University of Tennessee, Knoxville

TRACE: Tennessee Research and Creative Exchange

Doctoral Dissertations

Graduate School

12-1997

Investigation and Demonstration of a Rich Combustor Cold Start Device for Alcohol Fueled Engines

David Kim Irick
University of Tennessee - Knoxville

Recommended Citation

Irick, David Kim, "Investigation and Demonstration of a Rich Combustor Cold Start Device for Alcohol Fueled Engines. " PhD diss., University of Tennessee, 1997.
https://trace.tennessee.edu/utk_graddiss/1493

To the Graduate Council:

I am submitting herewith a dissertation written by David Kim Irick entitled "Investigation and Demonstration of a Rich Combustor Cold Start Device for Alcohol Fueled Engines." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Mechanical Engineering.

Jeffrey Hodgson, Major Professor

We have read this dissertation and recommend its acceptance:

Robert Bodenheimer, Ke Nguyen, Masood Parang

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by David Kim Irick entitled "Investigation and Demonstration of a Rich Combustor Cold Start Device for Alcohol Fueled Engines." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Mechanical Engineering.

J. L. Hodgson

Jeffrey Hodgson, Major Professor

We have read this dissertation
and recommend its acceptance:

Accepted for the Council:

Cw Minkel

Associate Vice Chancellor and
Dean of The Graduate School

Investigation and Demonstration of a Rich Combustor Cold Start Device for Alcohol Fueled Engines

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

David Kim Irick
December 1997

Dedication

This dissertation in dedicated to my daughter

Lauren Elizabeth Irick

who is my inspiration for this accomplishment.

Acknowledgments

This research was performed under Department of Energy Contract XCF-5-14050-03. I would like to thank the National Renewable Energy Laboratory for their support.

I would also like to thank my Dissertation Committee, Jeffrey Hodgson, Robert Bodenheimer, Ke Nguyen, and Masood Parang for their support and encouragement.

I would like to especially thank the chairman, Professor Hodgson. Over a period of twenty years, he has been my teacher, advisor, mentor, friend, and now colleague.

I am grateful for the support and advice from the staff of the Mechanical Engineering Shop, Steve Hunley, Danny Graham, Gary Hatmaker and Dennis Higdon. Prototype hardware development requires great support in fabrication, and these men provided it.

To my parents, family and friends, I owe thanks for understanding and encouragement during the past five years. The time not spent with family and friends is my only regret in this endeavor.

Finally, as I sit at the keyboard on our nineteenth wedding anniversary, I am most grateful to my understanding wife Carol. Without her, I could not have done this.

Abstract

Alcohols (methanol and ethanol) have been identified as having the potential to improve air quality when used to replace conventional gasoline. This potential is primarily due to the different organic species that are emitted by alcohol-fueled engines. The use of "near neat" alcohols gives greater benefits than fuels containing lower levels of alcohol, but neat alcohols present a significant cold starting problem.

The primary objective of this study was to develop a rich combustor device which will extend the cold start range of alcohol fueled engines to -30°C while reducing cold start emissions. In support of this objective a software model was developed which includes the thermodynamic operation of the device as part of a vehicle, considering engine parameters, vehicle parameters, and driving cycle requirements.

The analytical portion of this project consists of developing the software model. To understand the design and operating variables of the combustor, the model simulates operation of the combustor. The model predicts the output composition using the shifting equilibrium approach. The model includes engine and vehicle parameter inputs which will allow the simulation of a driving cycle. Chemical kinetics are not considered.

The experimental portion of this project includes design, fabrication, and testing of the rich combustor device(s). The combustor design and installation takes into

consideration all the criteria for proper vehicle operation such as requirements for actuating, shutdown, and phase-out of the device at different operating conditions. Because of the availability of cold temperature testing capabilities, including an engine dynamometer, and an engine identical to the engine in the test vehicle, the task of vehicle integration could logically be combined, to a great extent, with the testing and development of the prototype. We used the dynamometer driving the engine as a pump to test the combustor while the issues of fit, interfaces, and control were concurrently addressed. With this capability, a parallel approach allowed many of the problems associated with vehicle integration to be addressed early in the development of the combustor.

The development of the rich combustor was primarily targeted toward methanol; however, the device functions equally well using ethanol.

Preface

This project was originally defined by five tasks. The following task statements best describe them.

As the first task, a theoretical model shall be developed to determine what are the important combustor design variables. The model shall help determine the relationships between the combustor's inputs (fuel and air flow) and outputs (composition, temperature, and flow rate of exhaust products). The exhaust product flow rates needed to achieve adequate alcohol cold-start over a range of temperatures shall also be determined.

In the second task, using the understanding obtained from the modeling effort in the first task, a prototype combustor shall be designed and fabricated. The design shall include among other concerns: nozzle selection, air and fuel flow regulation, ignition system selection, material type, and fabrication method.

In the third task, the prototype rich combustor(s) shall be tested to determine its performance. The tests shall be done over a range of ambient temperatures and shall include performance mapping of the following parameters: fuel and air flow, output gas composition and temperature, and combustor temperature (at different locations on the apparatus). The evaluation shall initially be based on bench tests of the apparatus but shall include issues associated with vehicle applications such as air flow and pressure drops.

Based on the test results from the previous tasks, the prototype rich combustor device(s) shall be installed on vehicles for cold-start testing. The combustor and installation shall take into consideration all the criteria for proper vehicle operation such as requirements for actuating, shutdown, and phase-out of the device at different operating conditions. Based on the prototype rich combustor apparatus achieving satisfactory low temperature cold start results, the apparatus shall be tested to determine if the apparatus has any measurable effect on the vehicle's emissions and fuel consumption. The methanol fueled vehicle shall be emissions tested twice, using the rich combustor apparatus one of the times. The test cycle shall be determined by the operator but will need to take into consideration a low temperature cold start.

Because of the availability of cold temperature testing capabilities, including an engine dynamometer, and an engine identical to the engine in the test vehicle, the task of vehicle integration could logically be combined, to a great extent, with the testing and development of the prototype. We could use the dynamometer driving the engine as a pump to test the combustor while the issues of fit, interfaces, and control were concurrently addressed. With this capability, many of the problems associated with vehicle integration were addressed early in the development of the combustor. This parallel approach was taken rather than the original task breakdown which included bench testing. Due to a shift in the focus from methanol to ethanol, the vehicle emissions testing was replaced in the project with testing of the combustor using ethanol.

The results of the modeling, prototype design and fabrication, testing and development are presented for operation using methanol fuel. The conversion of the combustor system to use ethanol was simple and straightforward, and the results of testing using ethanol are also presented.

Table of Contents

List of Figures

List of Tables

Chapter I

Introduction

Background

Methanol (CH_3OH) has several properties which distinguish it as an attractive alternative to petroleum fuels. Its high octane rating and high latent heat of vaporization enhance its appeal as an alternative to gasoline in spark-ignition engines. Emissions from an alcohol fueled engine contain up to 58% less hydrocarbons (HC) and oxides of nitrogen (NO_x), compared to a gasoline-fueled equivalent (Höchsmann 1989). Compared to their gasoline equivalent, the emissions from methanol fueled engines are less photochemically reactive (Pefley 1984). The low volatility of methanol coupled with its high latent heat of vaporization reduce the evaporative emissions from a methanol fueled vehicle (Hodgson 1994). Together, these properties make it an attractive alternative fuel from an environmental perspective.

Cold Starting Issues

Unfortunately, the low vapor pressure and high latent heat of vaporization of methanol also lead to low vapor pressures at low temperatures which severely reduces the cold start performance of an alcohol fueled engine. This problem is compounded by the fact that the low air fuel ratio required for methanol means less air is available per unit mass of fuel in the fresh charge to supply the energy for vaporization. Unlike gasoline which can be blended to achieve desired phase equilibrium behavior (winter blends are more volatile than summer blends), methanol is a pure substance with a known fixed phase equilibrium behavior. The vapor pressure at various temperatures (Perry's 1993) is shown in Figure 1. The equivalence ratio assuming saturated vapor at the temperature considered, shown for methanol (Pefley 1986) in Figure 2, indicates the minimum (richest) achievable air/fuel ratio during cold starting.

The traditional manner in which an adequate air/vapor ratio is achieved for gasoline-fueled engines is to over-fuel the engine in order to assure that enough fuel evaporates. This is accomplished on carburetor-equipped engines by employing a choke, and on fuel-injected engines by a warm-up enrichment cycle. As is evident from Figure 2, over-fueling to provide an adequate air/vapor ratio is not possible for methanol below approximately 12°C.

This over fueling during cold start and drive away account for a substantial portion of the emissions generated during the U. S. Environmental Protection Agency (USEPA) Federal Urban Driving Schedule (FUDS) by a typical current model car. Due to over fueling, the exhaust is rich in unburned fuel and carbon monoxide. During this initial starting period, the catalyst is cold and ineffective. The net result is that the engine produces more undesirable emissions as a result of the over fueling at a time when the catalyst is unable to convert the emissions. Reducing the over fueling during cold starting would reduce the cold start emissions (Hodgson 1994).

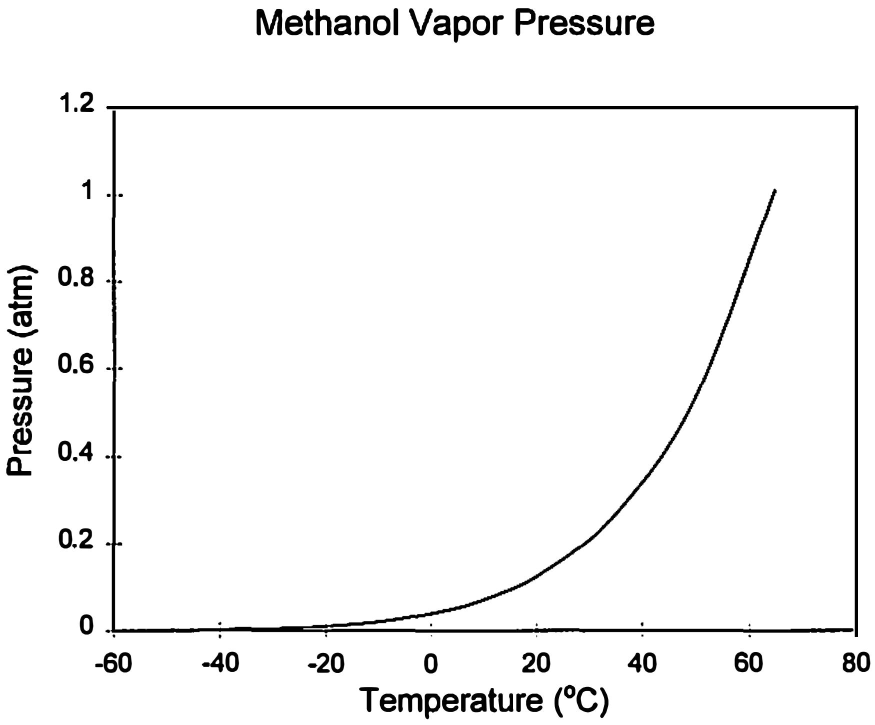


Figure 1 Methanol vapor pressure vs. temperature

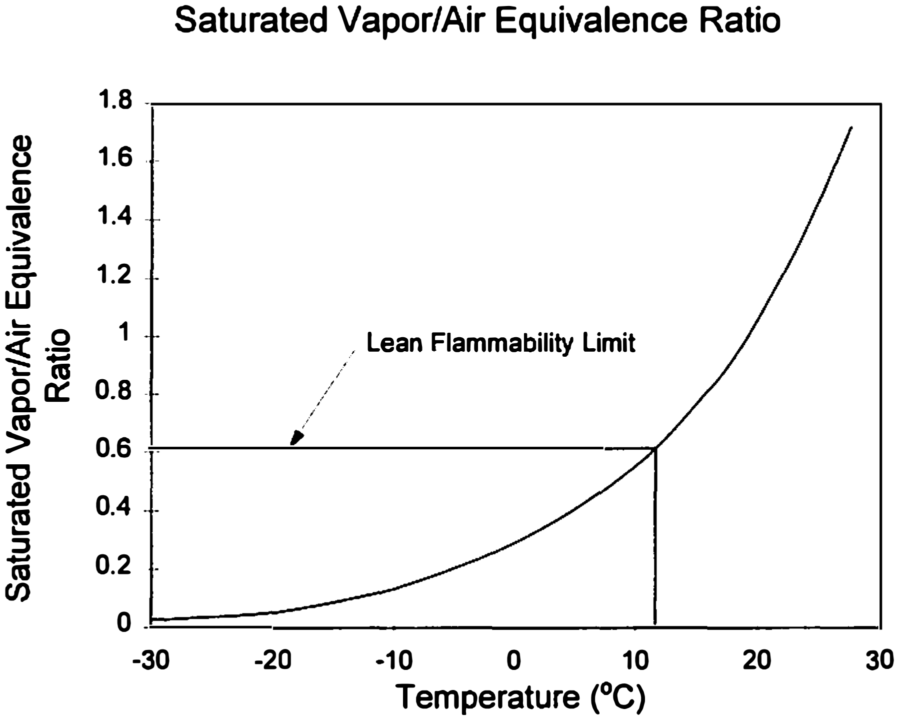


Figure 2 Saturated vapor/air equivalence ratio vs. temperature

Aside from the fuel and emissions issues concerning cold starting, there are other issues which must also be considered. The power required to crank the engine is increased due to the increased viscosity of the engine oil at low temperatures. This places a high current demand on the battery and therefore, the battery's ability to sustain cranking is reduced to the point where it becomes an issue during cold cranking and starting.

Cold Starting Solutions

Because the implementation of methanol as an automotive fuel may be limited by the cold start problems described above, much research has been done taking many approaches. Some of the approaches taken are described below (Hodgson 1994).

Supplementary Fuel

This approach involves using a volatile fuel, typically gasoline, to start the engine and warm it up, after which the engine is switched to the alcohol fuel. Obviously, this approach requires auxiliary fuel storage and delivery systems. On-board generated hydrogen has also been used (Hydrogen Consultants 1995) which requires heaters and catalysts.

Blended Fuel

This approach uses a volatile fuel as a blending agent to provide the required air/vapor ratio for starting. M85 (85% methanol, 15% unleaded gasoline) is an example of this approach. The advantage of this approach is that a single fuel system can be used, but the disadvantage is that the advantages of methanol described above are compromised by the addition of the blending fuel.

Engine Modifications

Many approaches to engine modifications have been taken to extend the cold start range of alcohol fueled engines. High energy ignition systems, "prompt EGR" (Gardiner 1993), and optimization of injection parameters (Höchsmann 1989) are among the research efforts in the field. The most effective of these (prompt EGR) requires internal modifications to the engine (installation of a special camshaft).

Air Heating

Two approaches have been taken to heating the intake air. A combustor was used to provide its exhaust to the hot side of a "air to air" heat exchanger located immediately upstream of the intake ports and injectors, with the engine intake air as the cool side (Höchsmann 1989). This combustor was fueled with liquid methanol from the fuel tank; a glow plug served as the ignition system. The air flow through the combustor was provided by an external fan.

The second approach taken (Hodgson, et al 1993) was to burn liquid methanol directly in the intake air stream. The air/fuel ratio was maintained very lean such that sufficient oxygen remained in the intake stream to support combustion in the engine.

In both cases, it was determined from cold start testing that the intake fresh charge would cool due to contact with the cold engine surfaces to the point that a combustible mixture was not present at the spark event.

Fuel Heating

Efforts to utilize fuel preheating have not significantly extended the cold start range of methanol (Höchsmann 1989). This results from the same cooling of the fresh charge that limits the contribution from air heating.

Fuel Reforming

In this approach, the fuel is reformed by various processes to produce other combustible species. Two approaches taken have been reforming methanol into dimethyl ether (Karpuk 1988), dissociating methanol into hydrogen and carbon monoxide (Karpuk 1989) and steam reforming (Lalk 1984).

All of these approaches require heat for the fuel and the catalysts to produce the reformed fuel. The required heat is not available during cold start and warmup, so the fuel must be reformed when operating temperatures are sufficient and stored on the vehicle for later use. This requires the additional systems described for the supplementary fuel approach.

Rich Combustion

This is the approach taken in this study. Liquid methanol is burned in a rich diffusion flame in the combustor to produce the non-condensable gases hydrogen and carbon monoxide. These combustible gases are then mixed with additional air and fed to the engine. Proof of concept testing has been conducted by multiple researchers (Hodgson 1993), (Iwai 1994) and (Pettersson 1991) which shows that rich combustion or partial oxidation is a viable means to extend the cold start range of alcohol fueled engines. This study addresses the development and application of this technology to a methanol fueled *vehicle*.

Chapter II

Modeling

Combustor Model

To understand the design and operating variables of the combustor, a design tool has been developed to simulate constant pressure adiabatic combustion, including dissociation of the products. The model uses the thermodynamic and reaction data, and follows the methodology, presented in Campbell (1979) to determine the composition and temperature of the products. Chemical kinetics are not considered.

The model performs an enthalpy balance of the reactants and products, using the thermodynamic data from Campbell. Dissociation of the product gases is included in the enthalpy balance; therefore, the final temperature (constant pressure adiabatic flame temperature) and equilibrium composition of the combustor products is determined using an iterative scheme. The following species are considered in the dissociated products- CO, CO_2, H_2, H, H_20, 0_2, O, OH, N_2, and NO. Chemical equilibrium among these species is governed by conservation of the elements C, O, H, and N, plus the following six equilibrium reaction equations:

$$CO + \frac{1}{2}O_2 \xleftrightarrow{k_1} CO_2$$

$$2\mathrm{O} \xleftrightarrow{k_2} \mathrm{O_2}$$

$$\mathrm{O_2} + \mathrm{N_2} \xleftrightarrow{k_3} 2\mathrm{NO}$$

$$2\mathrm{H} \xleftrightarrow{k_4} \mathrm{H_2}$$

$$\mathrm{O} + \mathrm{H} \xleftrightarrow{k_5} \mathrm{OH}$$

$$\mathrm{H_2} + \frac{1}{2}\mathrm{O_2} \xleftrightarrow{k_6} \mathrm{H_2O}$$

The reaction equilibrium constants k_i are functions of temperature; therefore, a product temperature is assumed, and the ten equations governing chemical equilibrium are solved using the algorithm presented in (Hodgson 1990). With an assumed product temperature and a calculated equilibrium composition, the enthalpy of the products is compared to the enthalpy of the fuel and air mixture. If the difference in the two enthalpy values is greater than the desired convergence criteria, a new product temperature is assumed. The product temperature and equilibrium composition are determined iteratively in this manner.

The combustor model allows examination of methanol, ethanol, n-octane, and other fuels not considered in this project. The air/fuel ratio and the temperatures of the fuel and air may be separately specified.

Calculations have been made using the model to simulate combustion of rich mixtures of both methanol and ethanol with air at -30°C to determine the concentration of hydrogen and carbon monoxide in the product gases under the equilibrium assumption; the results are shown in Figures 3 and 4. The calculated results for methanol are in good agreement with the calculated results presented in (Iwai 1984), which compare well with the experimental results that were presented.

Although it appears that extremely rich combustion (equivalence ratios >2) would give more hydrogen and carbon monoxide in the product gas, experimental results of Iwai showed that this did not occur. It is believed that this is due to non-homogeneous combustion. Iwai concluded that an equivalence ratio of 2 was appropriate.

As detailed in the following section on the engine model, the combustor product stream contains significant inert or non-combustible species that results in the charge being fed to the engine having a lower energy density than the charge with the rich combustor operating.

Engine Model

The design tool includes an engine model that simulates the use of the combustor products, as well as the primary fuel, for fueling the engine. This model follows the

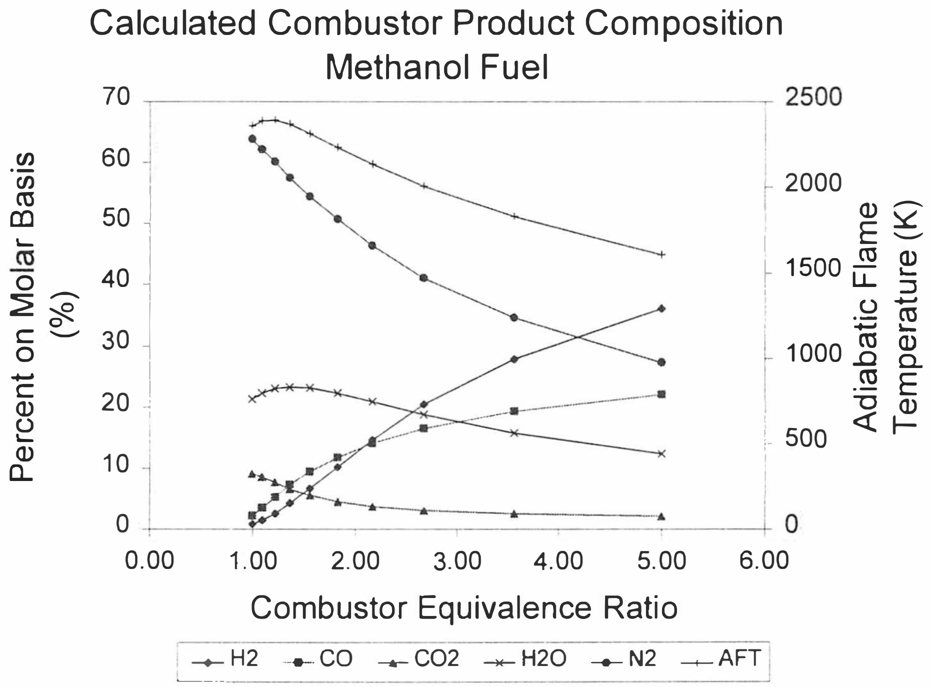


Figure 3 Product composition for methanol fueled combustor, 1 atm pressure

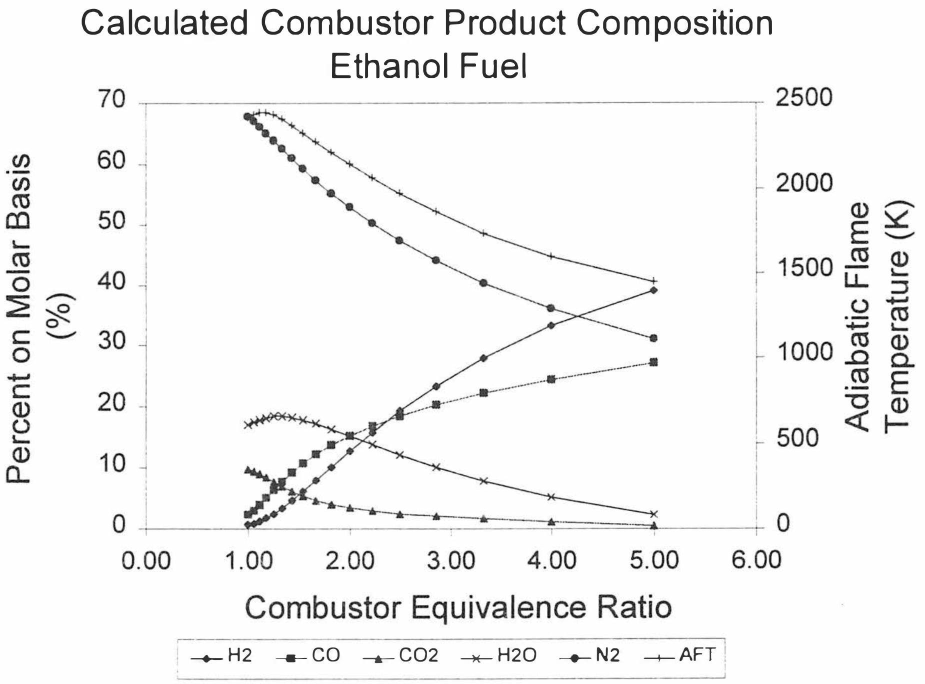


Figure 4 Product composition for ethanol fueled combustor, 1 atm pressure

same methodology as the rich combustor model for determining the thermodynamic properties and product composition. The four-stroke Otto cycle is modeled including the effects of progressive combustion (spark advance and combustion duration), volumetric efficiency, and part-throttle operation to calculate the range of output of the engine using engine geometry, engine volumetric efficiency vs. engine speed/manifold pressure tables, and ambient conditions as inputs.

The engine model requires a selection of fuel, equivalence ratio for the engine, and fueling strategy- primary fuel or rich combustor operation. Ambient temperature and pressure must be specified, as well as the pertinent engine dimensions- bore, stroke, compression ratio, connecting rod length, and number of cylinders. A lookup table approach was chosen for incorporating the effects of volumetric efficiency rather than calculating the flow rate through the engine based on engine characteristics and operating conditions. The latter approach requires knowledge or estimation of many engine dimensions and operating parameters as well as numerical modeling of their effects on volumetric efficiency. The lookup table approach requires only that the candidate engine's volumetric efficiency be calculated from flow rates measured as a function of engine speed and manifold pressure. The volumetric efficiency data file required by the model contains volumetric efficiency for operating points from 100 to 4000 RPM and 65 to 100 kPa, with operating point spacing of 100 RPM and 5 kPa.

The four stroke Otto cycle is used as the basis for the thermodynamic engine model. Valve events are modeled as occurring instantaneously at top dead center (TDC) or bottom dead center (BDC), with the exception of the intake closing event. This event is used to include the volumetric efficiency effect of reduced fresh charge. The theoretical fresh charge volume (volume at BDC minus volume at TDC) is multiplied by the volumetric efficiency to determine the volume of the actual fresh charge at intake manifold temperature and pressure. The engine geometry is then used to determine the crank angle (degrees after BDC) at which the intake valve should close corresponding to that volume. In this manner, the volumetric efficiency effects of port flow area, valve timing, exhaust gas recirculation (EGR), and other parameters too numerous to list are all incorporated in the volumetric efficiency lookup table and are all modeled by the late closing intake valve feature.

