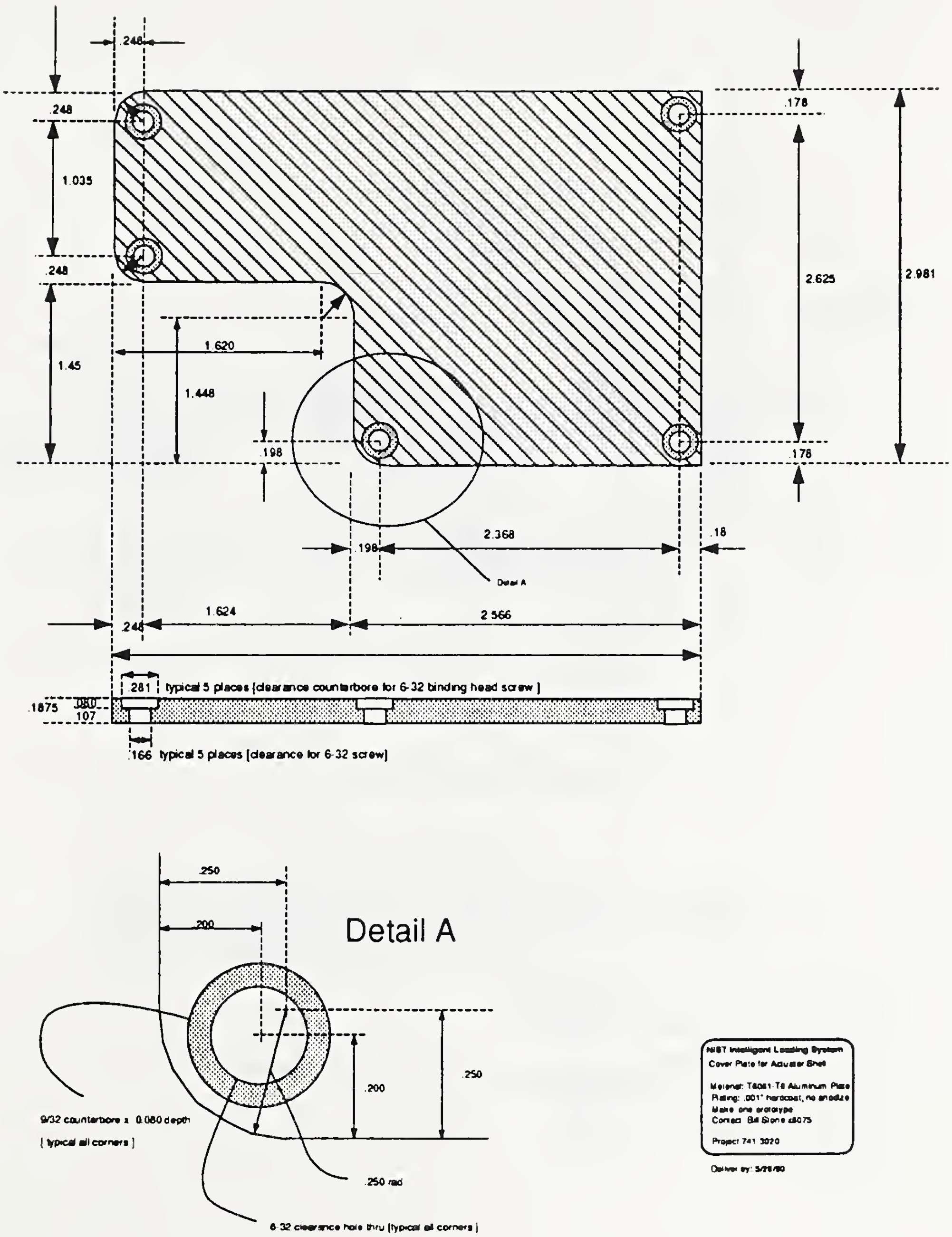