The intake manifold conditions, except pressure, are assumed to be the same as ambient. In the case of primary fuel use, both the fuel and the air are at the specified ambient temperature. If operation of the rich combustor is chosen, then the products are assumed to mix with air and be cooled to ambient temperature in the intake manifold as described below.

The constant volume combustion of the Otto cycle is replaced by progressive combustion in the engine model with spark advance and combustion duration as required inputs. A linear model of the burning rate is used; the portion of the fuel burned per degree of crankshaft rotation is assumed constant. The products of combustion are modeled using the shifting equilibrium method; chemical equilibrium among the combustion products is maintained throughout the cycle. Equilibrium composition is calculated at each state point based on the temperature, pressure, and element abundancies.

With engine speed and manifold pressure as inputs, and the lookup table used to determine volumetric efficiency, the engine model calculates the state point temperatures around the cycle. The temperature at the end of the intake stroke is used as the test for convergence; the model usually converges within three or four iterations. With the temperature and equilibrium composition at each of the state points determined, the thermodynamic data can be evaluated to calculate the indicated work for the cycle. The cycle calculations are based on one mole of fuel, which corresponds to a very large displacement engine. The actual displacement, as determined from the engine dimensions, is then used to proportionally reduce the calculated indicated work and amount of fuel to the appropriate values for the actual engine considered. The engine speed is then used to calculate the indicated power and fuel consumption rate. Engine friction power, including accessory load, is calculated using the following equation suggested by (Ross 1993).

$$\text{Friction Power} = \eta_{th} * .00567 * \text{Engine Speed} * \text{Engine Displacement}$$

where η_{th} is the thermal efficiency.

It is noteworthy for the current study that this equation does not include temperature effects. The friction power of a cold engine will be higher than that predicted by the equation. The engine friction was measured at various temperatures for the test engine. Although this data is engine specific, the trend should hold for other engine configurations and is included in the model. The engine brake power is then evaluated by subtracting the friction power from the indicated power. At each operating point (engine speed and manifold pressure), the following are calculated:

- Brake power output
- Fuel consumption
- Efficiency
- Cycle temperatures
- Cycle pressures
- Exhaust composition (equilibrium)

A preliminary analysis of the indicated output of the engine using the rich combustor versus using the primary fueling system gives a basis for evaluating the results of the model. An analysis of the two modes of operation based on the heating value of the fuels, treating all gases as ideal, should yield a good approximation of the relative values of the indicated output of the engine. The analysis is presented for methanol for comparison to the model prediction; calculations for ethanol are similar.

Assuming that the engine indicated torque is proportional to the heating value per mole of intake charge (proportional to the heating value per unit volume of intake charge), the following analysis can be used to estimate the effect on engine indicated torque using the rich combustor to fuel the engine.

From Obert (1973), the constant volume lower heating values of interest are:

$$\begin{aligned}
\mathrm{LHV}_{\mathrm{CH,OH}} &= 292089 * 2.326 = 679399\ \mathrm{kJ / kmol} \\
\mathrm{LHV}_{\mathrm{CO}} &= 103500 * 2.326 = 240741\ \mathrm{kJ / kmol\ (est)} \\
\mathrm{LHV}_{\mathrm{H_2}} &= 103435 * 2.326 = 240589\ \mathrm{kJ / kmol}
\end{aligned}$$

Operation of the rich combustor with an equivalence ratio of 2.0 gives the following calculated product mixture:

$$\mathrm{CH_3OH} \rightarrow .76\ \mathrm{CO}\ +\ .7325\ \mathrm{H_2}\ +\ 4.33\ \text{non - combustibles}$$

Therefore based only on considerations of the heating value per mole of fuel, the indicated output of the engine using the rich combustor products for fuel is reduced by the following factor.

$$C_{LHV} = \frac{(.76 * \mathrm{LHV}_{\mathrm{CO}} +\ .7325 * \mathrm{LHV}_{\mathrm{H_2}}) / 1.4925}{\mathrm{LHV}_{\mathrm{CH,OH}}} = .354$$

This reduction in indicated output accounts only for the lower heating value of the rich combustor products versus the heating value of the methanol. The indicated output is further reduced due to the fact that the total number of moles in the fresh charge for induction into the engine is greater using the rich combustor. For conventional operation under stoichiometric conditions, the total number of moles in the fresh charge is :

$$CH_3OH + 1.5\,(O_2 + 3.76\,N_2) = 8.14 \text{ moles}$$

For operation using the rich combustor with the engine operating at stoichiometric, the total number of moles in the fresh charge is:

$$.76\,CO + .7325\,H_2 + 4.33 \text{ non-combustibles} + .75\,(O_2 + 3.76\,N_2) = 9.39 \text{ moles}$$

Assuming that the indicated output of the engine is proportional to the energy density and that the fresh charge using the rich combustor is cooled to the same temperature as the fresh charge for the conventional fueling, the indicated output of the engine is further reduced by the factor:

$$C_m = 8.14/9.39 = .867$$

Therefore, assuming the incoming fresh charge is at the same temperature for the two modes of operation, the indicated output of the engine using the rich combustor is only 30.7% of the indicated output of the engine using the conventional fueling system. Further, if the fresh charge using the rich combustor is not cooled to the same temperature, the indicated output of the engine is reduced by a temperature factor based on the ideal gas law:

$$C_T = \frac{T_{\text{combustor}}}{T_{\text{conventional}}}$$

For the case considered, the uncooled fresh charge using the rich combustor would be approximately 1100 K, which would give an effective indicated output of only approximately 7% of a conventionally operated engine. Fortunately cooling of the rich combustor products does occur and is, in fact, the reason for considering the rich combustor rather than a simple air heater (Hodgson 1993). The results obtained from the design tool agree with this preliminary analysis.

Several operating points were examined using parameters obtained from the test vehicle to compare the *brake* output of the engine for the two modes of operation. Typical results are presented in Figure 5. These results include the low-temperature friction modification to the engine friction calculation.

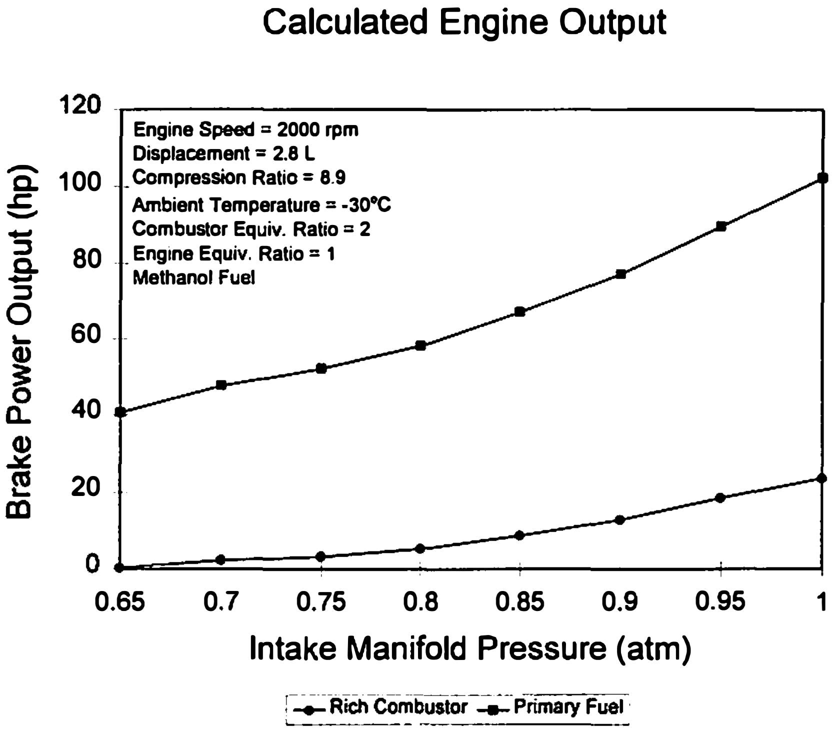


Figure 5 Calculated engine output for both modes of fueling

Vehicle Model

The design tool also includes a vehicle model for determining the road load power requirements for the driving cycle considered. The vehicle model operates in discrete time steps corresponding to the data for the prescribed driving cycle. For the Federal Urban Driving Schedule (FUDS), the time steps are one second. The vehicle model assumes that the engine is idling at *time = 0* and does not address cranking and starting. The model also assumes that the engine is idling during braking. Shifting of the transmission is per the default specifications of the driving cycle.

The vehicle parameters required by the model are vehicle mass, tire size, transmission gear ratios, and axle ratio. The model calculates the engine speed and the engine power required as a function of vehicle speed and acceleration. The engine speed is determined from the vehicle speed, tire size, transmission gear ratio, and axle ratio. The power required at the drive axle for constant speed is calculated using the following equation suggested by Obert(1979).

$$\mathrm{hp} = \frac{V}{375}(0.0148\,W + 0.00128\,AV^2$$

where

V = Vehicle Speed (mph)
W = Vehicle Weight (lbs.)
A = Frontal Area (ft^2)

The power required at the drive axle for acceleration is calculated using

$$\text{Power} = MV\frac{dV}{dt}$$

where

V = Vehicle Speed
M = Vehicle Mass corrected for rotating inertia
$\frac{dV}{dt} = \frac{V_{t+1} - V_t}{\Delta t}$

The total power required at the drive axle is then the sum of the two calculated values. For calculating the required engine brake power, the vehicle model assumes a 70% efficient driveline. Figure 6 shows the engine speed and Figure 7 shows the engine brake power required for the first 505 seconds of the FUDS calculated for the test vehicle, a 1988 Chevrolet Corsica.

Integrated Model

The three models described above are integrated into a single software package- “Thermodynamic Engine Model with Rich Combustor” written in Microsoft Visual

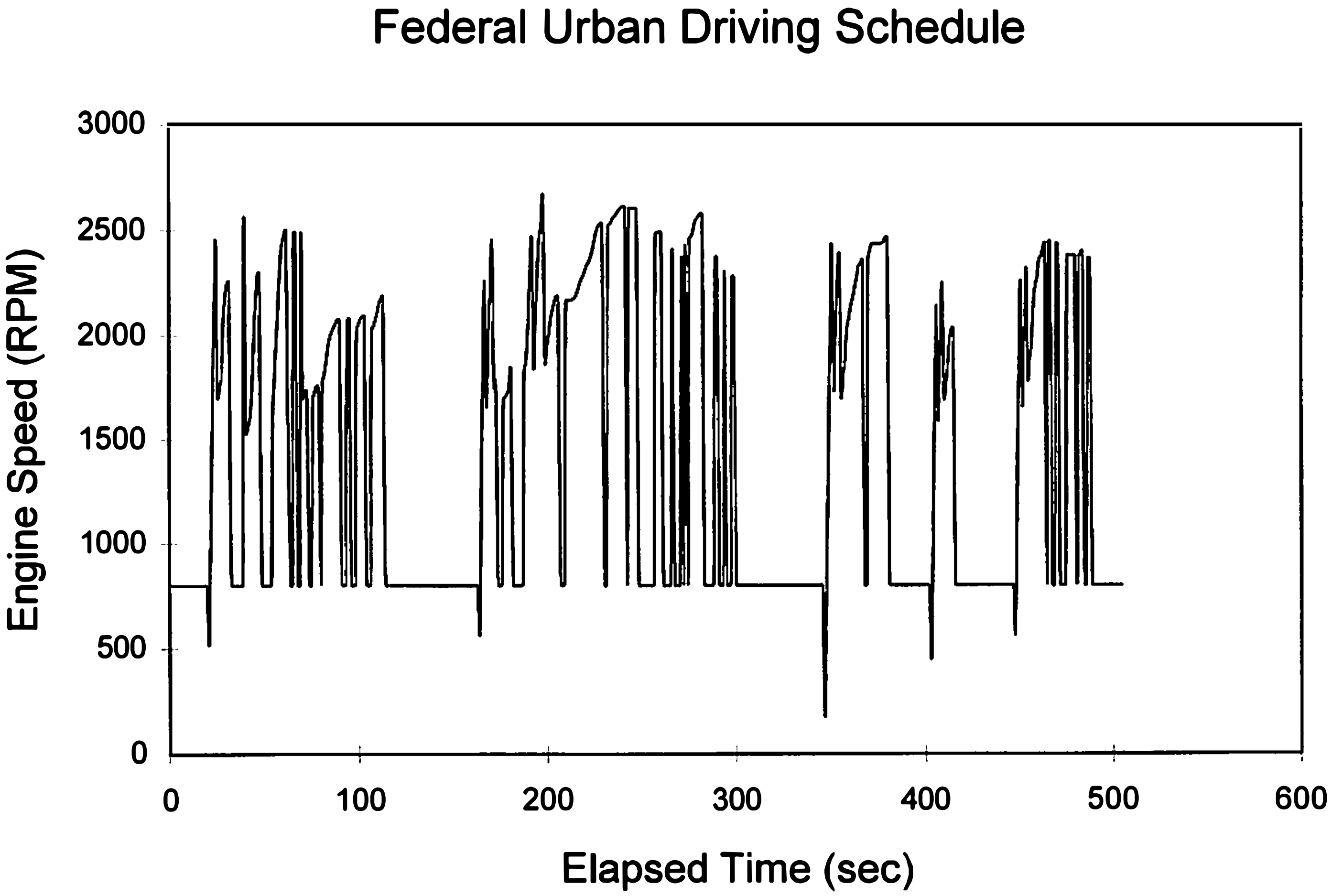


Figure 6 Engine speed for FUDS 505

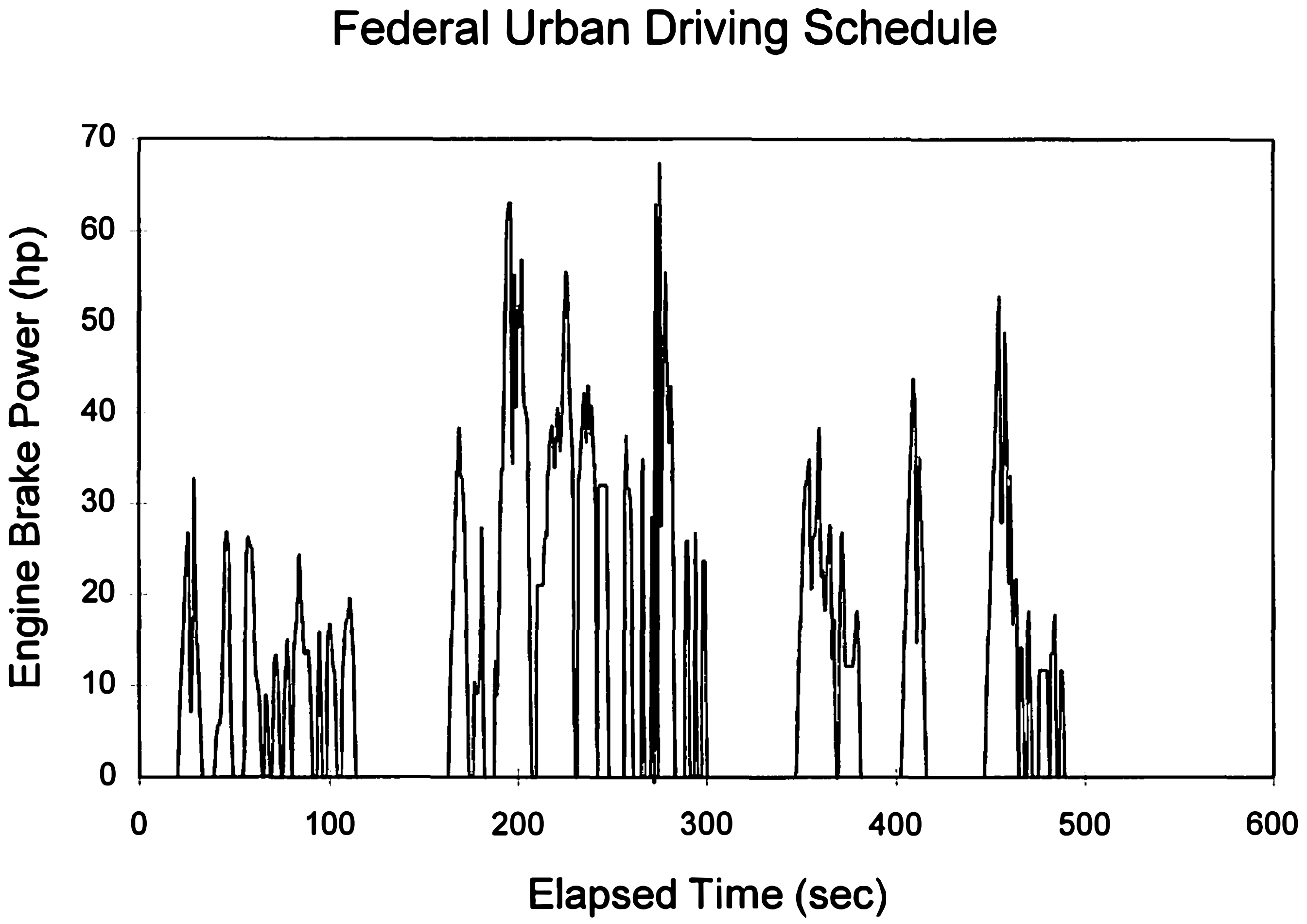


Figure 7 Engine power required for FUDS 505

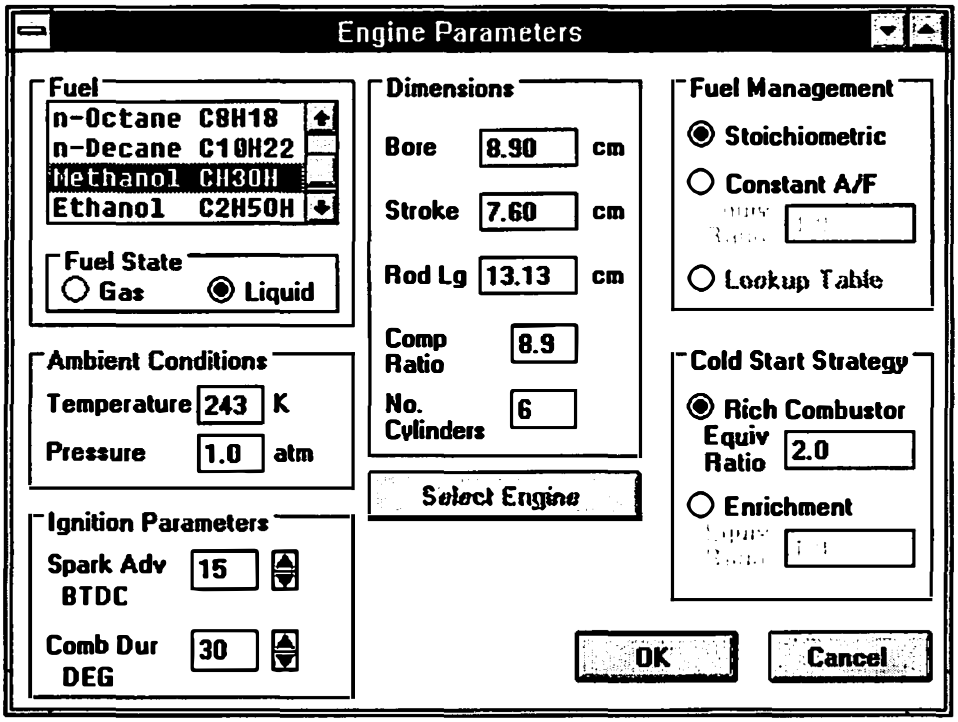
Basic$_{®}$ for Windows$_{TM}$. A source code listing is included as Appendix A. The logistics of the model are as follows.

Through the Engine Parameters dialog box, shown in Figure 8, the user specifies the ambient conditions, fuel, fueling strategy, rich combustor equivalence ratio (if applicable), engine equivalence ratio, engine dimensions, and ignition parameters.

If rich combustor operation is specified, the combustor model calculates the composition of the combustor output which is used to fuel the engine. Incorporating the user specified volumetric efficiency data file, the engine model calculates the performance map for the engine as specified in the Engine Parameters section. The results of this calculation are stored in a lookup table; engine brake power and fuel consumption rate are stored vs. engine speed and intake manifold pressure at 100 RPM and 5 kPa increments. Figure 9 shows a three dimensional plot of the calculated performance map for the test engine with the rich combustor providing the fuel., including the low-temperature friction calculation.

Through the Chassis Parameters dialog box, shown in Figure 10, the user specifies the vehicle weight, tire size, axle ratio, and transmission ratios. Using the chassis parameters and the specified driving cycle, the vehicle model calculates the engine speed and required engine brake power for each time step of the driving cycle.

For each time step in the driving cycle, the operating point of the engine is determined in the following manner. The engine speed is considered first. Since the performance map lookup table has data every 100 RPM, the two engine speed entries which bracket



Engine Parameters

Fuel

n-Octane C8H18
n-Decane C10H22
Methanol CH3OH
Ethanol C2H5OH

Fuel State
○ Gas ● Liquid

Ambient Conditions

Temperature 243 K

Pressure 1.0 atm

Ignition Parameters

Spark Adv BTDC 15

Comb Dur DEG 30

Dimensions

Bore 8.90 cm

Stroke 7.60 cm

Rod Lg 13.13 cm

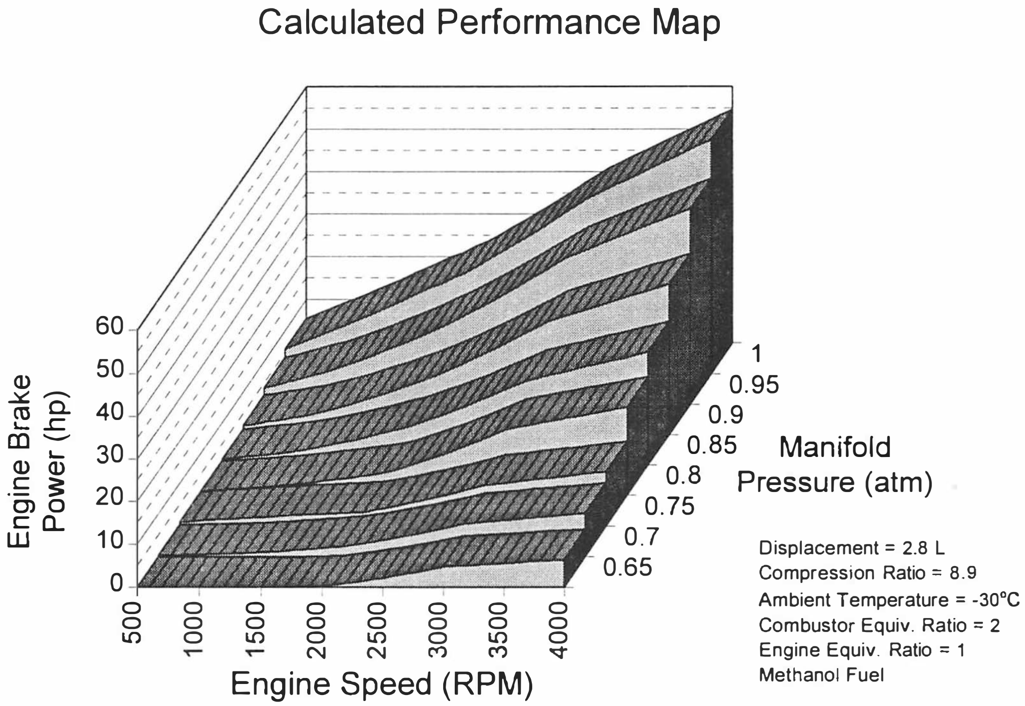
Comp Ratio 8.9

No. Cylinders 6

Fuel Management
● Stoichiometric
○ Constant A/F
○ Lookup Table

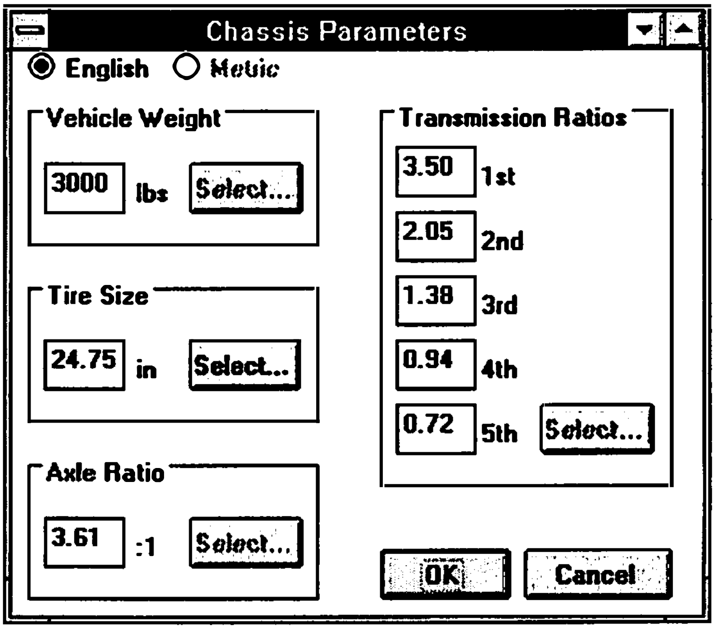
Cold Start Strategy
● Rich Combustor
Equiv Ratio 2.0
○ Enrichment

Figure 8 Engine parameters dialog box



Displacement = 2.8 L
Compression Ratio = 8.9
Ambient Temperature = -30°C
Combustor Equiv. Ratio = 2
Engine Equiv. Ratio = 1
Methanol Fuel

Figure 9 Calculated performance map for test engine operating on rich combustor



Chassis Parameters

● English ○ Metric

Vehicle Weight: 3000 lbs

Tire Size: 24.75 in

Axle Ratio: 3.61 :1

Transmission Ratios

3.50 1st

2.05 2nd

1.38 3rd

0.94 4th

0.72 5th

Figure 10 Chassis parameters dialog box

the operating point are determined. Then for each of the two engine speed entries, the brake power values are determined which bracket the required brake power calculated by the vehicle model. The operating point is then found by linearly interpolating in both directions (engine speed and intake manifold pressure) between the four operating points described above. Once the operating point is determined, the fuel consumption and other values available in the performance map lookup table are calculated using the same linear interpolation. Figure 11 shows the fuel consumption rate for the first 505 seconds of the FUDS for the engine operating on the rich combustor.

Cranking and Starting

Based on actual cold room engine starting tests at -30°C using gasoline, the combustor fuel flow rate was estimated as 0.25 gm/sec (methanol) during engine cranking at 100 RPM. This figure rises to 1.1 gm/sec at an idle speed of 800 RPM. These estimates were obtained from the design tool by calculating the output of the engine using the measured operating parameters (cranking speed, manifold temperature, pressure, and mass flow rate of air) from cold room testing using gasoline, then determining the operating parameters to provide that same output from the engine using the rich combustor fueled with methanol.

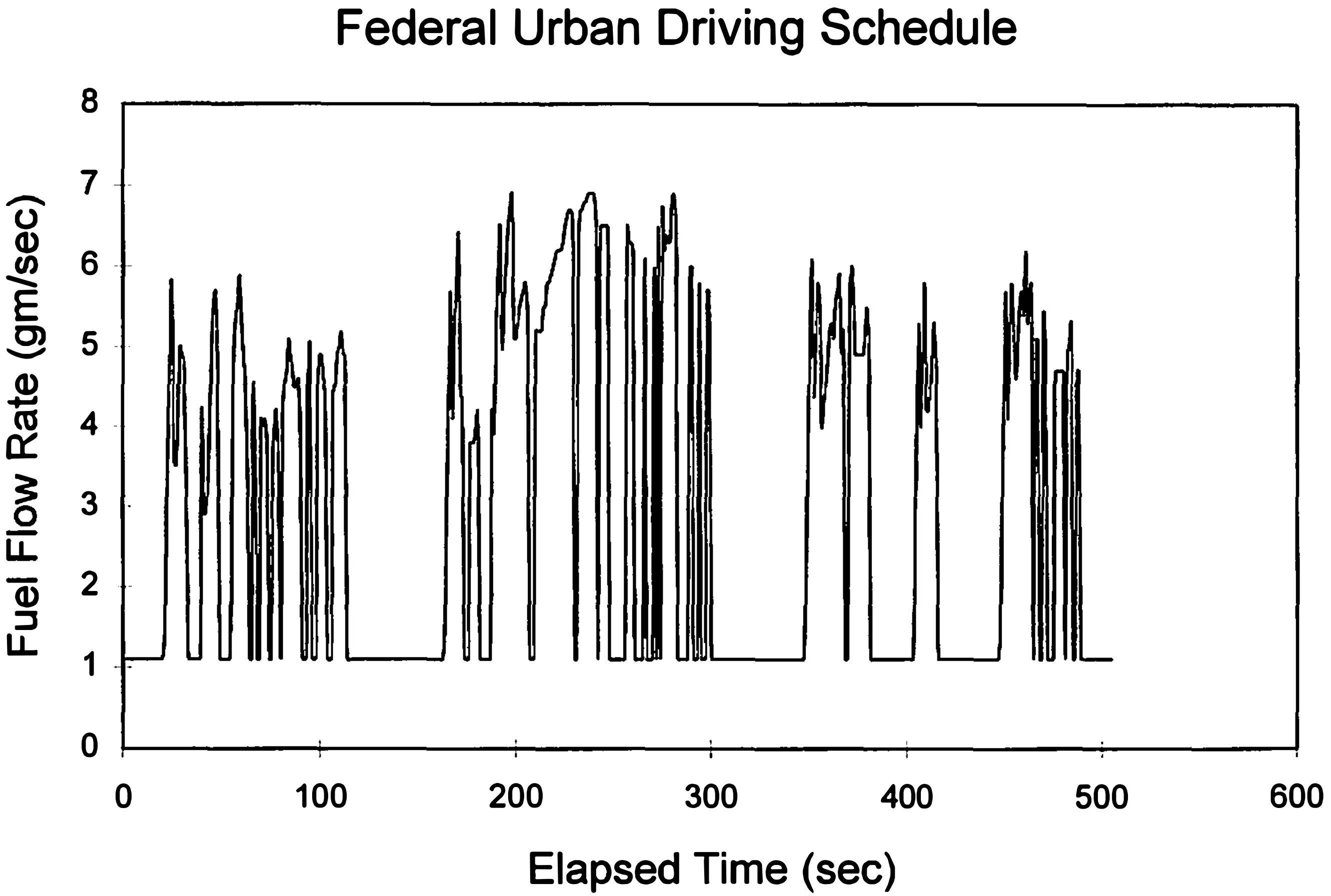


Figure 11 Calculated fuel consumption for test engine operating on rich combustor

Chapter III

Prototype Design and Fabrication

Approach

From a previous investigation of the rich combustor concept for cold starting (Hodgson 1994), the following areas of concern were noted.

1. The device should be downsized to be more suitable for under-hood installation.

2. Appropriate controls should be incorporated to make the system operation automatic.

3. More control over the air-fuel ratio is required.

These areas, together with the concerns listed in the above, were taken as the design criteria for the prototype rich combustor. The criteria will be addressed individually by describing their incorporation into the prototype design. Some features of the design address multiple concerns.

First Generation Combustor Design

A brief description of the prototype rich combustor and its operation will be beneficial. The prototype is fundamentally an "air-splitter." It is installed between the existing throttle body and the intake manifold on the engine, and divides the air flowing through the throttle body into two (physically three) streams. One stream flows through the combustion chamber where fuel is added and ignited. The fuel droplets burn in a rich diffusion flame to produce the desired products used to fuel the engine. This stream can be throttled to reduce the flow through it as the engine warms up and the requirement for the non-condensable gases is reduced. The remaining air flow is directed around the combustion chamber unaffected except for heat transfer from the combustion chamber and rejoins the combustor products prior to entering the intake manifold. A complete set of prints of the design drawings is included as Appendix B in this report.

Downsizing

Since this project involves the demonstration as well as investigation of the rich combustor cold start device, the ultimate goal is to install the device on a vehicle and demonstrate the ability to start and drive away under the cold ambient conditions required. The under-hood space available in a modern vehicle is very limited; therefore, the device must allow installation in the vehicle.

The prototype design was obviously not downsized to meet the requirement of vehicle installation. This prototype was not intended to be installed under-hood; instead, it was used in the dynamometer installation shown in Figure 12. Evaluation of this prototype on the dynamometer installation allowed the determination of the minimum size for the rich combustor to be installed on the vehicle.

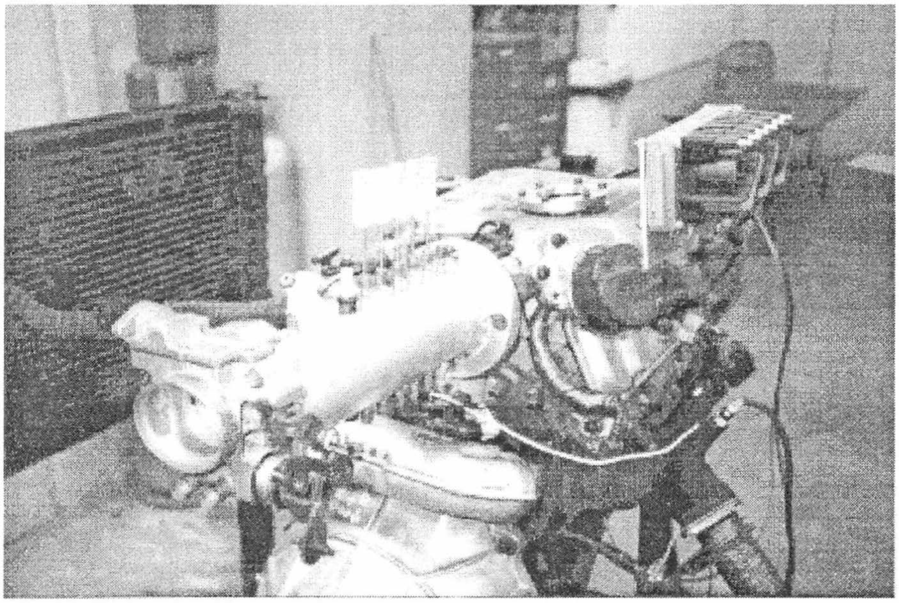


Figure 12 First generation rich combustor dynamometer installation

Control

The approach taken in the design of the prototype combustor was to minimize the amount of control required; the operation is to be as automatic as reasonable. The only control requirements anticipated were positioning of the throttle plate versus engine temperature, and regulating fuel flow to the combustor and the main fuel injection system.

The throttle plate can be positioned using temperature input to a positioning device such as a stepping motor. The goal is to position this throttle plate using some simple device. Control of the fuel flow will be managed by the engine control computer. The aftermarket engine controller used on the engine in the dynamometer installation allows programming of the enrichment schedule for fuel management. The design tool will predict the fuel flow requirements for operation of the rich combustor. Actual testing will determine the schedule of operation of the rich combustor as a function of engine warm-up, or the position of the throttle plate as described above. The fuel flow to the combustor's injector(s) must then be controlled to maintain the desired air/fuel ratio in the rich combustor, with the appropriate fuel directed to the main injector system to maintain the desired overall engine air/fuel ratio.

Air and Fuel Flow Regulation

Regulation of the air flow is the primary reason for choosing the "air-splitter" approach. The desired operating equivalence ratio for the rich combustor will be nominally 2.

This dictates that initially, when the engine is operating on the products of the rich combustor alone, half of the air flowing through the engine's throttle body will be used by the rich combustor, with the remaining half used to burn the combustor's products in the engine. The combustor is designed such that the flow areas of the combustion chamber is equal to the sum of the areas of the two air passages. Further, this flow area is equal to the flow area through the original throttle body, allowing undisturbed operation of the engine with the combustor's throttle plate completely closed.

Fuel flow is controlled by the engine control computer. The design tool has predicted the fuel flow requirements for operation of the rich combustor. Testing determines the schedule of operation of the combustor and therefore the air flow through it; the fuel flow to the combustor must then be controlled to maintain the desired air/fuel ratio in the rich combustor. The fuel flow to the main injection system must also be controlled in concert with operation of the rich combustor. The design tool can be used to predict this fuel flow requirement also, once the schedule of operation of the rich combustor is known.

Nozzle and Ignition System Selection

The fuel injectors and ignition system are basic to the operation of the rich combustor. Rather than designing the prototype for a particular injector and/or ignition system, provisions have been made to allow investigation and testing of available systems.

Material Type and Fabrication Method

Since the current prototype is known to not be the final configuration for installation on the vehicle, it was designed to minimize cost associated with material and construction. As shown on the included drawings, the prototype is primarily aluminum welded construction. The design considerations for the combustor to be installed in the vehicle will be primarily space limitations; therefore, the physical design of the device may be quite different. However, the operation of the device is not expected to change.

General

Ports are spaced along the axis of the prototype to allow installation of thermocouples and/or sampling probes to monitor the progression of the reaction. This will help to determine the minimum size required for the rich combustor. Access to the combustion chamber is provided to allow incorporation of flame-quenching devices. A port has been machined in the intake manifold plenum and fitted with a sight glass to allow visual inspection of the operation of the rich combustor.

Design Evolution

The first generation rich combustor design was modified as a result of the testing described below. The two areas which were modified were the ignition system and the fuel system.

The original design for the ignition system included conventional spark plugs. However, during fabrication it was decided to evaluate the use of small glow plugs as an ignition source. This type ignition has the advantage of requiring only a voltage source for the thin wires of the glow plugs; no other components such as coils, etc. are required. This configuration is shown in Figure 13. Testing of this configuration, as described below, showed that the glow plugs could not maintain sufficient temperature for ignition in the air stream as the engine was cranked. Raising the voltage supplied to the glow plugs to overcome this would result in immediate burnout when the air flow was disrupted.

The glow plug ignition system was replaced by conventional spark plugs. The spark plugs were mounted in the combustor as shown in Figure 14.

A coil assembly from a Saturn automobile was used to fire the spark plugs; a single coil fired both spark plugs in a manner similar to the ignition system used on the engine. A device which operates independently of the engine was fabricated to provide the required excitation signal to the coil assembly. The circuit diagram for the device is included in Appendix D. This made the combustor ignition system completely self-contained.

This configuration suffered from two problems. The spark plug gap was near the wall and was not exposed to very much of the fuel spray from the injectors. The other problem was a result of the vertical mounting and the conductivity of methanol. The

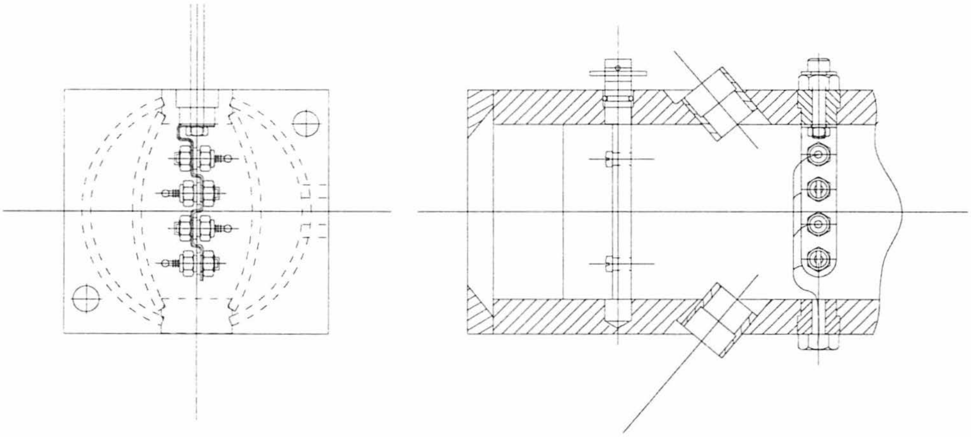


Figure 13 Glow plug configuration

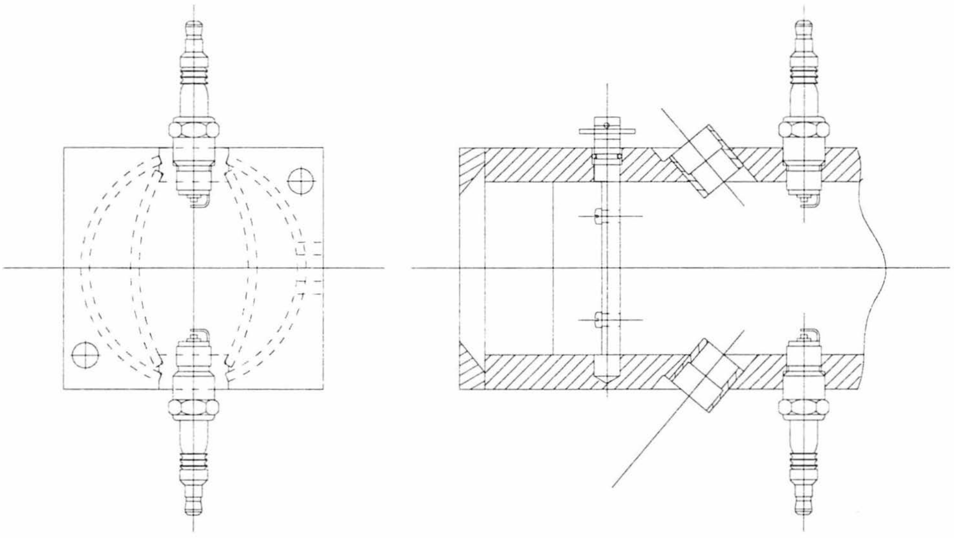


Figure 14 Spark plug configuration

annulus between the insulator and body of the lower plug filled up with methanol and essentially grounded the electrode, preventing that spark plug from firing.

This conventional spark plug configuration was replaced by igniters used in oil furnaces. This igniter has an electrode that extends well beyond the insulator and does not have a ground electrode since these igniters are intended to arc to a ground that is part of the furnace. For this application, however, two igniters were used with the modified Saturn ignition system and were installed such that the arc was from one igniter to the other (see Figure 15). The gap between the electrodes of the two igniters was approximately one quarter inch although the ignition system would allow a gap of greater than one inch. This configuration proved satisfactory and no problems were encountered during testing at temperatures as low as -30°C. This ignition system was carried over to the second generation combustor design.

The original design for the fuel system included conventional automotive injectors for delivering the fuel. Methanol compatible injectors with a rated flow rate of 3.9 gm/sec were obtained from Rochester Products. Calculations from the model indicated that two of these injectors would be adequate to deliver the fuel at a rate for drive-away. Initial testing of the injectors was conducted with the conventional spark plug configuration with poor results; the fuel was not reliably ignited. Subsequent off-line testing of the injectors revealed that the injectors had an almost jet-like spray pattern. From previous work (Hodgson 1993 and Iwai 1984), it was known that a fine spray would be required to ignite the methanol. The injectors could, however, provide the

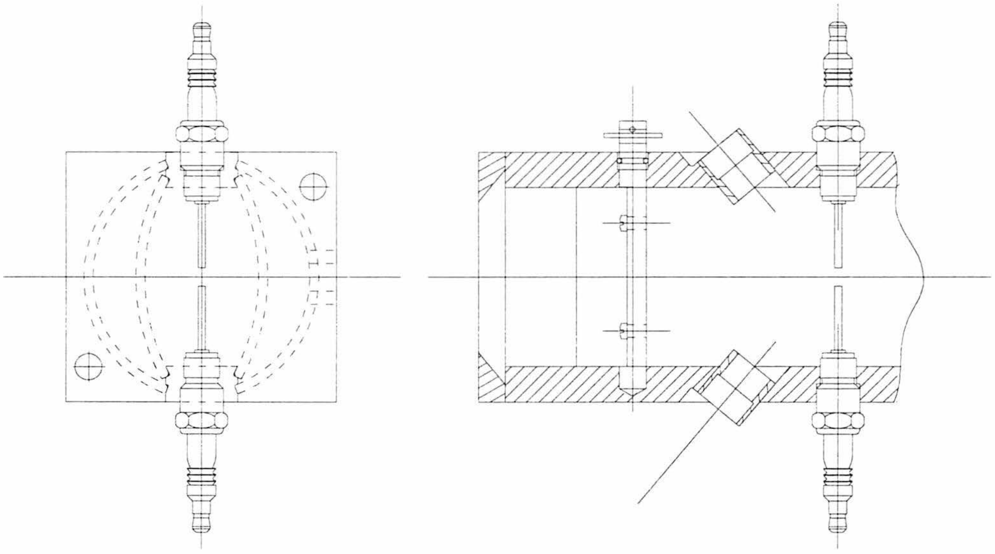


Figure 15 Furnace igniter configuration

control of the fuel flow as required. Nozzles such as those used in oil furnaces were chosen to provide the required spray pattern. As shown in Figure 16, the resultant design was an assembly of the injectors for flow control in line with the nozzles for providing the required spray pattern. This configuration proved satisfactory and was carried over to the second generation combustor design.

Several nozzles were flow tested to determine the flow rate and spray pattern. It was decided to use one nozzle with a measured flow rate of 0.97 gm/sec without an injector which meant that it delivered that flow rate at all times. That flow rate corresponded to the idle fuel flow rate requirements predicted by the model. The injector - nozzle assembly was used in the other position with a nozzle with a measured flow rate of 1.57 gm/sec for initial cold start testing.

Other modifications to the original design were incorporated as a result of, and during, the testing of the first generation combustor. A baffle arrangement was added to disrupt the flow through the combustor and provide more surface area for quenching the flame. In addition, a quenching nozzle was added to both quench the flame and cool the products. The final configuration of the first generation combustor design is shown in Figure 17. Figure 18 shows the final configuration installed on the test engine.

Second Generation Combustor Design

Two major problems with the first generation combustor design surfaced during testing. The combustion zone exceeded the length of the combustor and the flame continued

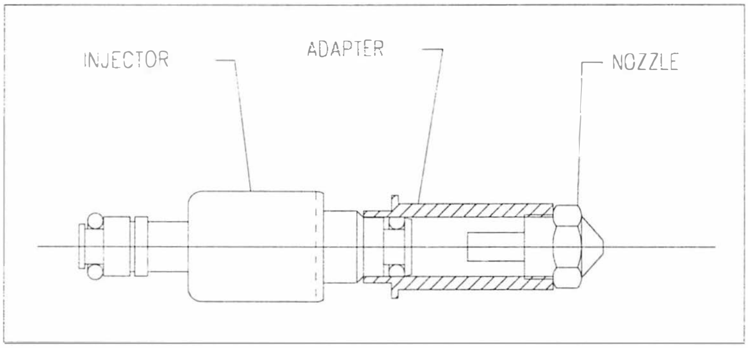


Figure 16 Injector - nozzle assembly

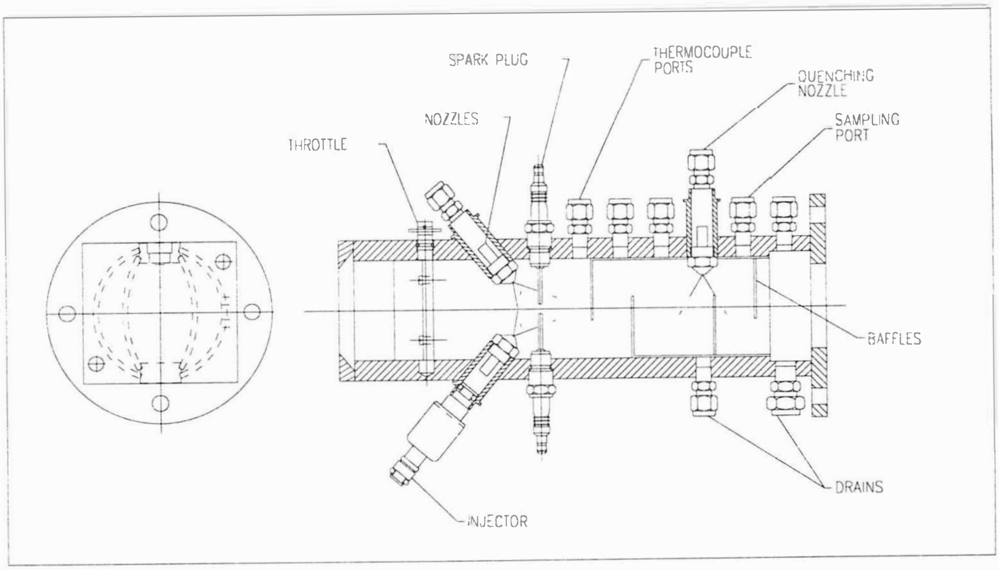


Figure 17 First generation design final configuration

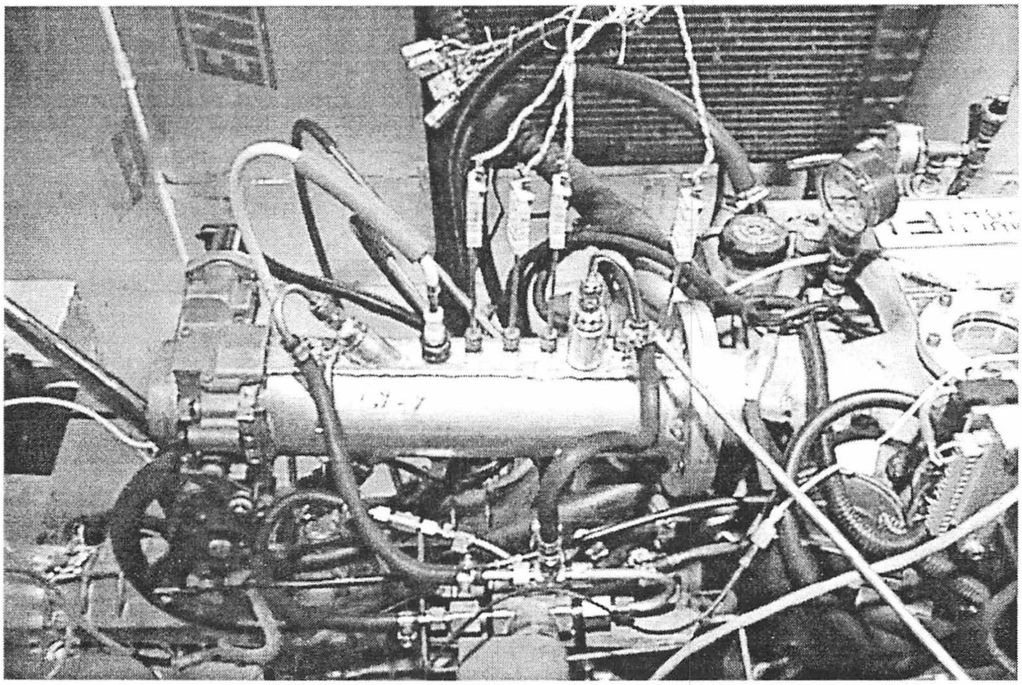


Figure 18 First generation final configuration installed on test engine

past the point where the fresh air rejoined the combustor products. In fact, the flame continued on into the intake manifold which resulted in all the combustible gases being consumed before they reached the engine cylinder. The second problem was associated with the splitting of the air. Due to the reaction occurring in the combustor area, splitting the air flow through the combustor and fresh air areas could not be accomplished by simply using equal flow areas.