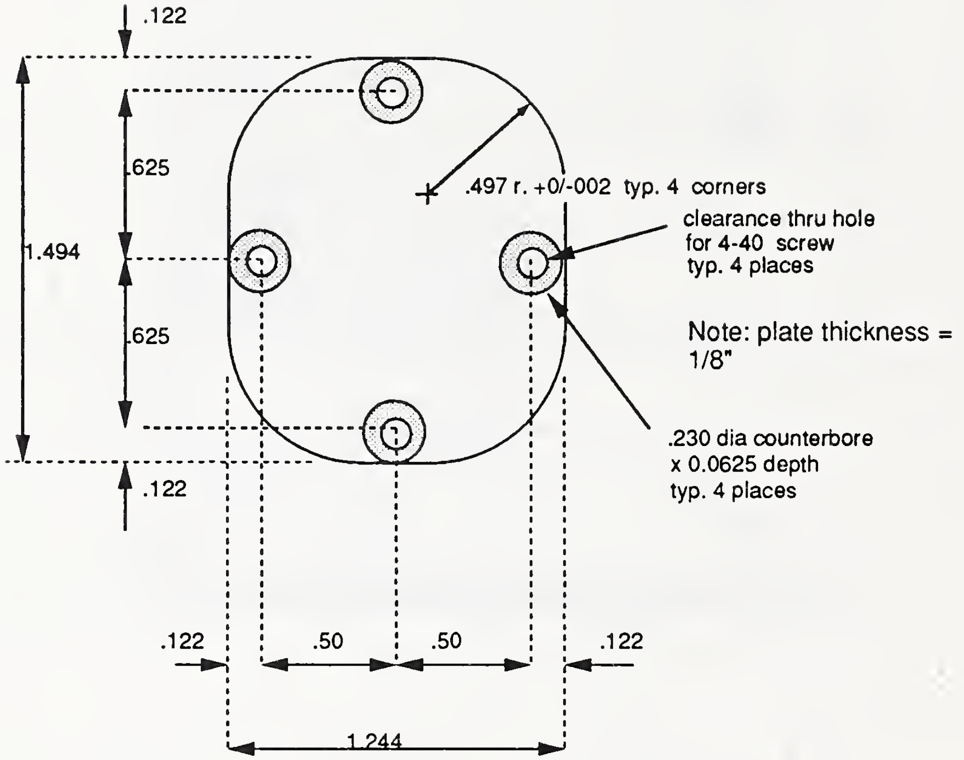