The second generation design addressed both of these problems while maintaining the original overall concept for operation of the rich combustor as described above. The ignition system developed for the first generation design was included in the second generation design. The injector/nozzle hybrid system for fuel delivery was also used in the second generation combustor design. In addition, the second generation design addressed under-hood installation in a vehicle. A complete set of prints of the design drawings is included as Appendix C in this report.

Design Features

The combustion zone length problem associated with the first generation design could not be overcome by the baffle arrangement as described above; therefore, the only solution was to increase the length of the combustion chamber in the second generation design. Keeping in mind the vehicle integration issue of limited under-hood space, a flat spiral type configuration that would be installed on top of the existing plenum emerged as the most promising configuration. However, manufacturing a design of this type would be very expensive; fabrication would require either extensive machining or very difficult forming of tubing. A compromise design was chosen which maintained the general configuration described above, but was considerably simpler to manufacture. Rather than a flat spiral, a square cornered maze-type configuration was used which simplified the fabrication considerably. The original concept drawing for this design is shown in Figure 19. This configuration could be fabricated as a weldment from rectangular steel mechanical tubing readily available. This configuration increases the length of the combustor flow path by a factor of approximately 10; this has proven adequate for the testing performed so far.

The second generation combustor is installed on top of the plenum with the combustor outlet mated to the added port in the plenum previously used for a sight glass. The overall height of the second generation combustor is 3 inches above the top of the plenum, which allows for under-hood installation.

The problem associated with air splitting was originally addressed by using larger tubing for the combustion chamber than for the fresh air. The flow area of the combustor was approximately 6 square inches while the fresh air side was approximately 4 square inches. This proved to be an improvement to the first generation design but did not completely solve the problem. Testing showed that too much air was flowing through the fresh air side resulting in a very lean mixture supplied to the engine. Modifications to the design for proper air splitting are described below.

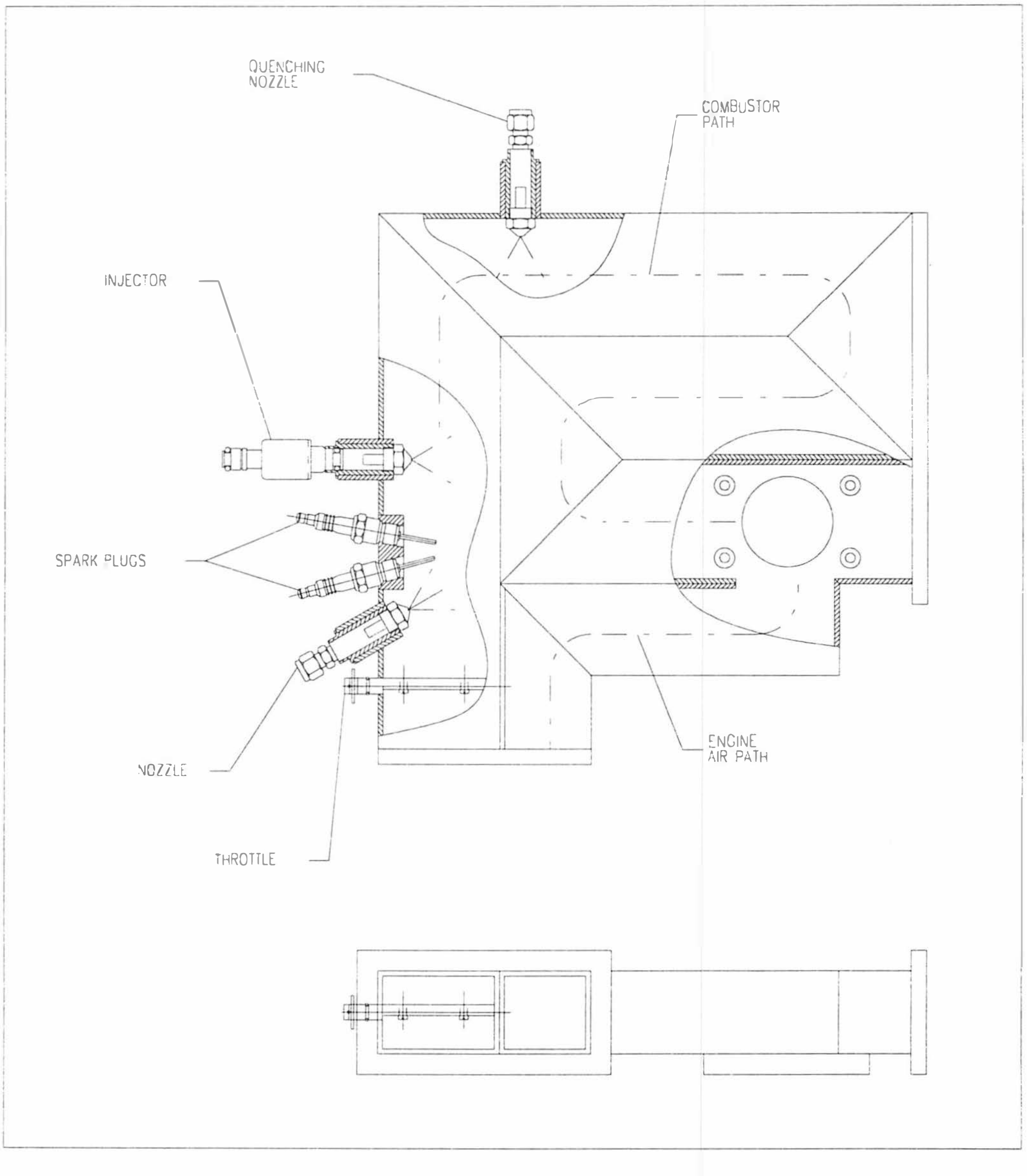


Figure 19 Second generation design concept drawing

General

With the configuration chosen for the second generation design, the combustor could not be fit between the main throttle body and the intake manifold as with the first generation design. To maintain the overall concept for control, the throttle body must be mounted at the inlet of the combustor. This required that the opening in the plenum be capped where the throttle body originally was installed.

The fuel delivery system was designed to deliver the fuel flow rate required for drive-away as predicted by the model; this would require two injectors (in addition to the one serving for idle). The configuration shown in Figure 20 was chosen. The idle injector is upstream of the ignition system where the flame is established. The two additional injectors are installed immediately downstream of the ignition system and deliver fuel, under the control of the engine controller, to the established flame. Provisions for a quenching nozzle are also provided.

Design Evolution

As described above, the air flow through the fresh air side was too high. An additional throttle was installed in the fresh air side, as shown in Figure 21, to further restrict the flow. Testing showed that the mixture provided to the engine was closer to stoichiometric.

The original design featured drains at both ends of the second leg of the combustion chamber. This configuration was adequate for removing any liquid that collected and

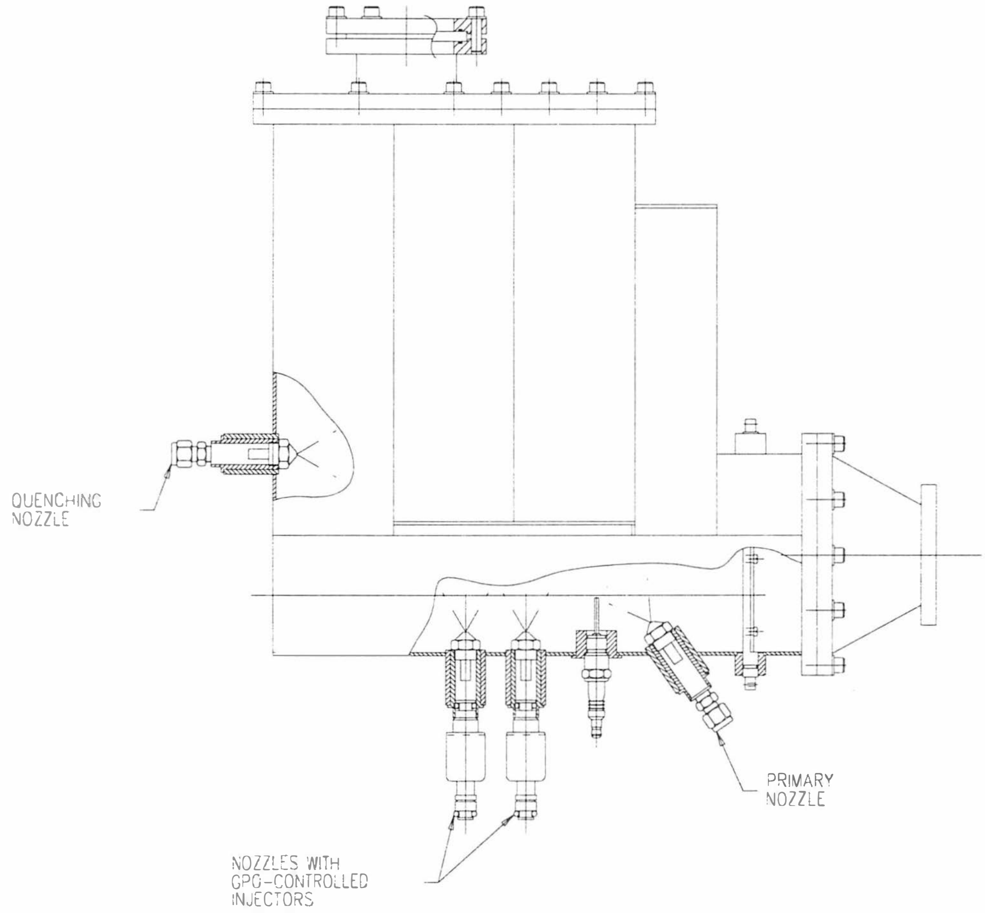


Figure 20 Fuel delivery configuration

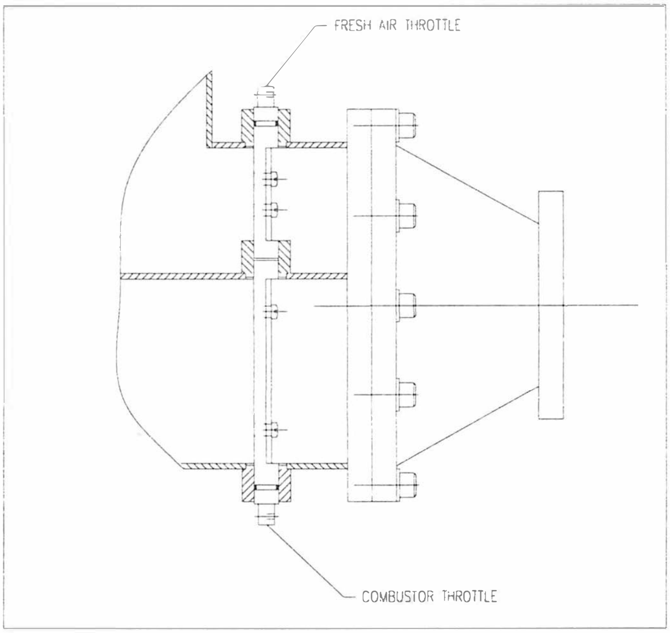


Figure 21 Fresh air throttle installation

flowed on the floor of the combustor. However, liquid droplets were observed through the sight glass being carried in the gas stream. To prevent these droplets from being carried to the engine, a fine mesh stainless steel wire cloth was installed immediately upstream of the second drain to knock the droplets down to the floor where they would flow out the drain. This screen was mounted to a frame that served also as a sort of dam to any liquid flowing along the floor. Testing showed that this screen heated up and glowed red which tended to cause whatever fuel was being carried by the gas stream to burn or evaporate when it reached the screen.

The current configuration of the second generation combustor is shown installed on the test engine in Figures 22 and 23.

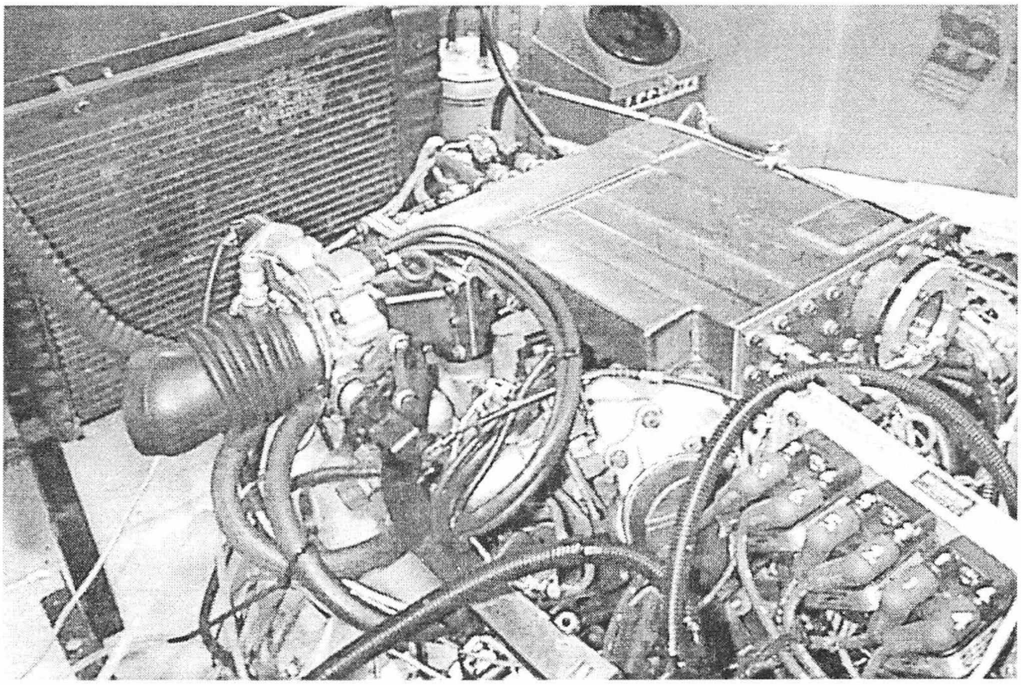


Figure 22 Second generation combustor installed on test engine

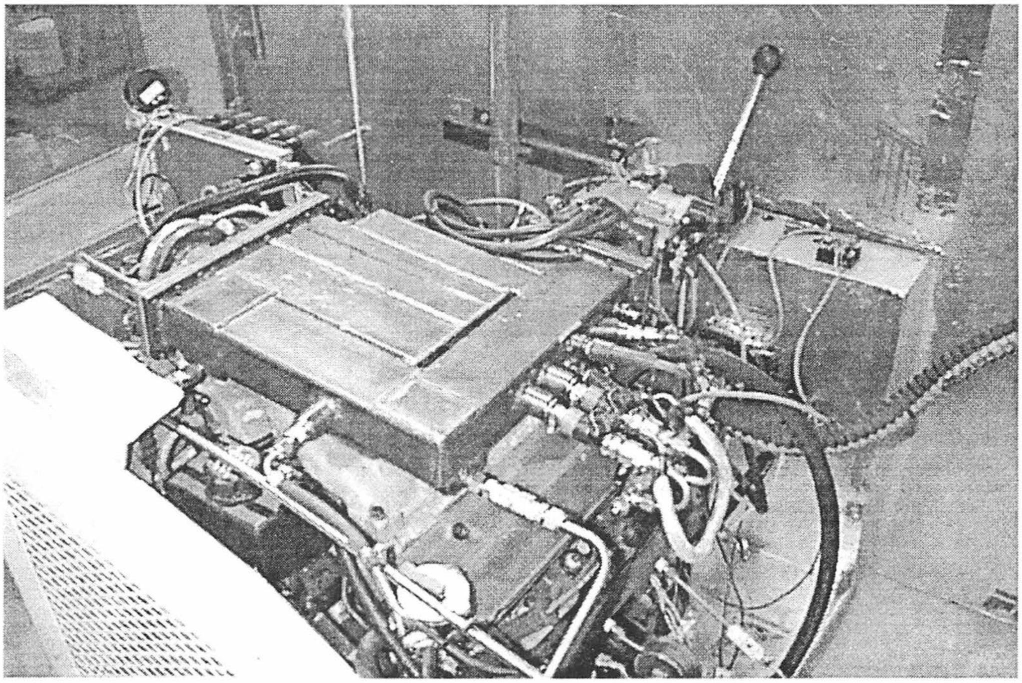


Figure 23 Second generation combustor on test engine

Chapter IV

Prototype Testing and Evaluation

Approach

Because of the availability of cold temperature testing capabilities, including an engine dynamometer, and an engine identical to the engine in the test vehicle, the task of vehicle integration could logically be combined, to a great extent, with the testing and development of the prototype. We could use the dynamometer driving the engine as a pump to test the combustor while the issues of fit, interfaces, and control were concurrently addressed. With this capability, many of the problems associated with vehicle integration were addressed early in the development of the combustor. This parallel approach was taken rather than the original task breakdown which included bench testing.

Testing Configuration

The testing facility consists of a refrigerated enclosure built on one end of the dynamometer mounting bed plate. The walls and floor are frame construction with approximately twelve inches of Styrofoam insulation. The refrigeration unit is capable of cooling the chamber, including a hot engine, to -30°C in approximately nine hours. Access to the dynamometer is through a driveshaft tunnel in the rear wall.

As mentioned above, the engine in the test cell is identical to the engine in the test vehicle, a 2.8 L V-6 Chevrolet. The engine is port-injected and comes originally with a distributor-less ignition system. The original engine controller has been replaced with an aftermarket Electromotive TEC-II engine controller for controlling the ignition, fuel injection, and fuel management for the rich combustor. Monitoring access to the data inputs to the engine controller is provided by a PC-based system, which also serves as the programming and file transfer software.

The engine in the test vehicle is mounted transverse for front wheel drive, but a transmission is used in the test cell to couple the engine to the dynamometer. Figure 24 shows the test engine, with the first generation combustor installed, in the test cell. The engine sensors outputs as well as thermocouples installed in the combustor and the engine are monitored and recorded using the data acquisition system. Complete descriptions of the test equipment are included in Appendix D.

First Generation Combustor Testing

Testing of the first generation combustor began with evaluation of the candidate ignition systems. A reliable ignition system is an absolute requirement for success of the combustor.

Ignition System Testing

Testing of the ignition system on the first generation combustor design is described in Chapter II. The basic configuration was carried over to the second generation design.

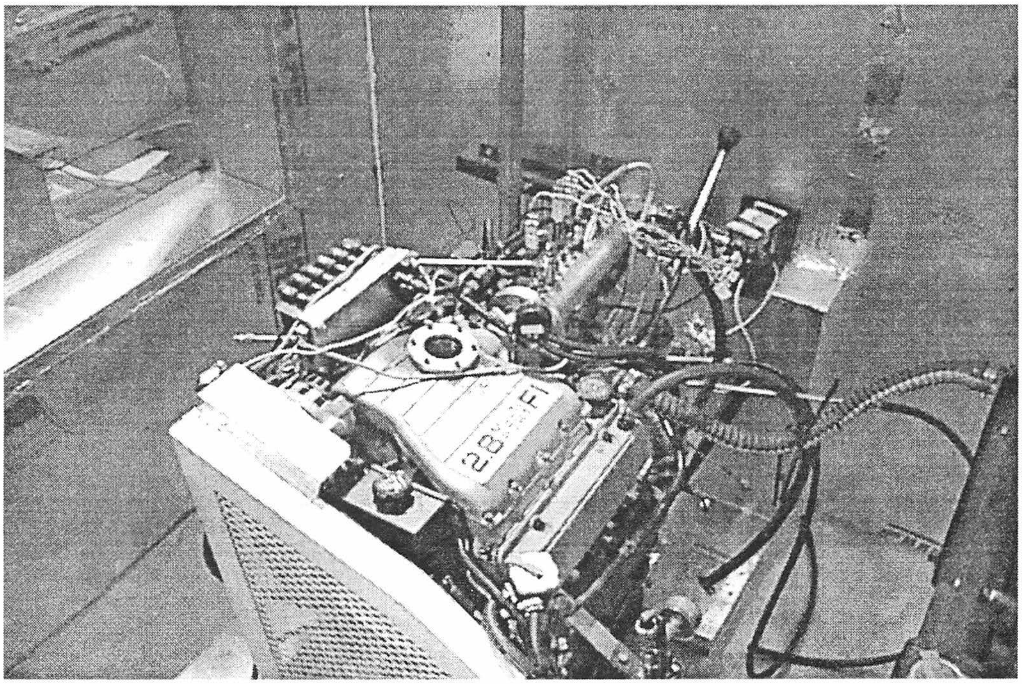


Figure 24 First generation combustor installed on test engine in test cell

Fuel System Testing

As stated above, the original design used automotive fuel injectors obtained from Rochester Products which were methanol compatible. The spray from these injectors was observed to be jet-like, unsuitable for this application. The nozzles were available from previous cold start research (Hodgson 1993) which were known to provide an ignitable spray of liquid methanol. These nozzles would be used to provide the desired methanol spray pattern; the injectors could however be used to control the flow through the nozzles.

The General Purpose Output (GPO) of the Electromotive controller would be used to adjust the methanol flow rate. The GPO is a 12 volt square wave output with pulse width modulation operating at 31 Hz. The percent-on time can be programmed in the controller software as a function of engine speed and intake manifold pressure. The GPO can drive any device (solenoid, etc.) with an impedance of greater than 45 Ohms. The injectors used are the saturation type and have an impedance considerably less than this requirement, 2.8 Ohms. A relay would be required to drive the injectors using the signal from the GPO as input. A solid state relay was chosen; details are included in Appendix D.

Before implementing the injector-nozzle configuration shown in Figure 16, the injectors were tested to determine the effect of the pulse width modulation on the flow rate. A bench test rig was assembled using a fuel pump identical to the pump used on the test engine feeding an injector being controlled by the GPO directed into a graduated cylinder. The results are shown in Figure 25. As Figure 25 shows, the flow rate through the injector is linear with percent GPO. This characteristic simplifies the task of calibrating the fuel flow rate.

In addition to the two nozzles used on the previous project, several other nozzles were obtained to have a range of flow rates. These nozzles are standard hollow cone spray nozzles commonly used in industrial applications for rinsing, cooling, etc. They are typically equipped with a sintered bronze filter element integral with the filter body. The nozzles are rated in gallons per hour at 100 psi and stamped with that number. Several of the nozzles were tested on the bench test rig described above, without the GPO-controlled injector, with the results shown in Table 1. Table 1 shows some variation in the measured flow rate versus "advertised" flow rate. Therefore the nozzles used in the combustor must be flow tested before use.

The nozzle identified 1.25 with a measured flow rate of 0.97 gm/sec was chosen to be used in the upper position in the combustor. This flow rate is close to the flow rate for idling as predicted by the model. The strategy for fuel delivery is then to use that nozzle, without an injector, supplying methanol to the combustor at the rate required for idling. The nozzle identified as 2.00 with a measured flow rate of 1.13 gm/sec was used in the lower position with an injector controlling the flow rate using GPO control. Other nozzles would be tested in the lower position. As a result of operational testing described below, a third nozzle was added downstream for quenching as shown in Figure 17.

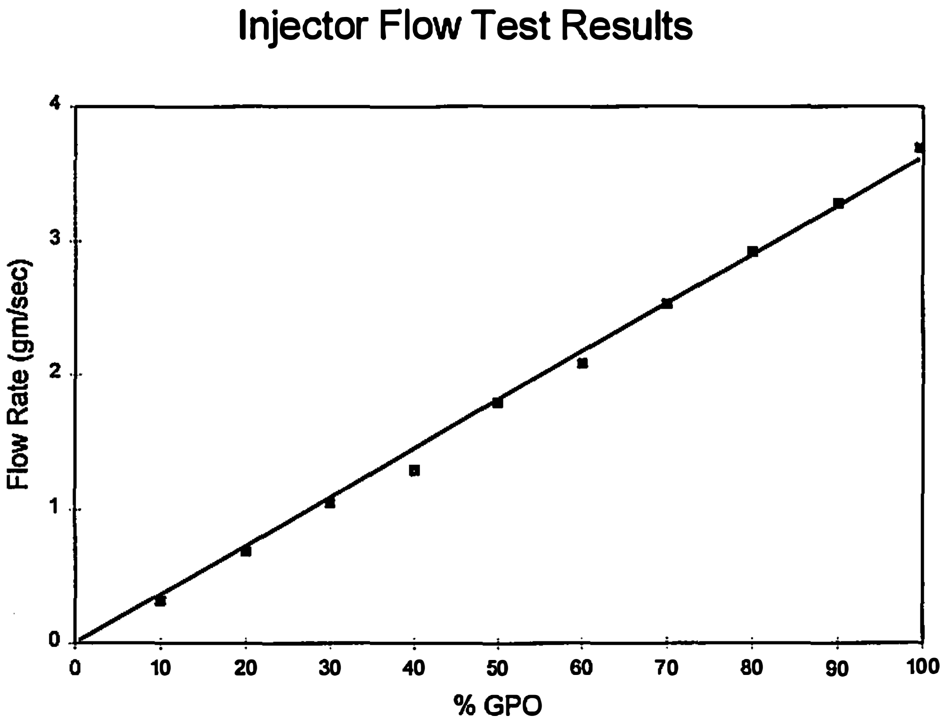


Figure 25 Injector flow test results

Table 1 Nozzle Flow Test Results

Nozzle ID	Mass Flow Rate (gm/sec)
.75	0.42
.85	0.44
1.00	0.50
1.25	0.97
1.35	1.59
1.50	0.74
1.75	0.98
2.00	1.13
2.00	0.39 *nozzle from previous work

Operational Testing

From the testing of the first generation design combustor, the following conclusions were reached:

- The rich combustor concept for cold starting is viable.
- The combustion zone length provided was too short for the reaction to reach completion before rejoining the fresh air stream.
- The flow restriction associated with the baffles caused the flame to travel upstream and curl around the wall separating the combustor passage from the fresh air passage.
- The quenching nozzle is required to cool the combustor products to prevent the combustibles from being consumed in the intake manifold.
- The series throttle arrangement of the original design must be maintained for simplicity of control.
- The first generation combustor was too large for underhood application

These conclusions drove the design for the second generation rich combustor design. The second generation design significantly increased the combustion zone length, thus eliminating the need for baffles and the resulting flow restriction. The physical layout of the combustor was completely changed to accommodate underhood installation . The series throttle arrangement, as well as the fuel and ignition systems developed thus far, were carried over to the second generation design.

Second Generation Combustor Testing

The initial configuration of the fuel system for the second generation combustor was the same as the final configuration for the first generation combustor. The nozzle identified 1.25 with a measured flow rate of 0.97 gm/sec was installed in the upper position in the combustor. The nozzle identified 2.00 with a measured flow rate of 1.13 gm/sec was used in the lower position with an injector controlling the flow rate using 30% GPO control. The nozzle identified 0.75 with a measured flow rate of 0.42 gm/sec was used for the quenching nozzle. An additional nozzle/injector assemble was installed in the third port, but the injector was not connected to the controller. The extended tip spark plugs were again used; however, the transformer was replaced by the Saturn ignition system driven by the in-house developed device.

The first start attempt after installation, at room temperature, was successful; the engine started using the starter. A series of cold start attempts were performed, first at -30°C and then at -8°C. Although the engine fired sporadically during several of the attempts, there were no successful starts. Data from one of the start attempts is shown in Figure 26. After a few unsuccessful attempts, the spark plugs in the engine would become very wet with methanol and water and then would not fire. Figure 26 shows this plainly; note the data shows no evidence of firing after the second attempt at 0% GPO. The cold start testing was suspended in favor of more systematic testing to determine the operating parameters of the combustor required to maintain the correct stoichiometry in the combustor and the engine.

Stoichiometry Testing

Determining the operating equivalence ratio of the combustor was accomplished by motoring the engine with the dynamometer with the engine's ignition system disabled such that the engine is serving as an air pump for the combustor. The gas analyzer is used to determine the CO and CO_2 content of the combustor product stream by sampling from the port near the sight glass as shown in Figure 27. The downstream sampling port was used to determine if the product composition changed as it flowed through the third "leg" of the combustor which would indicate that the reaction was continuing.

The output of the gas analyzer is on a dry basis since the water vapor condenses from the sample stream. The measured value of CO concentration was compared to the equilibrium composition calculations, shown in Figure 3, to estimate the operating equivalence ratio for the *combustor*.

The operating equivalence ratio of the *engine* was determined using the gas analyzer sampling from a port in the exhaust pipe. For the range of combustor equivalence ratios considered, the concentrations of hydrogen and carbon monoxide are approximately equal. For the stoichiometric combustion of an equimolar mixture of hydrogen and carbon monoxide, the molar amounts of the CO, H_2, and 0_2 should be equal.

$$CO + H_2 + O_2 \rightarrow CO_2 + H_2O$$

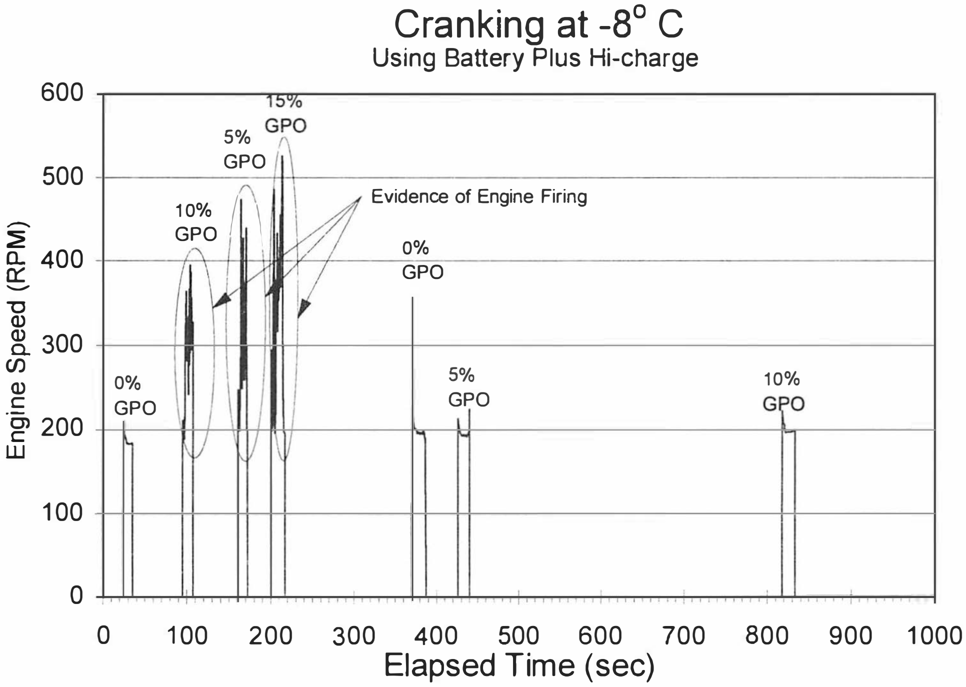


Figure 26 Unsuccessful cold start attempt

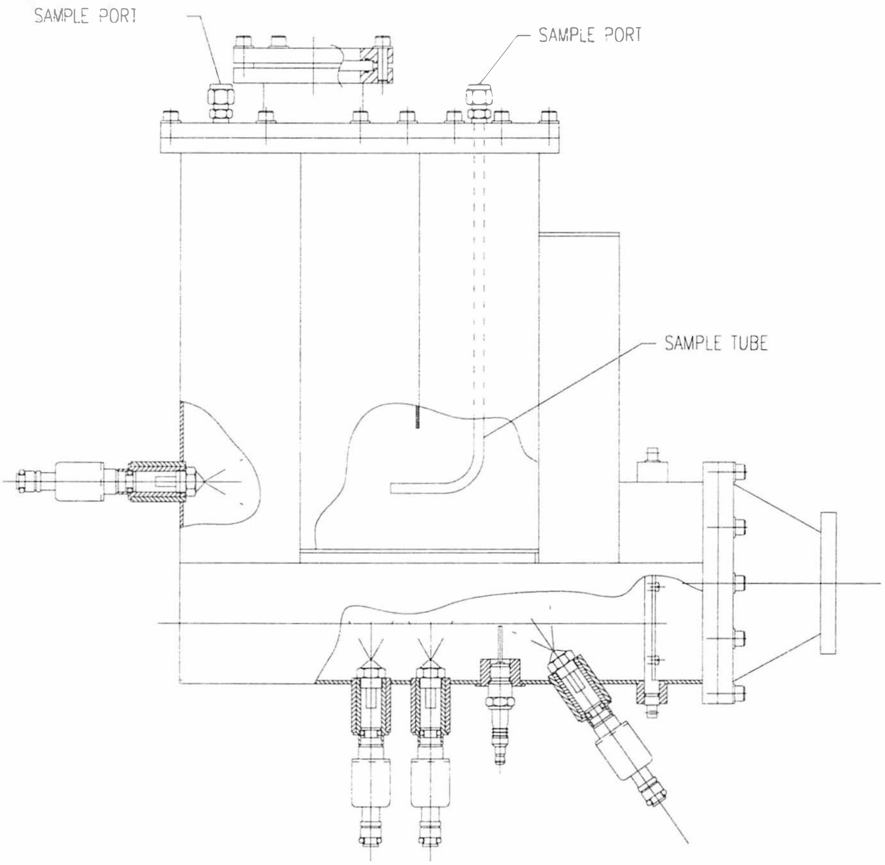


Figure 27 Sampling port locations

The concentrations of the CO and O_2 were compared to estimate the operating equivalence ratio for the engine. This estimate is then used to determine the balance of the air flow through the two sides of the combustor. A lean condition, indicated by more O_2 than CO, indicates that too much air is flowing through the engine fresh air side of the combustor; a rich mixture indicates too much flow through the combustor side.

Testing began with the fuel system unchanged from the initial cold start testing- the nozzle identified as 1.25 with a measured flow rate of 0.97 gm/sec was installed in the position upstream of the spark plugs (position #1). The nozzle identified as 2.00 with a measured flow rate of 1.13 gm/sec was used in the first position downstream of the spark plugs (position #2) with an injector controlling the flow rate. The nozzle identified as 1.50 with a measured flow rate of 0.74 gm/sec was installed in the second position downstream of the spark plugs (position #3) with an injector controlling the flow rate. The port for the quenching nozzle was plugged.

The test cell was cooled to approximately -9°C, and the temperature was monitored to assure that the engine had reached a steady state temperature. The engine was then motored at various speeds while the % GPO was varied. The gas analyzer, sampling from the downstream sampling port initially, was used to determine the concentrations of CO, CO_2, and O_2 . Table 2 shows the results of testing with only the first downstream nozzle operating.

The sampling location was moved to the upstream port to compare the results from the two locations. The results for the upstream location are shown in Table 3.

The results indicate that mixing in the area of the port into the intake manifold , where the fresh air joins the combustor product stream, may be causing oxygen from the fresh air stream to be introduced into the sample. The upstream location was used for all subsequent combustor product sampling.

As shown in Table 2, the CO content of the combustor product stream is fairly insensitive to the percent GPO, above 5% GPO. The nozzle in the first downstream position was changed to investigate whether this phenomenon was related to that particular nozzle and/or the effect of small % GPO values on the spray pattern or flow rates from the nozzle. The nozzle identified as 1.00 with a measured flow rate of 0.50 gm/sec was assembled with an injector and installed in the first downstream position (position #2) , with the other two nozzles unchanged The procedure described above was repeated with the results shown in Table 4.

Based on the results of this test, a GPO table was constructed to be downloaded to the Electromotive controller. After adjusting the set point values for the engine speed, the percent GPO was specified for each RPM set point. The GPO table also allows setting the percent GPO for manifold pressure set points at each engine speed set point; however, a single value of percent GPO was used for all values of manifold pressure at each engine speed set point. This should be sufficient for starting and idling, but the

Table 2 Combustor Product Composition- Downstream Port

Engine Speed (RPM)	% GPO	% CO	% CO_2	% O_2
200	0	9.9	7.2	2.3
200	5	10.2	7.0	2.1
200	10	9.9	6.9	1.9
200	15	9.6	6.9	2.1
300	0	9.9	6.8	1.8
300	5	10.5	6.7	3.2
300	10	10.5	6.7	3.4
300	15	10.1	6.8	2.8
400	0	9.7	7.0	2.9
400	5	10.3	6.8	2.7
400	10	10.4	6.8	1.6
400	15	10.5	6.7	3.3
400	20	10.3	6.8	1.8
500	0	9.1	7.4	2.3
500	5	10.2	6.9	2.7
500	10	10.4	6.8	1.9
500	15	10.8	6.7	1.7

Table 3 Combustor Product Composition- Upstream Port

Engine Speed (RPM)	% GPO	% CO	% CO_2	% O_2
500	0	8.4	8.8	0.9
500	5	10.9	8.87.6	0.5
500	10	10.7	7.5	1.0
500	30	11.0	7.2	1.4

Table 4 Combustor Product Composition by Nozzle

Engine Speed (RPM)	Active Nozzle Number	% GPO	% CO	% CO_2	% O_2
200	3	0	10.4	7.7	1.7
200	3	5	10.5	7.5	1.4
200	3	10	10.3	7.3	1.5
200	3	15	10.0	7.0	2.6
200	3	20	9.9	6.9	2.3
200	3	5	10.5	6.8	2.7
200	2	0	10.8	7.0	1.9
200	2	5	10.6	7.1	-
200	2	10	10.2	7.3	1.3
200	2	15	10.4	-	-
200	2	20	9.3	6.7	4.6
500	2	0	5.3	9.9	8.0
500	2	5	6.5	9.2	8.8
500	2	10	7.0	-	-
500	2	15	10.9	7.2	2.8
500	2	20	10.9	6.9	6.1
500	3	0	5.1	10.4	-
500	3	5	6.7	9.8	1.8
500	3	10	6.5	10.1	2.6
500	3	15	10.2	7.9	1.6
500	3	20	10.2	7.7	1.5

Table 4 (continued)

Engine Speed (RPM)	Active Nozzle Number	% GPO	% CO	% CO_2	% O_2
500	3	30	10.1	7.7	1.8
500	2 & 3	0	3.8	11.1	7.9
500	2 & 3	5	4.6	10.5	10.0
500	2 & 3	10	9.5	8.2	1.1
500	2 & 3	15	10.6	7.3	1.2
500	2 & 3	20	10.7	7.2	1.4
200	2 & 3	0	10.3	7.2	2.4
200	2 & 3	5	10.4	7.2	1.1
200	2 & 3	10	9.6	7.1	3.0

table will need to be modified later to include manifold pressure dependence for off-idle operation. Table 5 lists the GPO settings.

With both injectors downstream of the spark plugs (positions 2 & 3) being controlled per this GPO table, the engine was cranked using the starter. Samples were analyzed from the combustor as well as the exhaust pipe. The combustor product stream composition was 11.4% CO, 7.7% CO_2, and 0.8% O_2. The composition of the sample taken from the exhaust pipe was 4.5% CO, 3.8% CO_2, and 12.4 O_2 . This indicates a very lean mixture being fed to the engine resulting from too much air flowing through the fresh air side of the combustor. A thin rectangular shim was installed between the flanges of the main combustor body and the adapter for the engine throttle body. This shim could be positioned to restrict the flow through the fresh air side by various amounts depending on the position of the shim. The shim was initially positioned such that there remained approximately 1 in^2 of flow area for the fresh air side compared to 4 in^2 for the unrestricted case.

The tests described above were repeated at various shim location settings with the engine motored by the dynamometer. The engine operating equivalence ratio, as indicated by the composition of the sample taken from the exhaust pipe with the engine not firing, could be affected by adjusting the position of the shim. Cold start attempts resulted in sporadic firing of the engine, but no true start and idling. After a few unsuccessful attempts, the engine spark plugs became wet and would not fire. To

Table 5 GPO Settings

Engine Speed (RPM)	% GPO
200	5
500	15
1000	30
1500	45

address the spark plug wetting problem, a device was fabricated and installed just upstream of the second drain consisting of a double thickness of fine stainless steel wire cloth mounted to an aluminum block. The wire cloth was used to knock down any unburned fuel droplets being carried by the product stream, and the aluminum block served as a dam to prevent the liquid from flowing along the floor of the combustor and direct it into the drain.

Testing as described above revealed that the double thickness of wire cloth restricted the flow through the combustor side too much, resulting in a lean mixture measured at the exhaust pipe. One thickness of the wire cloth was removed and the wire cloth/dam assembly reinstalled. With the shim restricting the flow through the fresh air side as much as possible, the composition of the combustor product stream was measured as 12.7% CO, 8.0% CO_2, and 0.8% O_2. The composition of the sample taken from the exhaust pipe was 6.1% CO, 4.1% CO_2, and 11.2 O_2 . This composition indicates a lean mixture being fed to the engine, but no further adjustment to the shim was possible. The divider between the two sides of the combustor does not extend to the flange face, and thus the shim, which means that the flow area could only be reduced to approximately 0.75 in^2 . Since hydrogen has a wide flammability range, the mixture being fed to the engine may be combustible even though lean.

Fresh dry spark plugs were installed in the engine, the ignition system of the engine was enabled, and a cold start attempt was made at -25°C. The engine started and idled on the third attempt as well as the fourth attempt as shown in Figures 28 and 29. The

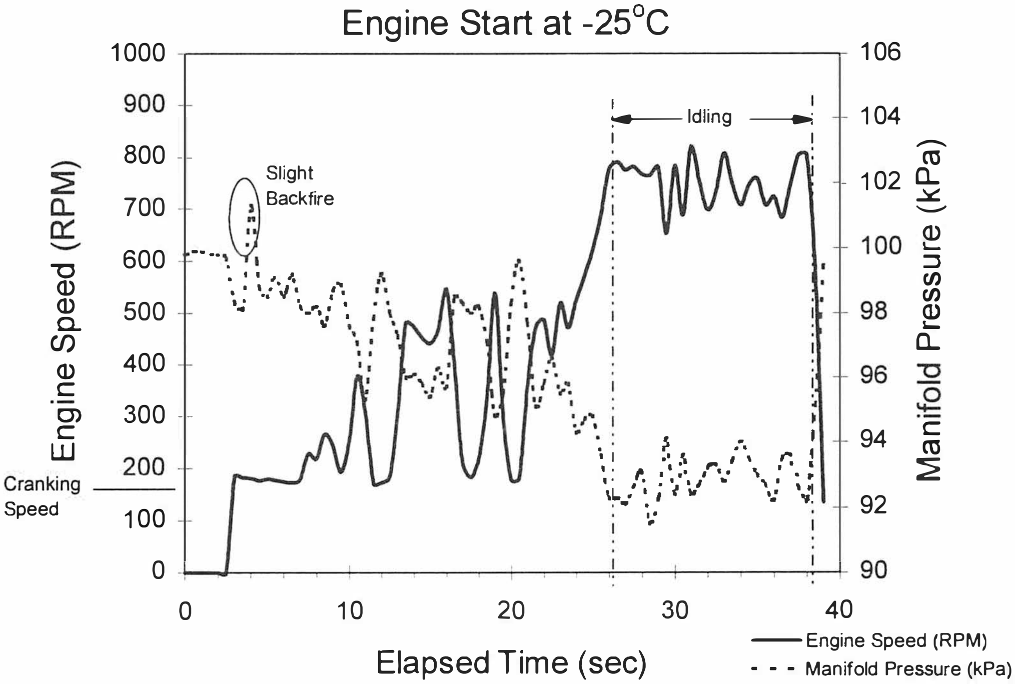


Figure 28 Successful cold start- third attempt

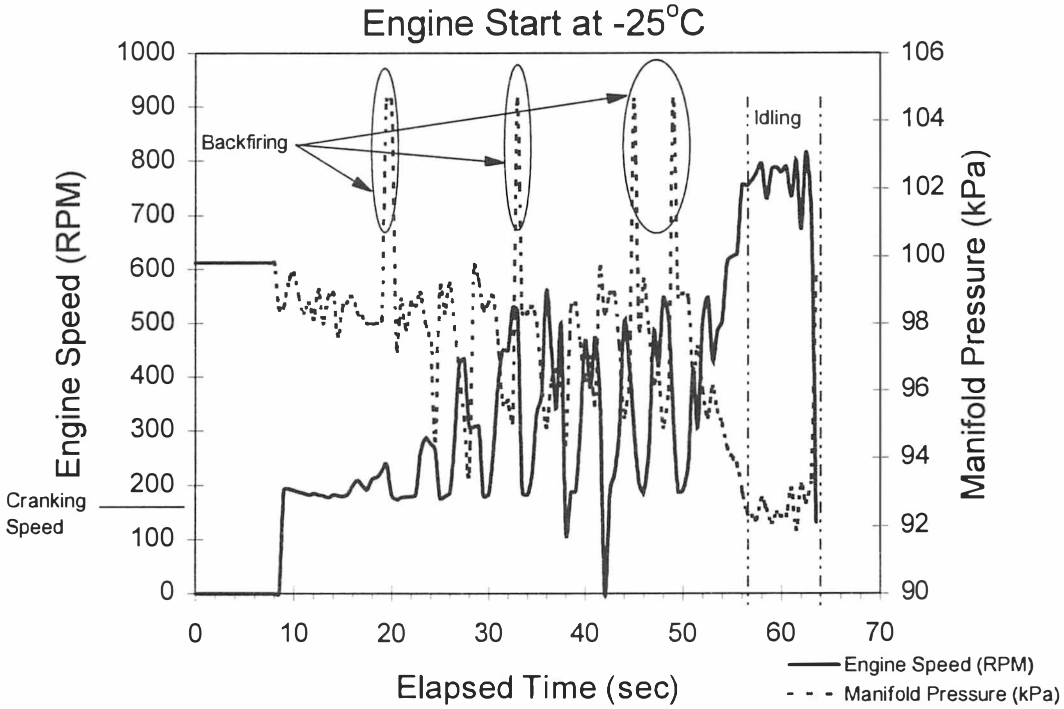


Figure 29 Successful cold start- fourth attempt

backfiring indicated is not fully understood, but is believed to be associated with the high flame speed of hydrogen combustion. As a result of this testing, the combustor was modified to add a throttle plate, as shown in Figure 21, to replace the temporary shim. The added throttle plate would allow restricting the flow through the fresh air side to the point of practically blocking it.

After the additional throttle plate was installed in the fresh air side of the combustor, calibration testing was performed using the dynamometer to drive the engine at the approximate cranking speed of 170 RPM. The results of that testing are shown in Table 6. The testing began with the throttle positioned such that the flow area was approximately equal to the 0.75 in^2 minimum achievable with the shim as described above. The first two samples taken indicate a lean mixture being supplied to the engine. To enrich the mixture being fed to the engine, the fresh air throttle plate was then positioned at its closed position for the remainder of the testing. The samples from the engine exhaust were taken with the engine spark plugs not firing. The sample mixture then represents the mixture being fed to the engine. As shown in Table 6, the CO content of the combustor product stream is fairly insensitive to the percent GPO, above 10% GPO.

As shown in Table 5, the %GPO value of 10 corresponds to the value programmed in the engine controller GPO lookup table for cranking speeds. Thus, the programmed GPO table should be the correct fueling schedule for the combustor for starting the engine with the fresh air throttle fully closed. The test cell was cooled to -29°C and

Table 6 Calibration testing with fresh air throttle

% GPO	% CO	% CO_2	% O_2	Sample Location
10	12.7	7.9	0.7	combustor
10	5.7	3.9	11.8	exhaust
10	6.5	4.3	10.7	exhaust
10	12.4	7.7	0.7	combustor
12	12.7	8.1	0.7	combustor
12	12.8	8.0	0.7	combustor
14	12.5	7.7	0.6	combustor
20	12.2	7.7	0.6	combustor
30	12.3	8.0	0.7	combustor
40	12.1	7.8	0.7	combustor
10	12.4	8.0	0.7	combustor
10	6.3	4.3	10.0	exhaust

several start attempts were made at that temperature and then later at other temperatures. None of the tests resulted in a truly successful start. The engine would fire and run at approximately 300 RPM, with typical results shown in Figure 30. Examination of the engine spark plugs after the start attempts showed extreme wetting (the plugs dripped when removed from the engine) which was no doubt causing fouling. The plug fouling problem was compounded by the lowered battery voltage during cranking at low temperature. If the battery voltage is below 10 volts, the coils may not be able to fire fouled plugs (Electromotive, 1995). It was suspected that the screen device may not be performing properly. It was removed and found to have an area that had burned through. The screen was replaced with a new one, and the 1.59 gm/sec quenching nozzle was re-activated. However, the spark plug wetting problem persisted.

The spark plug wetting problem was of course due to the high content of H_2O in the combustor product stream. The water vapor, laden with methanol and partial combustion products, condenses when exposed to the cold surfaces of the engine. This is particularly true in the cylinder, where additional condensation occurs due to the compression of the incoming charge on the compression stroke. The cavity between the spark insulator and the metal body is especially prone to collect the condensate. Since no practical method exists to prevent the H_2O from forming in the combustor product stream (a catalyst would not be practical), it was decided to attempt to condense the

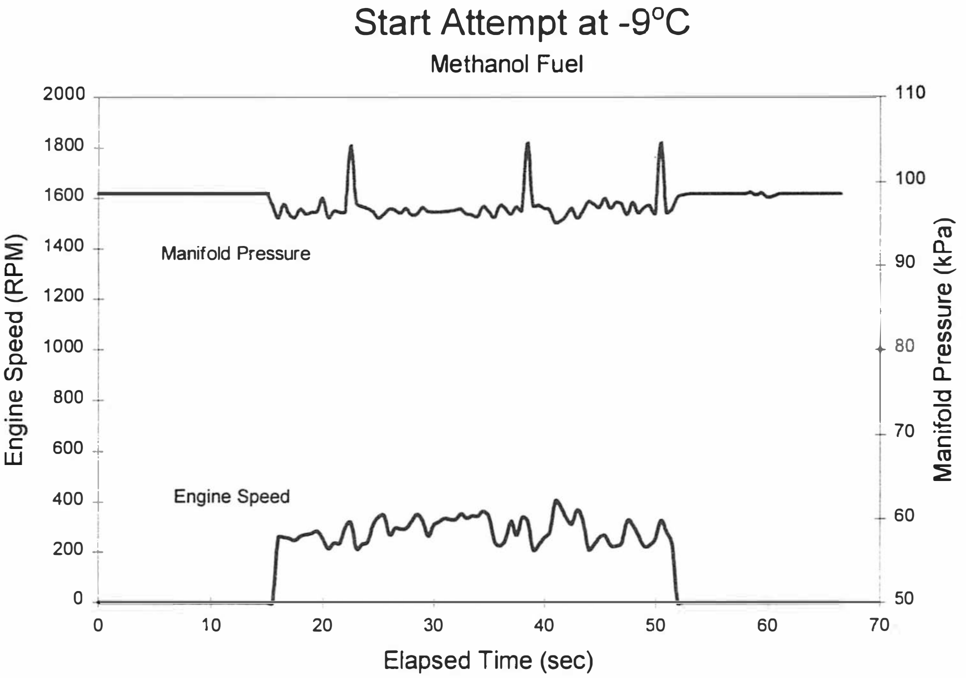


Figure 30 Start attempt after fresh air throttle installation

water vapor before it reaches the cylinder. Heat exchangers were designed and installed in the combustor as shown in Figure 31. A similar heat exchanger was installed in the plenum above the intake manifold. Using an external auxiliary pump, coolant from the radiator was circulated through the three heat exchangers in parallel and returned to the radiator.