.248
.248
1.035
.248
1.620
1.45
1.448
.198
.178
2.625
2.981
.178
.198
2.368
.18
Detail A
.248
1.624
2.566
.281 typical 5 places [clearance counterbore for 6-32 binding head screw]
.1875
.080
.107
.166 typical 5 places [clearance for 6-32 screw]
.250
.200
Detail A
9/32 counterbore x 0.080 depth
[typical all corners]
.200
.250
.250 rad
6-32 clearance hole thru [typical all corners]
NIST Intelligent Loading System
Cover Plate for Actuator Shell
Material: T6061-T6 Aluminum Plate
Plating: .001" hardcoat, no anodize
Make one prototype
Contact Bill Stone x6075
Project 741 3020
Deliver by: 5/28/90

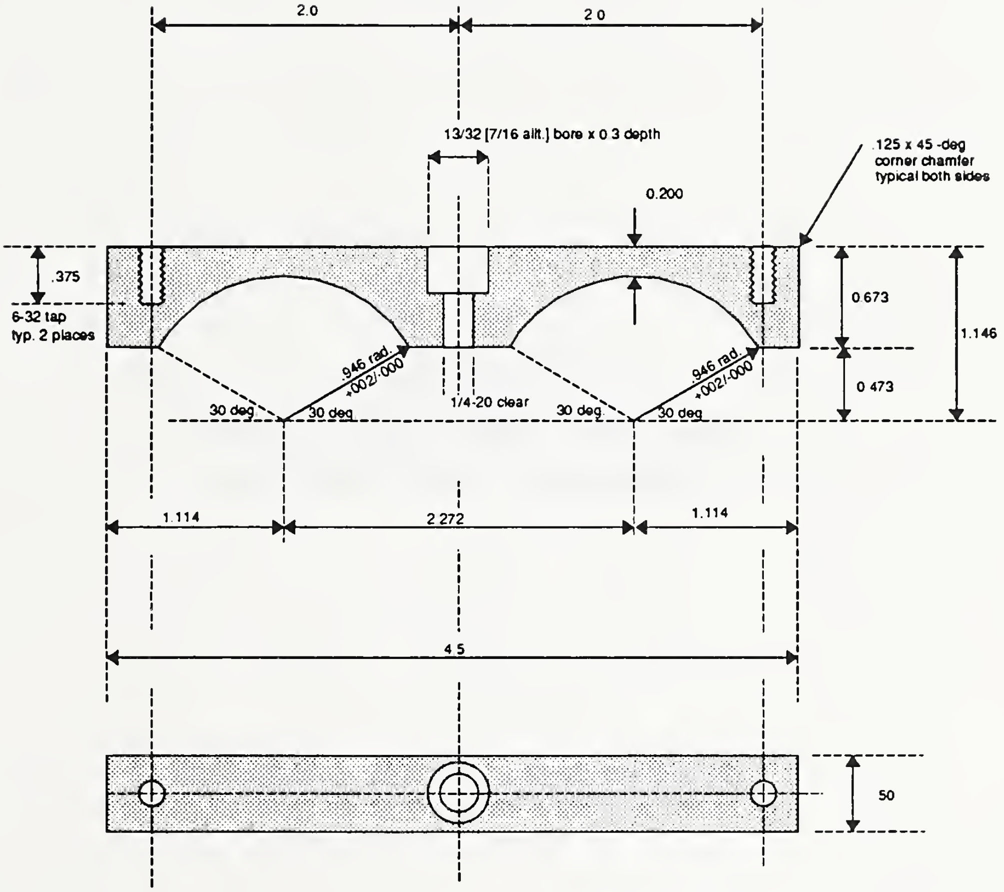
Fig. E.20: Main cover plate for the HSILS housing. Units shown are inches; 1 inch = 25.4 mm.



NIST High Speed Intelligent Loading System

Part: Access hole dust cover
Material: T6061 Aluminum; hardcoat .001 ; no anodize   Base material: 1/8 plate
Version 6/04/90
Contact: Stone x6075
Fabricate: 1 unit
Page 1/1

Fig. E.21: Secondary cover for HSILS housing. This allows access to the opposite end of the piezoelectric stack for initial assembly of the unit. Units shown are inches; 1 inch = 25.4 mm.



NIST Intelligent Loading System
Tank Restrainers

Material T6061-T6 Aluminum Plate
Plating .001" hardcoat, no anodize
Make: four (4) copies
Contact Bill Stone x6075
Sheet 1 of 1
Project 741-3020

Deliver by: 5/30/90

Fig. E.22: Gas accumulator attachment brackets. These bolt to the attachment holes shown in Figs. E.2 and E.3. Units shown are inches; 1 inch = 25.4 mm.

Introduction

In the actual hardware implementation of the HSILS prototype the onboard microcontroller and pulse power supply were housed in a grounded metal box. Various external connections, indicators, and switches were mounted on the front of the box and are shown in Fig. F.1. Onboard software was as described in Appendix C. The host PC contained a common commercial RS232 communications protocol program which permitted downloading of voltage-time profiles.