The test cell was cooled to approximately -10°C (the highest setting on the thermostat) to test the effectiveness of the heat exchangers. To reduce the amount of vapor to be condensed by the heat exchangers, the quenching nozzle was disabled. The composition of the combustor product stream was measured as 12.4% CO, 6.6% CO_2, and 2.2% O_2. The composition of the sample taken from the exhaust pipe was 6.5% CO, 3.8% CO_2, and 10.8 O_2 . This composition indicates a lean mixture being fed to the engine, but since hydrogen has a wide flammability range, the mixture being fed to the engine should be combustible even though lean, as demonstrated earlier.

The engine started and idled at approximately 800 RPM for several repeated tests, over several days, at this temperature. It was later suspected that these start attempts and subsequent periods of idling (some were somewhat lengthy periods of idling) caused the melting of the block supporting the fine wire screen. After these successful start attempts, the cold start performance of the engine was severely degraded. The engine would fire and run at approximately 300-400 RPM; sometimes, assist from the starter

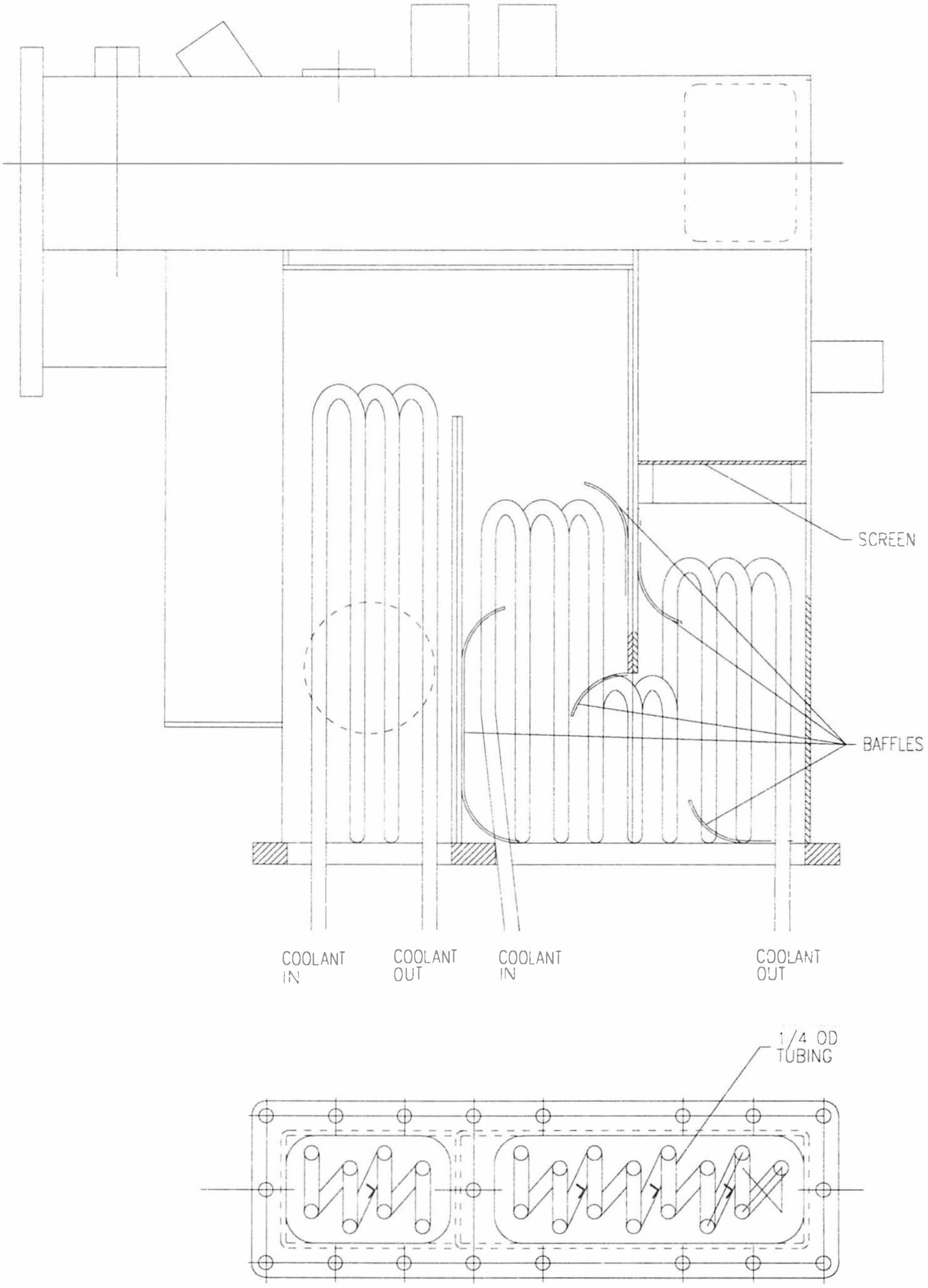


Figure 31 Heat exchanger installation

was required to maintain even this speed. Results were similar to those shown in Figure 30. Samples taken from the combustor, however, showed adequate combustibles to start the engine.

Over the next several weeks, several problems were found and addressed which improved the performance slightly, but the engine still would not reach idle speed. The battery voltage drop was found to be affecting the fuel pressure due to the fuel pump slowing down at the lower voltage. The fuel pump was placed on an independent power supply to maintain the fuel pressure during cranking.

The solenoid valve in the fuel supply line to the combustor nozzles was found to wired such that the switch was energized at all times while the engine controller was on, such that if the combustor fuel switch was inadvertently left on after a test, the fuel continued to flow to the combustor. This was thought to be causing over-fueling of the combustor, so the valve was rewired to be open only when the combustor fuel switch was closed and the engine was turning.

The transmission input shaft had become stuck in the pilot bearing in the rear of the crankshaft due to either corrosion or thermal expansion differences or both. At low temperatures, even with the transmission in neutral, the dynamometer would turn as a result of cranking the engine. It must be pointed out that even though the transmission is a manual shift type, it is lubricated using automatic transmission fluid. The torque to turn and accelerate the dynamometer must be supplied by the engine in addition to what is required to accelerate the engine itself. This problem could be overcome, however, by simply motoring the dynamometer at a speed to match the engine speed during cranking and starting. A very important consequence of this problem was the generation of engine friction data at low temperature as shown in Figure 32. Although this data is somewhat engine specific, it clearly illustrates the increase in engine friction as the temperature decreases. This information will be incorporated in the engine model to more realistically determine the operating parameters required for starting at low temperatures.

The engine controller used has a feature that sets the spark advance to 0° BTDC for any engine speed below 400 RPM. The controller logic assumes that, at engine speeds below 400 RPM, the engine is being cranked and is not running. The spark advance may be adjusted via the PC interface once the engine reaches 400 RPM, but may not be changed below 400 RPM. It was felt that the diluted mixture being fed to the engine may require some spark advance due to the dilution effect on the flame propagation. The crank-mounted trigger wheel that is read by the magnetic sensor to determine crankshaft position was modified to allow rotating the wheel with respect to the crankshaft. This allowed several mounting positions to achieve a mechanical spark advance of 5°, 10°, 15°, 20°, 25°, or 35° BTDC. The engine controller would still set the spark advance to 0° BTDC; however, the actual spark advance below 400 RPM would be the amount the trigger wheel had been rotated with respect to the crankshaft.

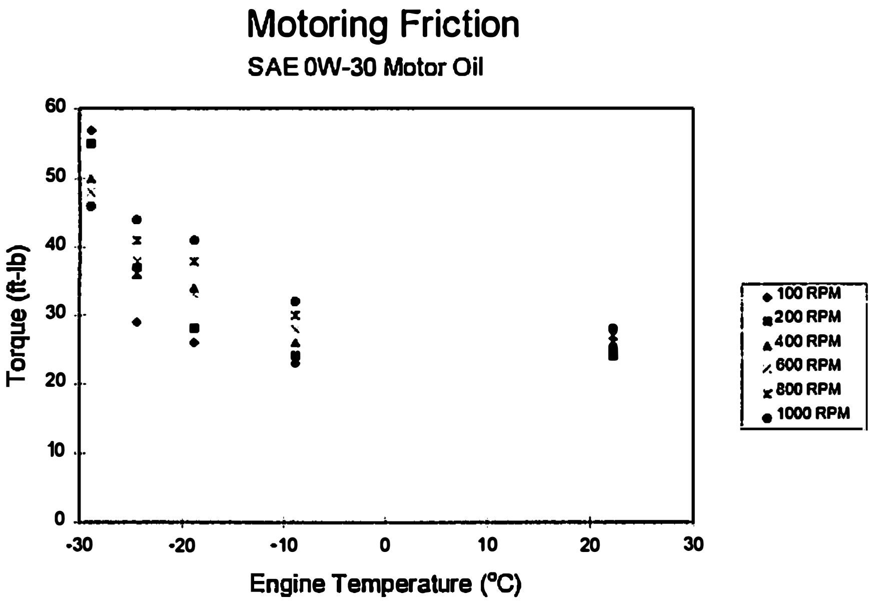


Figure 32 Engine friction at various temperatures

The actual spark advance at all operating points will be similarly offset. There is little potential for damage due to excessive spark advance, however, at the engine operating speeds under consideration.

After all these developments, the engine was still capable of running at only about 400 RPM at all temperatures. We knew that the system was capable of much better performance, as previously demonstrated. However, the combination of parameters that produced a successful start previously could not be simply repeated since the combustor had been modified in the interim. After reviewing the test results up to this point, a problem with the screen became the popular diagnosis of the problem. Since the installation of the heat exchangers and baffles had obscured the view from the sight glass of the screen, the combustor was disassembled to inspect the screen. The screen was found to be intact; however, the aluminum block to which the stainless steel screen was attached had melted into several pieces. The screen was found lying on the floor of the third leg of the combustor and thus was not performing its intended function. A new mounting block was fabricated from stainless steel and the screen assembly replaced. No other anomalies were found, and the combustor was reassembled.

The test cell was cooled to -10°C and several starting attempts were made to determine the effect of mechanical advance on the cold start performance. It was found that 10° of mechanical spark advance gave the best results- the engine started in 8 seconds and idled at 600 - 700 RPM. Adjusting the number of steps that the Idle Air Control (IAC) resets after opening all the way at power up resulted in improved performance; a value of 50 steps to reset was found to be optimum. The most improvement, however, was a result of opening the fresh air throttle slightly. The effect of opening the throttle was to increase the equivalence ratio in the combustor while maintaining approximately the same equivalence ratio in the engine, which is lean. This adjustment resulted in consistent starts at temperatures as low as -20°C, with less than consistent performance at -30°C. Encouraged by these results, methanol compatible fuel injectors were installed in the engine for operational testing of the transition to the main fuel injection system. Scrap injectors had been used not connected to the engine controller during the cold start testing to avoid damaging the ones planned for use in operational testing. The engine controller had previously been calibrated for normal operation on methanol fuel.

Transition Testing

For development of the transition schedule from operation on the combustor alone to operation on the main fuel injection system alone, a switch was installed in the power lead to the main fuel injectors. This switch would allow turning on and off the main fuel injection system; with the switch off, the output from the controller to the fuel injectors (they are pulled to ground) does nothing since there is no power to the injectors. When the switch is turned on to activate the main fuel injection system, the injectors are driven at whatever schedule the controller has determined for that operating point. Since the engine is operating lean during starting, the engine controller is trying to enrich the mixture by adjusting the injector-on time. Fortunately, the engine controller acts quickly enough to adjust the mixture before it becomes too rich. The combustor fuel system was already controlled by a switch, allowing the combustor operation to be controlled.

For operation on methanol, it was found that the main fuel injection system could be activated fairly soon after the engine reach idle speed operating on the rich combustor. The engine is operating on a lean mixture at that point and the engine controller adjusts the amount of liquid fuel injected based on the input from the Exhaust Gas Oxygen (EGO) sensor. The EGO voltage reaches a steady state value around the stoichiometric set point and the engine runs with both the combustor and the main fuel injection system operating. It should be pointed out that, for this engine controller, the EGO signal is not used when the coolant temperature is below 0^oC. To implement this strategy, a 10 kΩ resistor was placed across the connector for the coolant temperature sensor to simulate a 10^oC signal from the thermistor. When the engine surfaces have warmed up sufficiently, the fuel supply to the combustor is turned off, and the engine continues to run on just the main fuel injection system.

Figure 33 shows a start to transition at -9^oC. The temperature of the test is considered to be the temperature reported by a thermocouple installed through the block water jacket and in contact with the cylinder wall on the coolant side. The rapid increase in the EGO voltage indicates the point at which the main fuel injection system was activated. The voltage reaches a steady value when the engine equivalence ratio comes under control.

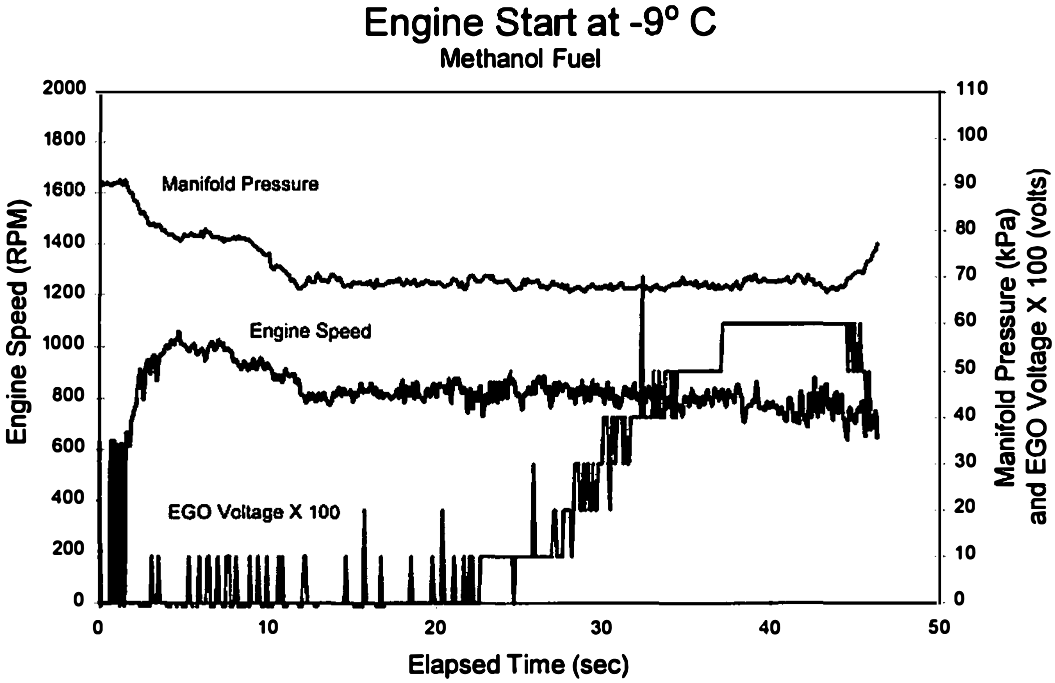


Figure 33 Start and transition to idle at -9°C

Figures 34 shows a start and transition to idle on the main fuel injection system at -6°C. The point at which the main system is turned on is clear and Figure 35 shows the point at which the combustor is turned off. Also shown in Figure 35 is the engine running at approximately 3500 RPM on the main fuel injection system alone after less than three minutes. Figures 36 and 37 illustrate a start and transition at -18°C. The point at which the main fuel injection system is activated is evident. The first attempt occurred after approximately 2 minutes and resulted in the engine stalling. The next three attempts were made immediately after the engine reached idle speed operating on the combustor, and all three resulted in the engine stalling. The fifth attempt at transition came after allowing the engine to idle on the combustor for approximately 90 seconds and resulted in the engine continuing to run on both systems and finally the main fuel injection system alone.

Ethanol Testing

The performance of the combustor using ethanol was evaluated. The ethanol used was fuel ethanol or E95 (95% ethanol and 5% denaturant). The only adjustments to the combustor system to switch from methanol to ethanol were made to the fueling parameters programmed in the engine controller. The method of determining the injector-on time employed by the controller allows adjusting a single parameter, essentially the maximum injector-on time, to recalibrate the fueling schedule for a different fuel. The controller does not display the correct desired or actual air fuel ratio (it displays the equivalent values for gasoline), but the equivalence ratio is correct since

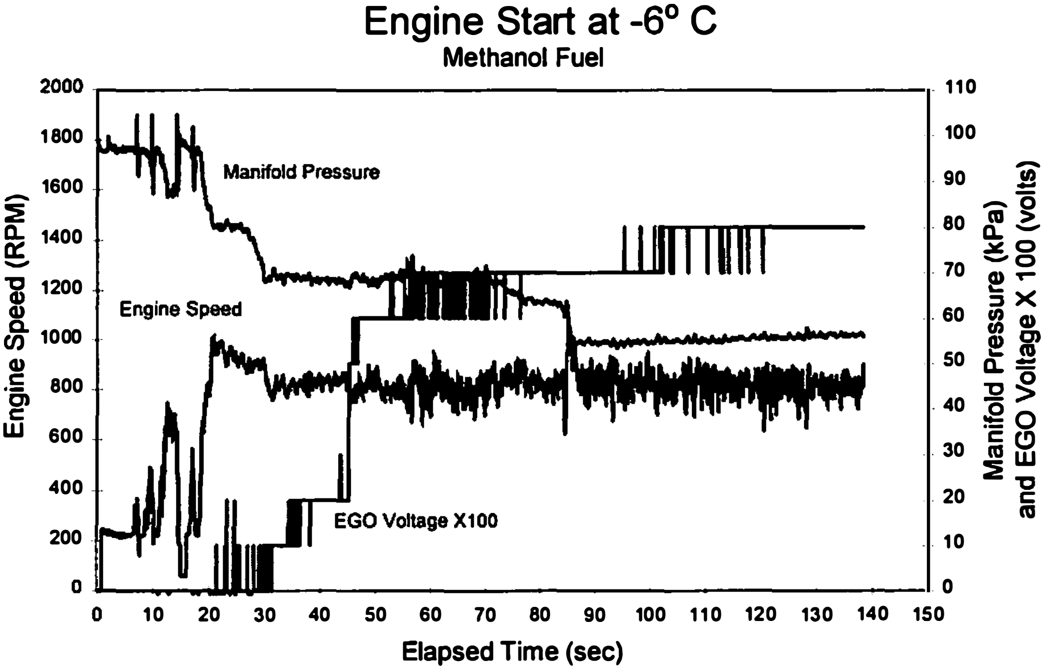


Figure 34 Start and transition at -6°C

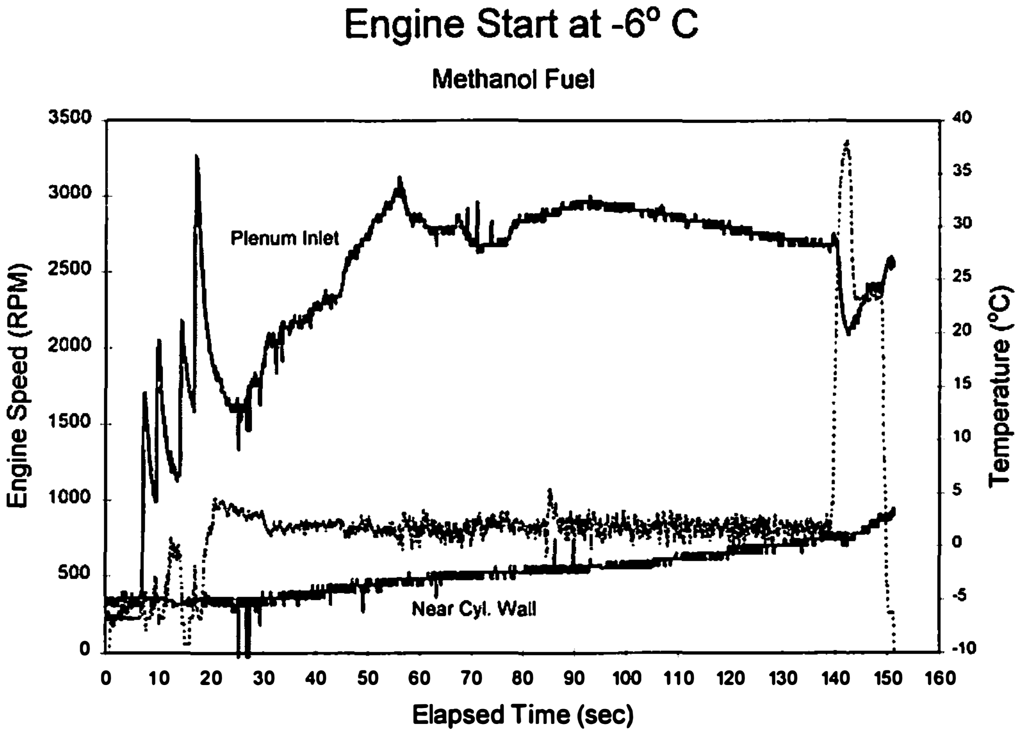


Figure 35 Temperatures for start and transition at -6°C

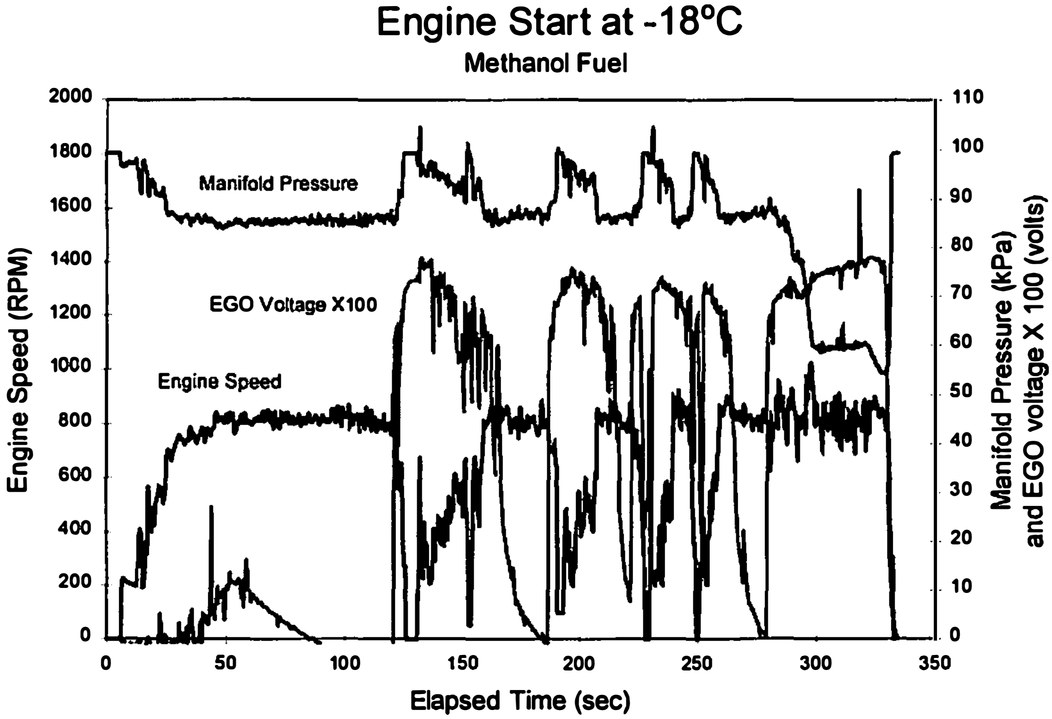


Figure 36 Start and transition at -18°C

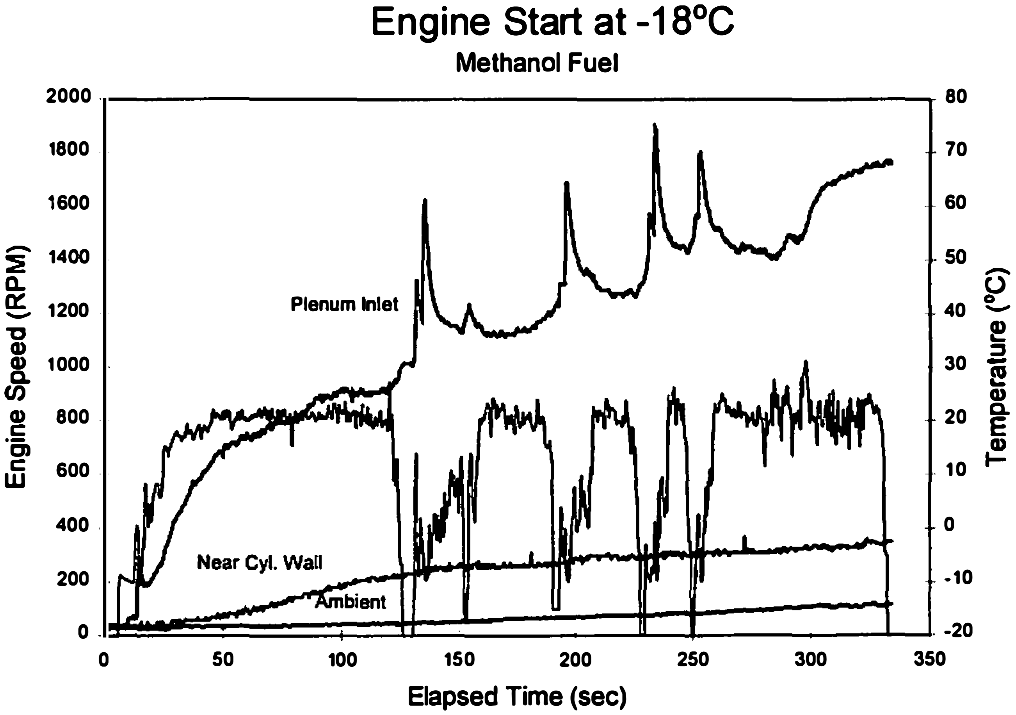


Figure 37 Temperatures for start and transition at -18°C

the ratio is determined and controlled by the EGO signal. The GPO table was also adjusted for the different stoichiometry of ethanol. With no further modifications, the combustor was tested using ethanol.

Figures 38 and 39 show a start and transition at -11°C using ethanol. The point at which the main fuel injection system is turned on is evident. The engine continues to run with both systems operating until the combustor is turned off causing the engine to stall. The engine was restarted and the transition attempted again, resulting in the engine running for a short period of time after the combustor is switched off. The third attempt resulted in the engine continuing to run after the combustor was switched off.

Figures 40 and 41 show a start and transition at -20°C. The point at which the main fuel injection system is activated is evident. Rather than turning the combustor off completely, the amount of fuel to the combustor was reduced by entering a negative GPO offset via the PC link to the engine controller. This method of phasing out the combustor operation via the fuel flow with constant air flow converts the rich combustor eventually to an air heater before being switched off completely. The points at which the GPO was offset by 10% each step are evident by a downward spike in the EGO voltage as well as a slope change in the plenum inlet temperature trace. This step-wise method was an improvement over the binary control previously used for the combustor, but the operator's impatience caused the engine to stall. The engine was restarted and the transition was completed. It is worthwhile to note that nothing unexpected occurs when the engine stalls and is restarted.

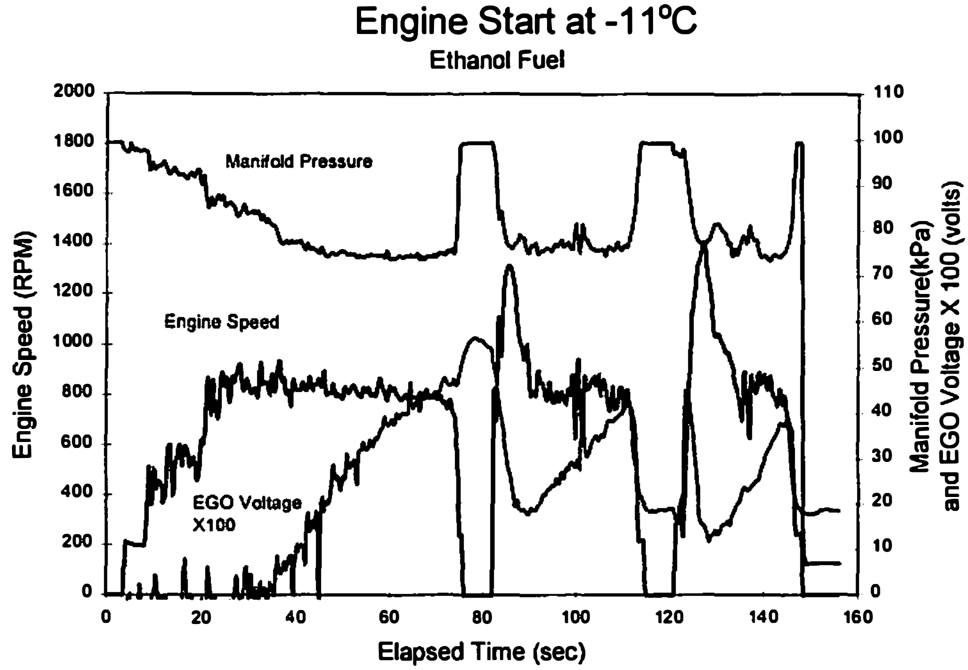


Figure 38 Start and transition at -11°C with ethanol

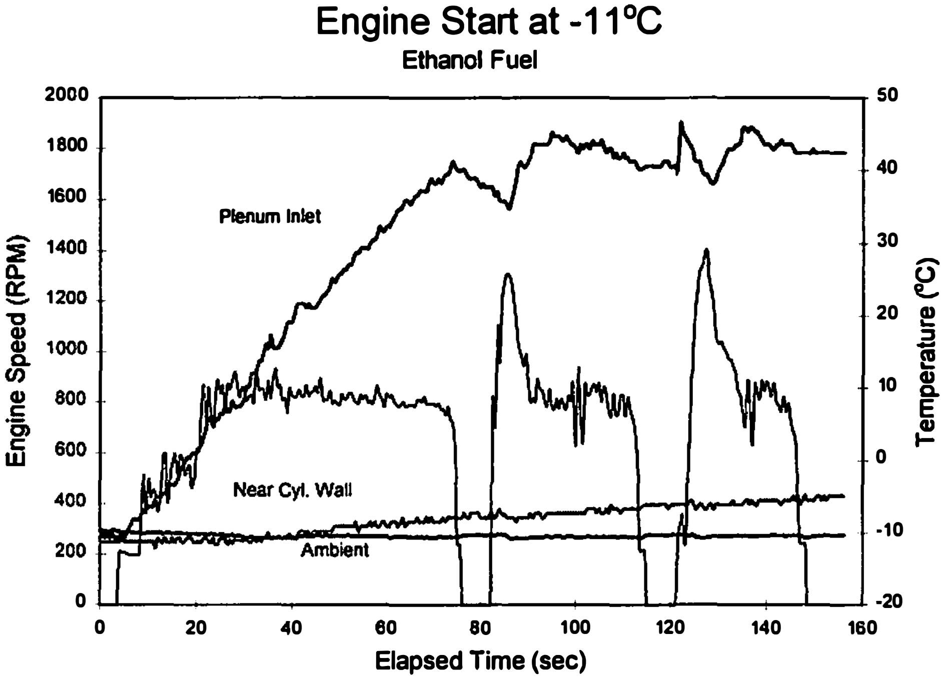


Figure 39 Temperatures for start and transition at -11°C with ethanol

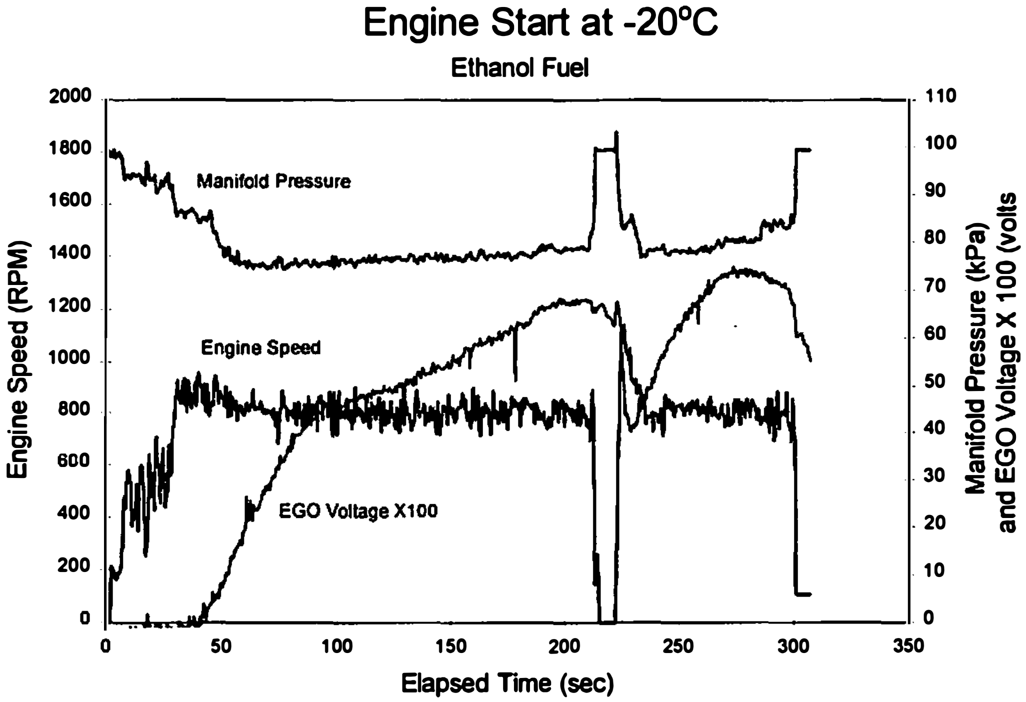


Figure 40 Start and transition at -20°C using ethanol

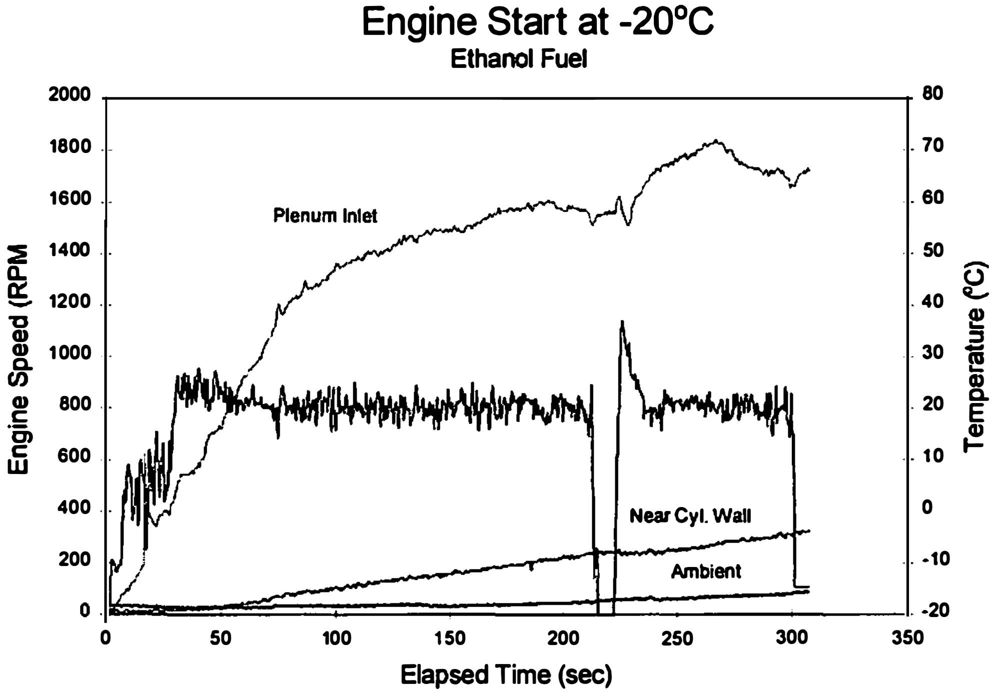


Figure 41 Temperatures for start and transition at -20°C using ethanol

Figures 42 and 43 show a start and transition at -30°C. Note the extremely long time required for the engine to accelerate to idle speed. The point where the main fuel injector system was activated is evident. Two steps of the step-wise schedule for combustor phase-out are evident in the EGO voltage trace. The engine stalled after the second step indicating that the schedule should be lengthened. The trace for the plenum inlet temperature shows that the combustor could be operated for a longer time with the temperature still in a reasonable range.

Figures 44 and 45 show a start and transition at -11°C. The step-wise method of combustor phase-out was employed. Note the plenum inlet temperature trace; the temperature reaches a somewhat steady value as the fuel to the combustor is decreased in the step-wise fashion indicating that the schedule may be extended to achieve a smooth transition to operation on the main fuel injection system.

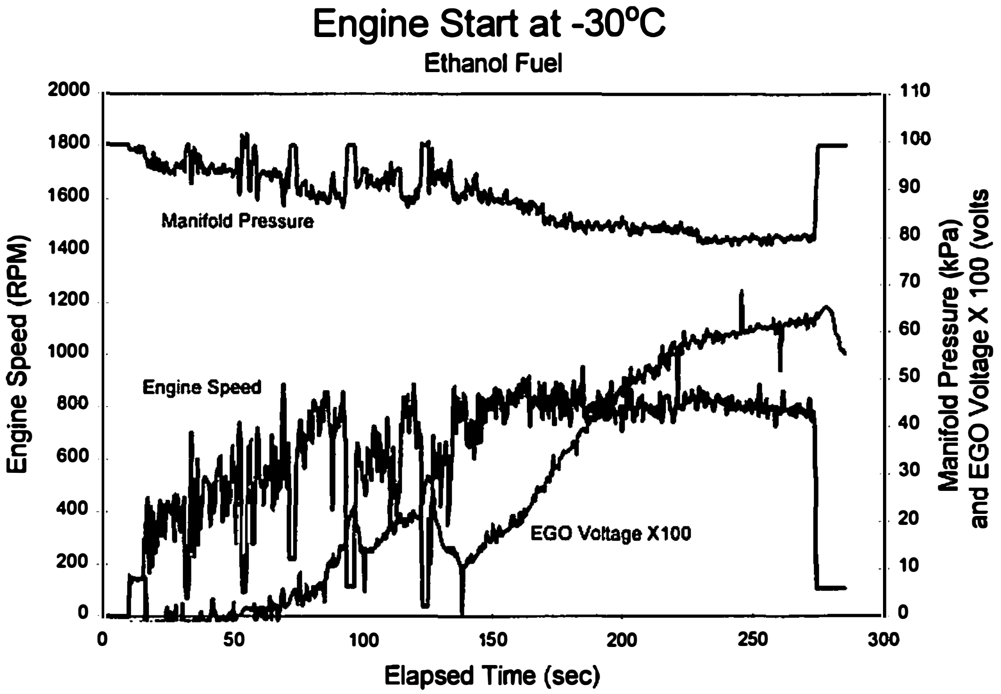


Figure 42 Start and transition at -30°C using ethanol

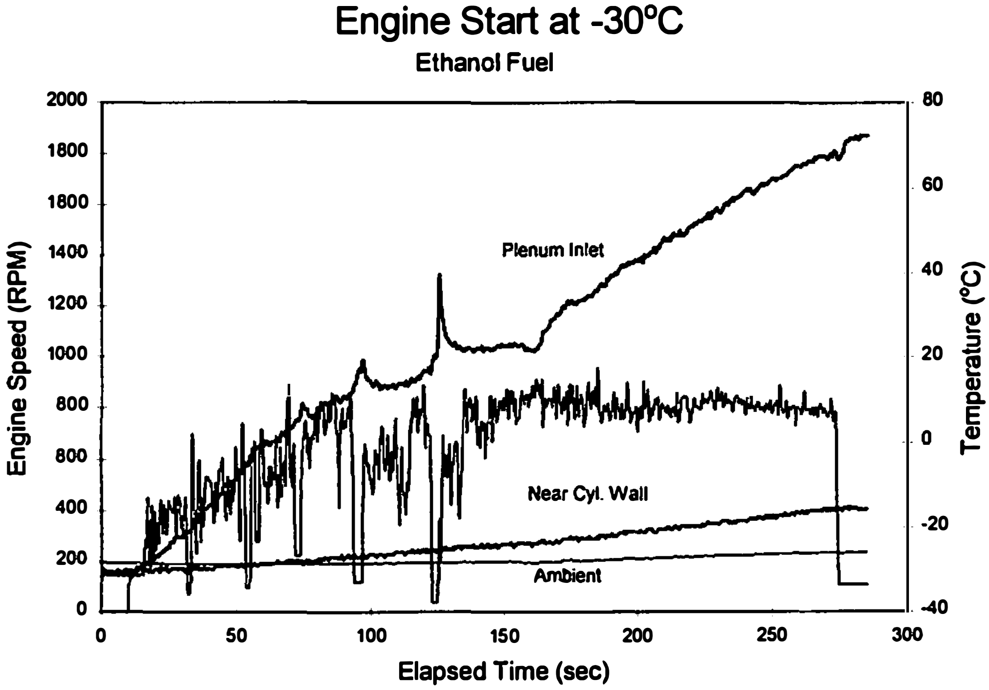


Figure 43 Temperatures for start and transition at -30°C

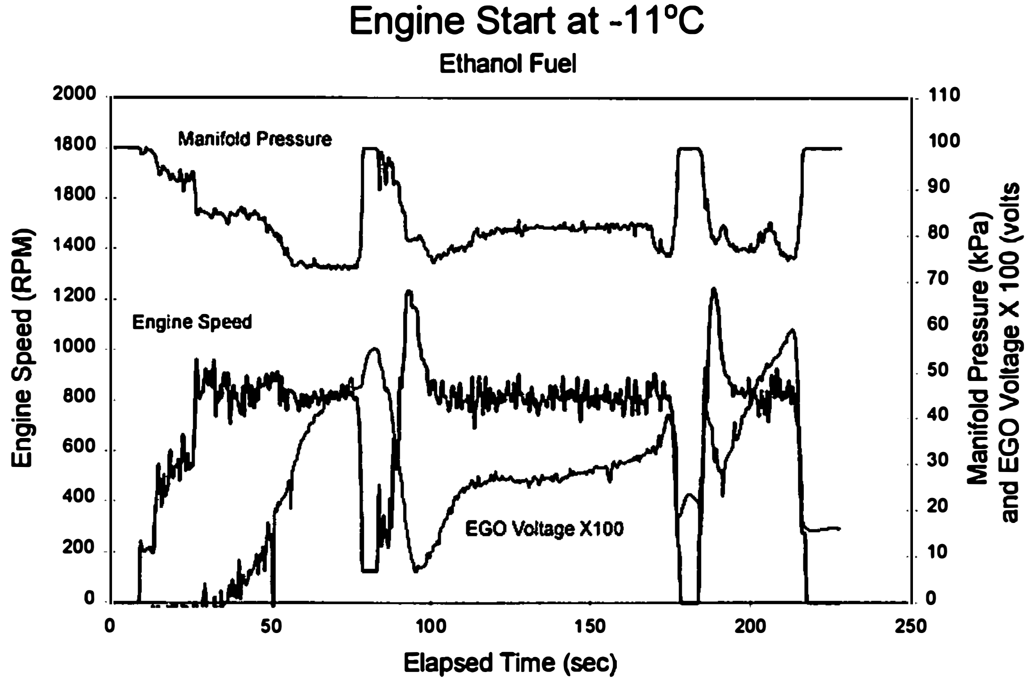


Figure 44 Start and transition at -11°C using ethanol

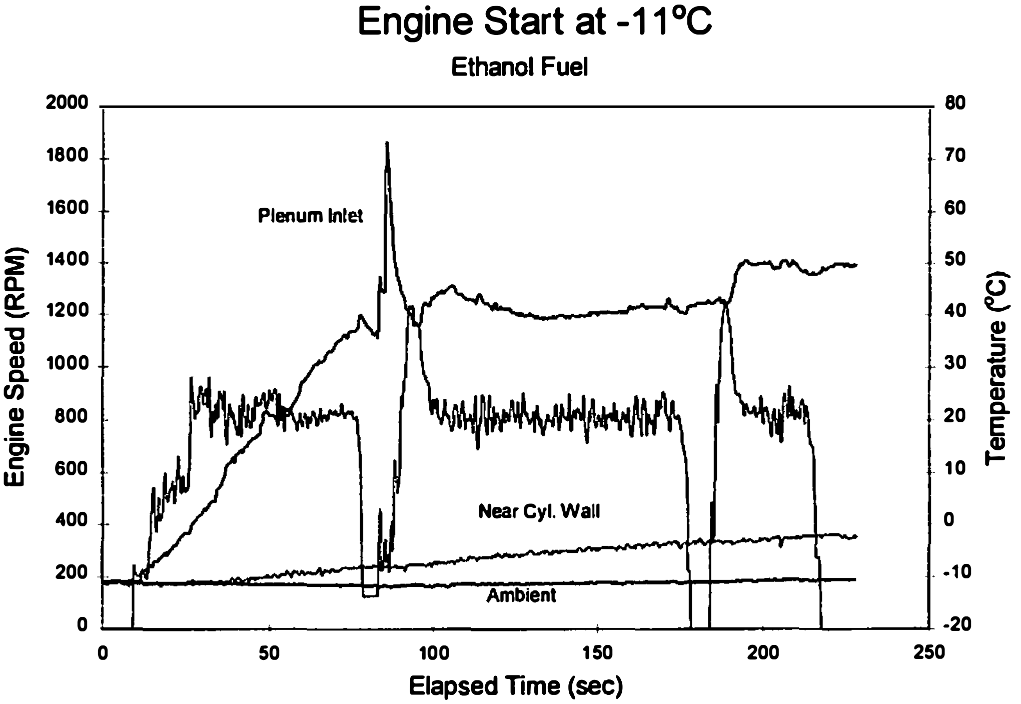


Figure 45 Temperatures for start and transition at -11°C using ethanol

Chapter V

Conclusions and Recommendations for Future Work

The rich combustor concept for improving the cold start performance of alcohol fueled engines has been shown to be practical. The ability to start an alcohol fueled engine to temperatures as low as -30°C has been demonstrated. Although -30°C starts were not consistently successful, reliable cold starting has been demonstrated at -20°C for both methanol and ethanol; a schedule for the transition to operation on the main fuel injection system has been developed but not demonstrated reliably. Reliable cold starting and transition to operation on the main fuel injection system has been demonstrated for approximately -10°C for both methanol and ethanol.

Although the system was not installed on a vehicle, most of the vehicle integration issues were resolved during the development of the rich combustor system. Reordering of interest in the alcohols resulted in a change in the focus from methanol to ethanol. The ethanol testing was performed in lieu of emissions testing the system using methanol.

The development of the transition to main fuel injection system schedule should be refined to give a smooth transition at all temperatures. The test results presented indicate that it should be achievable. The ability to phase-in the main fuel injection system rather than switching it on would allow an earlier beginning of the transition process and would supply additional fuel to the lean-operating engine operating on the combustor alone. Modifications to the engine controller software would be required to achieve this phase-in control of the main fuel injection system. Once this transition schedule is optimized, off-idle operation of the system should be developed with the goal of start up and drive away.

References

Campbell, A.S. (1979), Thermodynamic Analysis of Internal Combustion Engines. John Wiley & Sons, New York, 1979.

Dempsey, A. B. and Goetz, W. A. (1986), "Cold Start Testing of Methanol Powered Vehicles", Technical Report No. 5030-1, ES/CAFU/86-16, Transport Canada, Centre for Alternative Fuel Utilization, Ottawa, Ontario, 1986.

Electromotive (1995), "Total Engine Control Engine Management Systems-Installation and User's Manual", Electromotive, Inc., Chantilly, VA, 1995.

Gardiner, D. P. (1993), "Sub-Zero Cold Starting of a Port-Injected M100 Engine Using Plasma Jet Ignition and Prompt EGR", SAE Paper No. 930331, 1993.

Höchsmann, G. (1989), "Cold Start Capability of Methanol-Fueled Engines", Porsche AG R&D Center, Weissach,Germany, 1989.

Hodgson, J. W. (1994), "The Development of a Device for Cold Starting Alcohol Fueled Vehicles and Reducing Cold Start Emissions from Gasoline Fueled Vehicles", Proposal to National Renewable Energy Laboratory, 1994.

Hodgson, J. W., Pike, M. S., Guglielmello, T. J. (1993), "Development of a Cold-Start Device for Methanol-Fueled Engines", SAE Paper No. 932772, (also published in SAE Special Publication SP-995), 1993.

Hodgson, J. W., and Lamas, J. A (1990). "A Simple Algorithm For Determining The Equilibrium Composition Of Gaseous C/H/O/N Systems Involving Ten Species", 1990 ASEE-SE Section Proceedings, April 8-19, 1990, Charlotte, North Carolina.

Iwai, N., Akai, M., Hosogai, d., Nishimura, j., and Saori, H. (1984), "Cold Startability Improvement of Neat Methanol S.I. Engine by Using Ultrasonic Partial Oxidation Combustor"

Karpuk, M. E. (1989), “Design and Testing of a Dissociated Methanol Vehicle”, DOE/CH/10093-H5, U. S. Department of Energy, Office of Vehicle and Engine R&D, 1989.

Karpuk, Michael E., and Cowley, Scott W. (1988), “On Board Dimethyl Ether Generation to Assist Methanol Engine Cold Starting”, SAE Paper No. 881678, 1988.

Lalk, T. R., McCall, D. M., and McCanlies(1984), “Evaluation of Dissociated and Steam-Reformed Methanol as Automotive Engine Fuels”, COE/NBB-0064, DOE/NASA/01420-1, NASA Cr-168242, U. S. Department of Energy, Office of Vehicle and Engine R&D, 1984.

Obert, E. F. (1973), Internal Combustion Engines and Air Pollution, Harper and Row, New York, 1973.

Pefley, R. K., and Browning, L. H.(1986), "Research and Development of Alcohol Fuel Usage in Spark-Ignited Engines", DOE/CE/50036-H1, U. S. Department of Energy, Office of Transportation Systems, 1986.

Pefley, R. K., Pullman, B., and Whitten, G., (1984), "The Impact of Alcohol Fuels on Urban Air Pollution: Methanol Photochemistry Study", DOE/CE/50036-1, U. S. Department of Energy, Office of Vehicle and Engine R&D, 1984.