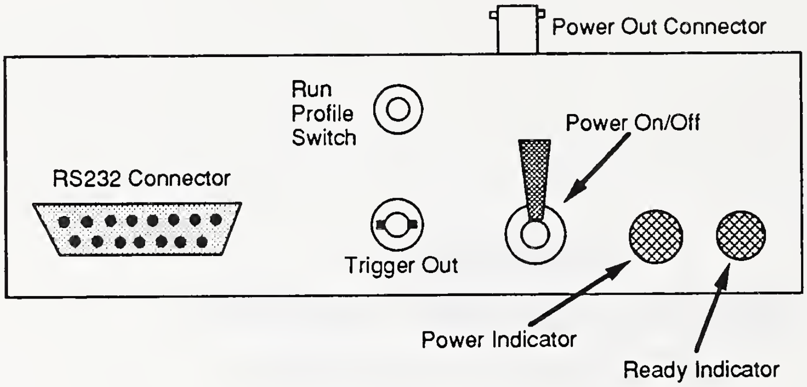


Fig. F.1: HSILS controller front panel.

The basic components shown in Fig. F.1 include:

- **RS232 Connector.** A 25 pin subminiature D female connector for interfacing to a host computer.
- **Run Profile Switch.** Pressing this button will cause the controller to execute a download profile.
- **Trigger Connector.** A BNC connector internally connected to an open collector transistor. This transistor turns on (grounding the center conductor) while a profile or pulse is being run. Use this output to trigger an oscilloscope or other recording device.
- **Power Switch.** Controls AC power to the unit. Also discharges the high voltage circuits when off.
- **Power Indicator.** A red LED which is lit whenever ower is applied and the unit is operating normally. If not lit when the power switch is on, check to make sure the unit is plugged in and that the fuse, on the rear panel, has not blown.
- **Ready Indicator.** This green LED will be on when the high voltage capacitors have reached normal charge. It will flash when a profile has been successfully loaded and the unit is ready to run.
- **Power Out Connector.** The output of the controller is output to the HSILS unit via this BNC connector directly to the piezoelectric stack.

Connections

The controller is configured as a DCE device and may be connected to the host computer using a standard RS232 cable. The communications parameters are 9600 baud, 8 bits, no parity.

Installing the Host Computer Program

The host computer program is a shareware communication program called Powercomm and is commonly available. Install this on your hard drive. Once the program is installed, you should print out and read the manual. Note that the program comes preconfigured to use COM1. You will have to reconfigure it (using the alt I command) if the controller is connected to a different communications port.

Testing the HSILS Controller

Once the controller has been hooked up, start powercomm in the "computer to computer" mode by typing:

```
C>HSILS
```

Now turn on the controller and the following prompt should appear on the host PC:

```
HSILS Controller V1.0
>
```

Press the enter key. The controller should respond with:

```
Valid Commands are:
U                   - Upload a profile
P width,height      - Issue a pulse
```

If the system did not behave as above check the RS232 cable, Powercomm configuration, and power connections. Looking at the controller box, the Power (red) LED should be lit and the Ready (green) LED should come on after a one to two minute delay (to allow for charging of the capacitor bank).

Operation

The controller software has two commands:

- **P width, height** causes the controller to pulse the output to height volts for width microseconds. This, in effect, creates a half square wave drive pulse. As described earlier in the report, the slope of this "square" wave will be approximately 1 volt/microsecond and is not capable of being modified without changing the pulse power amplifier design.
- **U** instructs the controller to upload a profile for later execution. A profile is a list of

time (microseconds) and voltage pairs terminated with a line containing the word "end". Any line starting with a semicolon (;) is ignored. Fig. F.2 lists part of a sample profile.

A profile can be entered manually, by simply typing it in, or it can be uploadeed from a pre-generated file. To upload a file, press ALT A. Powercomm will ask for the name of the file. Once the file name has been entered, Powercomm will upload it and signal when done. If the profile was successfully loaded, the ready LED should begin flashing. To run the profile, press the <u>Run Profile</u> button. When completed the microcontroller sends a "done" message to the host PC.

```
;This is a comment in a sample profile
30  25.82
60  49.38
90  70.87
120 90.43
150 108.20
180 124.34
.
.
.
9810   3.33
9840   3.28
9870   3.24
9900   3.19
9930   3.14
9960   3.09
9990   3.05
END
```

Fig. F.2: Sample download profile for local execution on the HSILS microcontroller.