Pettersson, L. and Sjöström, K. (1991), "Onboard Hydrogen Generation By Methanol Decomposition for the Cold Start of Neat Methanol Engines", International Journal of Hydrogen Energy, Vol. 16, No. 10, pp. 671-676, 1991.

Perry's Chemical Engineers Handbook, (1993), McGraw-Hill, 1993.

Ross, Marc and DeCicco, John (1993), "An Updated Assessment of the Near-Term Potential for Improving Automotive Fuel Economy", American Council for an Energy-Efficient Economy, 1993.

Appendices

Appendix A

```
Sub Form_Load ()
   optMetric.Enabled = False
  On Local Error GoTo Err14
   Open "tiresize.txt" For Input As 1
   Do While Not EOF(1)
     Line Input #1, TireSize$
     lstTireSize.AddItem TireSize$
   Loop
   Close #1
   Exit Sub

Err14:
   msg$ = "I CAN'T FIND THE FILE tiresize.txt" + Chr$(13) + Chr$(10) + "MAKE SURE
ALL THE REQUIRED FILES ARE AVAILABLE"
   Value% = MsgBox(msg$, 1)
   If Value% >= 0 Then
      Exit Sub
   End If
   Resume

End Sub


Sub cmdExit_Click ()
 frmTires.Hide
End Sub


Sub lstTireSize_Click ()
   TireDia = Val(Mid$(lstTireSize.Text, 21, 10))
    If optMetric.Value = -1 Then
      TireDia = TireDia * 25.4
    End If
    txtTireDiameter.Text = Format$(TireDia, "####.00")
    clipboard.SetText txtTireDiameter.Text
End Sub


Sub optEnglish_Click ()
  lblUnits.Caption = "in"
End Sub


Sub optMetric_Click ()
 lblUnits.Caption = "mm"
```

```
End Sub

VERSION 2.00
Begin Form frmEngineParameters
  Caption         =   "Engine Parameters"
  ClientHeight    =   4890
  ClientLeft      =   1080
  ClientTop       =   1770
  ClientWidth     =   6945
  FontBold        =   -1  'True
  FontItalic      =   0   'False
  FontName        =   "System"
  FontSize        =   9.75
  FontStrikethru  =   0   'False
  FontUnderline   =   0   'False
  Height          =   5295
  Left            =   1020
  LinkTopic       =   "Form1"
  ScaleHeight     =   4890
  ScaleWidth      =   6945
  Top             =   1425
  Width           =   7065
  Begin Frame fraAmbientConditions
    Caption         =   "Ambient Conditions"
    Height          =   1095
    Left            =   120
    TabIndex        =   41
    Top             =   2160
    Width           =   2415
    Begin TextBox txtAmbientPressure
      Height          =   285
      Left            =   1250
      TabIndex        =   3
      Top             =   690
      Width           =   495
    End
    Begin TextBox txtAmbientTemperature
      Height          =   285
      Left            =   1250
      TabIndex        =   2
      Top             =   330
      Width           =   495
    End
    Begin Label Label19
      Caption         =   "atm"
      Height          =   255
      Left            =   1800
    TabIndex     = 42
    Top          = 720
    Width        = 300
  End
  Begin Label Label12
    Caption      = "Pressure"
    Height       = 255
    Left         = 120
    TabIndex     = 45
    Top          = 720
    Width        = 855
  End
  Begin Label Label7
    Caption      = "K"
    Height       = 255
    Left         = 1800
    TabIndex     = 44
    Top          = 360
    Width        = 135
  End
  Begin Label Label6
    Caption      = "Temperature"
    Height       = 255
    Left         = 120
    TabIndex     = 43
    Top          = 360
    Width        = 1095
  End
End
Begin Frame fraColdStart
  Caption      = "Cold Start Strategy"
  Height       = 1935
  Left         = 4920
  TabIndex     = 34
  Top          = 2160
  Width        = 1935
  Begin TextBox txtCombustorEquivRatio
    BackColor    = &H00FFFFFF&
    ForeColor    = &H00000000&
    Height       = 285
    Left         = 840
    TabIndex     = 37
    Top          = 660
    Width        = 960
  End
  Begin OptionButton optRichCombustor
    Caption      = "Rich Combustor"
    Height        =  255
    Left          =  120
    TabIndex      =  12
    Top           =  360
    Value         =  -1 'True
    Width         =  1695
  End
  Begin OptionButton optEnrichment
    Caption       =  "Enrichment"
    Height        =  255
    Left          =  120
    TabIndex      =  36
    Top           =  1080
    Width         =  1455
  End
  Begin TextBox txtEnrichmentEquivRatio
    BackColor     =  &H00FFFFFF&
    Enabled       =  0 'False
    ForeColor     =  &H00000000&
    Height        =  285
    Left          =  840
    TabIndex      =  35
    Top           =  1400
    Width         =  960
  End
  Begin Label lblEnrichmentEquivRatio
    BackColor     =  &H00FFFFFF&
    Caption       =  "Equiv Ratio"
    Enabled       =  0 'False
    ForeColor     =  &H00000000&
    Height        =  375
    Left          =  240
    TabIndex      =  39
    Top           =  1320
    Width         =  495
  End
  Begin Label lblCombustorEquivRatio
    BackColor     =  &H00FFFFFF&
    Caption       =  "Equiv Ratio"
    ForeColor     =  &H00000000&
    Height        =  375
    Left          =  240
    TabIndex      =  38
    Top           =  600
    Width         =  495
  End
End
```

```
Begin Frame fraFuelManagement
  Caption       =  "Fuel Management"
  Height        =  1815
  Left          =  4920
  TabIndex      =  30
  Top           =  120
  Width         =  1935
  Begin OptionButton optLookupTable
    Caption       =  "Lookup Table"
    Enabled       =  0  'False
    Height        =  255
    Left          =  120
    TabIndex      =  33
    Top           =  1440
    Width         =  1455
  End
  Begin TextBox txtEquivRatio
    BackColor     =  &H00FFFFFF&
    Enabled       =  0  'False
    ForeColor     =  &H00000000&
    Height        =  285
    Left          =  840
    TabIndex      =  32
    Top           =  1020
    Width         =  960
  End
  Begin OptionButton optConstantAF
    Caption       =  "Constant A/F"
    Height        =  255
    Left          =  120
    TabIndex      =  31
    Top           =  720
    Width         =  1455
  End
  Begin OptionButton optStoichiometric
    Caption       =  "Stoichiometric"
    Height        =  255
    Left          =  120
    TabIndex      =  11
    Top           =  360
    Value         =  -1 'True
    Width         =  1455
  End
  Begin Label lblEquivRatio
    BackColor     =  &H00FFFFFF&
    Caption       =  "Equiv Ratio"
    Enabled       =  0  'False
    ForeColor       =   &H00000000&
    Height          =   375
    Left            =   240
    TabIndex        =   40
    Top             =   960
    Width           =   495
  End
End
Begin CommandButton cmdSelectEngine
  Caption         =   "Select Engine"
  Enabled         =   0   'False
  Height          =   375
  Left            =   2640
  TabIndex        =   29
  Top             =   3120
  Width           =   2055
End
Begin CommandButton cmdEngineParametersCancel
  Caption         =   "Cancel"
  Height          =   375
  Left            =   5640
  TabIndex        =   14
  Top             =   4320
  Width           =   1215
End
Begin CommandButton cmdEngineParametersOK
  Caption         =   "OK"
  Default         =   -1  'True
  Height          =   375
  Left            =   4200
  TabIndex        =   13
  Top             =   4320
  Width           =   1215
End
Begin Frame fraFuel
  BackColor       =   &H00FFFFFF&
  Caption         =   "Fuel"
  ForeColor       =   &H00000000&
  Height          =   1935
  Left            =   120
  TabIndex        =   28
  Top             =   120
  Width           =   2415
  Begin Frame fraFuelState
    BackColor       =   &H00FFFFFF&
    Caption         =   "Fuel State"
    ForeColor       =   &H00000000&
    Height       =  525
    Left         =  120
    TabIndex     =  46
    Top          =  1300
    Width        =  2175
    Begin OptionButton optGaseous
      BackColor     =  &H00FFFFFF&
      Caption       =  "Gas"
      ForeColor     =  &H00000000&
      Height        =  240
      Left          =  120
      TabIndex      =  48
      TabStop       =  0  'False
      Top           =  240
      Width         =  615
    End
    Begin OptionButton optLiquid
      BackColor     =  &H00FFFFFF&
      Caption       =  "Liquid"
      ForeColor     =  &H00000000&
      Height        =  240
      Left          =  1200
      TabIndex      =  47
      Top           =  240
      Value         =  -1  'True
      Width         =  855
    End
  End
  Begin ListBox lstFuel
    BackColor     =  &H00FFFFFF&
    FontBold      =  0  'False
    FontItalic    =  0  'False
    FontName      =  "Fixedsys"
    FontSize      =  9
    FontStrikethru  =  0  'False
    FontUnderline  =  0  'False
    ForeColor     =  &H00000000&
    Height        =  930
    Left          =  120
    TabIndex      =  1
    Top           =  240
    Width         =  2175
  End
End
Begin Frame fraDimensions
  BackColor     =  &H00FFFFFF&
  Caption       =  "Dimensions"
```

```
ForeColor     =  &H00000000&
Height        =  2895
Left          =  2640
TabIndex      =  20
Top           =  120
Width         =  2055
Begin TextBox txtNumberCylinders
  BackColor     =  &H00FFFFFF&
  ForeColor     =  &H00000000&
  Height        =  285
  Left          =  1080
  TabIndex      =  10
  Text          =  "6"
  Top           =  2400
  Width         =  495
End
Begin TextBox txtCompRatio
  BackColor     =  &H00FFFFFF&
  ForeColor     =  &H00000000&
  Height        =  285
  Left          =  1080
  TabIndex      =  9
  Top           =  1920
  Width         =  495
End
Begin TextBox txtRodLength
  BackColor     =  &H00FFFFFF&
  ForeColor     =  &H00000000&
  Height        =  285
  Left          =  840
  TabIndex      =  8
  Top           =  1410
  Width         =  735
End
Begin TextBox txtStroke
  BackColor     =  &H00FFFFFF&
  ForeColor     =  &H00000000&
  Height        =  285
  Left          =  840
  TabIndex      =  7
  Top           =  930
  Width         =  735
End
Begin TextBox txtBore
  BackColor     =  &H00FFFFFF&
  ForeColor     =  &H00000000&
  Height        =  285
```

```
      Left            =   840
      TabIndex        =   6
      Top             =   450
      Width           =   735
   End
   Begin Label lblRatio
      BackColor       =   &H00FFFFFF&
      Caption         =   "Comp Ratio"
      ForeColor       =   &H00000000&
      Height          =   495
      Left            =   120
      TabIndex        =   27
      Top             =   1920
      Width           =   615
   End
   Begin Label Label1
      BackColor       =   &H00FFFFFF&
      Caption         =   "No. Cylinders"
      ForeColor       =   &H00000000&
      Height          =   375
      Left            =   120
      TabIndex        =   19
      Top             =   2400
      Width           =   855
   End
   Begin Label Label14
      BackColor       =   &H00FFFFFF&
      Caption         =   "cm"
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   1680
      TabIndex        =   26
      Top             =   1440
      Width           =   240
   End
   Begin Label Label13
      BackColor       =   &H00FFFFFF&
      Caption         =   "cm"
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   1680
      TabIndex        =   25
      Top             =   960
      Width           =   240
   End
   Begin Label Label11
      BackColor       =   &H00FFFFFF&
```

```
    Caption         =   "cm"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   1680
    TabIndex        =   24
    Top             =   480
    Width           =   240
  End
  Begin Label Label10
    BackColor       =   &H00FFFFFF&
    Caption         =   "Rod Lg"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   120
    TabIndex        =   23
    Top             =   1440
    Width           =   720
  End
  Begin Label Label9
    BackColor       =   &H00FFFFFF&
    Caption         =   "Stroke"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   120
    TabIndex        =   22
    Top             =   960
    Width           =   720
  End
  Begin Label Label8
    BackColor       =   &H00FFFFFF&
    Caption         =   "Bore"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   120
    TabIndex        =   21
    Top             =   480
    Width           =   600
  End
End
Begin Frame fraIgnition
  BackColor       =   &H00FFFFFF&
  Caption         =   "Ignition Parameters"
  ForeColor       =   &H00000000&
  Height          =   1455
  Left            =   120
  TabIndex        =   0
  Top             =   3360
```

```
Width       = 2415
Begin SpinButton spnCombDuration
  Height      = 300
  Left        = 1800
  Top         = 960
  Width       = 200
End
Begin SpinButton spnSparkAdvance
  Height      = 300
  Left        = 1800
  Top         = 360
  Width       = 200
End
Begin TextBox txtCombustionDuration
  BackColor   = &H00FFFFFF&
  ForeColor   = &H00000000&
  Height      = 300
  Left        = 1200
  TabIndex    = 5
  Top         = 960
  Width       = 495
End
Begin TextBox txtSparkAdvance
  BackColor   = &H00FFFFFF&
  ForeColor   = &H00000000&
  Height      = 300
  Left        = 1200
  TabIndex    = 4
  Top         = 360
  Width       = 495
End
Begin Label Label18
  BackColor   = &H00FFFFFF&
  Caption     = "DEG"
  ForeColor   = &H00000000&
  Height      = 240
  Left        = 240
  TabIndex    = 18
  Top         = 1200
  Width       = 480
End
Begin Label Label17
  BackColor   = &H00FFFFFF&
  Caption     = "BTDC"
  ForeColor   = &H00000000&
  Height      = 240
  Left        = 240
```

```
      TabIndex     =  17
      Top          =  600
      Width        =  480
    End
    Begin Label Label16
      BackColor    =  &H00FFFFFF&
      Caption      =  "Comb Dur"
      ForeColor    =  &H00000000&
      Height       =  240
      Left         =  120
      TabIndex     =  16
      Top          =  960
      Width        =  975
    End
    Begin Label Label15
      BackColor    =  &H00FFFFFF&
      Caption      =  "Spark Adv"
      ForeColor    =  &H00000000&
      Height       =  240
      Left         =  120
      TabIndex     =  15
      Top          =  360
      Width        =  975
    End
  End
End
  Dim tmpBore$, tmpStroke$, tmpRodLength$, tmpSparkAdvance$,
tmpCombustionDuration$
  Dim tmpNumberCylinders$, tmpCompRatio$, tmpAmbientPressure$,
tmpAmbientTemperature$
  Dim tmpEquivRatio$, tmpEnrichmentEquivRatio$, tmpCombustorEquivRatio$,
tmpFuelSelection%


Sub cmdEngineParametersCancel_Click ()
  txtBore.Text = tmpBore$
  txtStroke.Text = tmpStroke$
  txtRodLength.Text = tmpRodLength$
  txtSparkAdvance.Text = tmpSparkAdvance$
  txtCombustionDuration.Text = tmpCombustionDuration$
  txtNumberCylinders.Text = tmpNumberCylinders$
  txtCompRatio.Text = tmpCompRatio$
  txtAmbientPressure.Text = tmpAmbientPressure$
  txtAmbientTemperature.Text = tmpAmbientTemperature$
  txtEquivRatio.Text = tmpEquivRatio$
  txtEnrichmentEquivRatio.Text = tmpEnrichmentEquivRatio$
  txtCombustorEquivRatio.Text = tmpCombustorEquivRatio$
  lstFuel.ListIndex = tmpFuelSelection%
  optRichCombustor.Value = -1
  optStoichiometric.Value = -1
  frmEngineParameters.Hide

End Sub


Sub cmdEngineParametersOK_Click ()
  If txtAmbientTemperature.Text = "" Then
    MsgBox "PLEASE SPECIFY AN AMBIENT TEMPERATURE"
    txtAmbientTemperature.SetFocus
    Exit Sub
  End If
  If txtAmbientPressure.Text = "" Then
    MsgBox "PLEASE SPECIFY AN AMBIENT PRESSURE"
    txtAmbientPressure.SetFocus
    Exit Sub
  End If
  If txtEquivRatio.Text = "" And optConstantAF.Value = -1 Then
    MsgBox "PLEASE SPECIFY AN EQUIVALENCE RATIO"
    txtEquivRatio.SetFocus
    Exit Sub
  End If
  If txtEnrichmentEquivRatio.Text = "" And optEnrichment.Value = -1 Then
    MsgBox "PLEASE SPECIFY AN EQUIVALENCE RATIO"
    txtEnrichmentEquivRatio.SetFocus
    Exit Sub
  End If
  If txtCombustorEquivRatio.Text = "" And optRichCombustor.Value = -1 Then
    MsgBox "PLEASE SPECIFY AN EQUIVALENCE RATIO"
    txtCombustorEquivRatio.SetFocus
    Exit Sub
  End If
  If lstFuel.Text = "" Then
    MsgBox "PLEASE SELECT A FUEL"
    lstFuel.SetFocus
    Exit Sub
  End If
  If txtCompRatio.Text = "" Then
    MsgBox "PLEASE SPECIFY A COMPRESSION RATIO"
    txtCompRatio.SetFocus
    Exit Sub
  End If
  If txtBore.Text = "" Then
    MsgBox "PLEASE SPECIFY A BORE"
    txtBore.SetFocus
```

```
   Exit Sub
End If
If txtStroke.Text = "" Then
   MsgBox "PLEASE SPECIFY A STROKE"
   txtStroke.SetFocus
   Exit Sub
End If
If txtRodLength.Text = "" Then
   MsgBox "PLEASE SPECIFY A CONNECTING ROD LENGTH"
   txtRodLength.SetFocus
   Exit Sub
End If
If txtSparkAdvance.Text = "" Then
   MsgBox "PLEASE SPECIFY A SPARK ADVANCE"
   txtSparkAdvance.SetFocus
   Exit Sub
End If
If txtCombustionDuration.Text = "" Then
   MsgBox "PLEASE SPECIFY A COMBUSTION DURATION"
   txtCombustionDuration.SetFocus
   Exit Sub
End If

If txtNumberCylinders.Text = "" Then
   MsgBox "PLEASE SPECIFY THE NUMBER OF CYLINDERS"
   txtNumberCylinders.SetFocus
   Exit Sub
End If


Bore = Val(txtBore.Text)
Stroke = Val(txtStroke.Text)
RodLength = Val(txtRodLength.Text)
StrokeRatio = RodLength / Stroke
SparkAdvance = Val(txtSparkAdvance.Text)
CombustionDuration = Val(txtCombustionDuration.Text)
NumberCylinders = Val(txtNumberCylinders.Text)
CompRatio = Val(txtCompRatio.Text)
EnrichmentEquivRatio = Val(txtEnrichmentEquivRatio.Text)
CombustorEquivRatio = Val(txtCombustorEquivRatio.Text)
frmEngineParameters.Hide
EngineDisplacement = 3.1416 * Stroke * Bore ^ 2 / 4 * NumberCylinders
AmbientPressure = Val(txtAmbientPressure.Text)
AmbientTemperature = Val(txtAmbientTemperature.Text)
   MC = Val(Mid$(lstFuel.Text, 18, 6))
   MH = Val(Mid$(lstFuel.Text, 24, 6))
   MO = Val(Mid$(lstFuel.Text, 30, 5))
```

```
    Hrp = Val(Mid$(lstFuel.Text, 35, 10))
    Hfg = Val(Mid$(lstFuel.Text, 45, 8))
    If optLiquid.Value = -1 Then
      Hrp = Hrp - Hfg
    End If
    FuelACP = Val(Mid$(lstFuel.Text, 53, 6))
    FuelBCP = Val(Mid$(lstFuel.Text, 59, 8))
End Sub


Sub Form_Activate ()
  tmpBore$ = txtBore.Text
  tmpStroke$ = txtStroke.Text
  tmpRodLength$ = txtRodLength.Text
  tmpSparkAdvance$ = txtSparkAdvance.Text
  tmpCombustionDuration$ = txtCombustionDuration.Text
  tmpNumberCylinders$ = txtNumberCylinders.Text
  tmpCompRatio$ = txtCompRatio.Text
  tmpAmbientPressure$ = txtAmbientPressure.Text
  tmpAmbientTemperature$ = txtAmbientTemperature.Text
  tmpEquivRatio$ = txtEquivRatio.Text
  tmpEnrichmentEquivRatio$ = txtEnrichmentEquivRatio.Text
  tmpCombustorEquivRatio$ = txtCombustorEquivRatio.Text
  tmpFuelSelection% = lstFuel.ListIndex

End Sub


Sub Form_Load ()
  Open "Fuels.txt" For Input As 1
    Do While Not EOF(1)
     Line Input #1, FuelStuff$
     frmEngineParameters.lstFuel.AddItem FuelStuff$
    Loop
  Close 1

  txtCompRatio.Text = "8.9"
  txtBore.Text = "8.90"
  txtStroke.Text = "7.60"
  txtRodLength.Text = "13.13"
  txtSparkAdvance.Text = "15"
  txtCombustionDuration.Text = "30"
  lstFuel.ListIndex = 5
  txtNumberCylinders.Text = "6"
  txtAmbientPressure.Text = "1.0"
  txtAmbientTemperature.Text = "243"
  txtEquivRatio.Text = "1.0"
  txtEnrichmentEquivRatio.Text = "1.0"
  txtCombustorEquivRatio.Text = "2.0"
  Bore = Val(txtBore.Text)
  Stroke = Val(txtStroke.Text)
  RodLength = Val(txtRodLength.Text)
  StrokeRatio = RodLength / Stroke
  SparkAdvance = Val(txtSparkAdvance.Text)
  CombustionDuration = Val(txtCombustionDuration.Text)
  NumberCylinders = Val(txtNumberCylinders.Text)
  CompRatio = Val(txtCompRatio.Text)
  EnrichmentEquivRatio = Val(txtEnrichmentEquivRatio.Text)
  CombustorEquivRatio = Val(txtCombustorEquivRatio.Text)
  frmEngineParameters.Hide
  EngineDisplacement = 3.1416 * Stroke * Bore ^ 2 / 4 * NumberCylinders
  AmbientPressure = Val(txtAmbientPressure.Text)
  AmbientTemperature = Val(txtAmbientTemperature.Text)
    MC = Val(Mid$(lstFuel.Text, 18, 6))
    MH = Val(Mid$(lstFuel.Text, 24, 6))
    MO = Val(Mid$(lstFuel.Text, 30, 5))
    Hrp = Val(Mid$(lstFuel.Text, 35, 10))
    Hfg = Val(Mid$(lstFuel.Text, 45, 8))
    If optLiquid.Value = -1 Then
      Hrp = Hrp - Hfg
    End If
    FuelACP = Val(Mid$(lstFuel.Text, 53, 6))
    FuelBCP = Val(Mid$(lstFuel.Text, 59, 8))

End Sub


Sub optConstantAF_Click ()
  lblEquivRatio.Enabled = -1
  txtEquivRatio.Enabled = -1
  txtEquivRatio.SetFocus
End Sub


Sub optEnrichment_Click ()
  lblCombustorEquivRatio.Enabled = 0
  txtCombustorEquivRatio.Enabled = 0
  lblEnrichmentEquivRatio.Enabled = -1
  txtEnrichmentEquivRatio.Enabled = -1
  frmAdFlameTemperature.lblCombustorProductsTemp.Visible = False
  frmAdFlameTemperature.lblCombTempUnits.Visible = False
  frmAdFlameTemperature.txtCombustorTemp.Visible = False
  txtEnrichmentEquivRatio.SetFocus
End Sub
```

```
Sub optRichCombustor_Click ()
   lblEnrichmentEquivRatio.Enabled = 0
   txtEnrichmentEquivRatio.Enabled = 0
   lblCombustorEquivRatio.Enabled = -1
   txtCombustorEquivRatio.Enabled = -1
   frmAdFlameTemperature.lblCombustorProductsTemp.Visible = True
   frmAdFlameTemperature.lblCombTempUnits.Visible = True
   frmAdFlameTemperature.txtCombustorTemp.Visible = True
   txtCombustorEquivRatio.SetFocus
End Sub


Sub optStoichiometric_Click ()
   lblEquivRatio.Enabled = 0
   txtEquivRatio.Enabled = 0
   txtEquivRatio.Text = "1.0"
End Sub


Sub spnCombDuration_SpinDown ()
   txtCombustionDuration.Text = Str$(Val(txtCombustionDuration.Text) - 1)
End Sub


Sub spnCombDuration_SpinUp ()
   txtCombustionDuration.Text = Str$(Val(txtCombustionDuration.Text) + 1)
End Sub


Sub spnSparkAdvance_SpinDown ()
   txtSparkAdvance.Text = Str$(Val(txtSparkAdvance.Text) - 1)
End Sub


Sub spnSparkAdvance_SpinUp ()
   txtSparkAdvance.Text = Str$(Val(txtSparkAdvance.Text) + 1)
End Sub


Sub Blowdown (TempProducts, PressProducts, VolumeProducts, SProducts)
   PressExpand = AmbientPressure
   TempExpand = TempProducts
   Volume5 = VolumeState1
```

```
blow1:
  If TempExpand > 1600 Then
        Call Equilibrium(TempExpand, PressExpand, 0)
  End If
  SDelta = SProducts - EntropyProducts(TempExpand, PressExpand)
'
' Decrease TempExpand until SDelta becomes positive
'
            If SDelta < 0 Then
                  TempExpand = TempExpand - 100
                  GoTo blow1
            End If
'
' The Isentropic Expansion Temperature has just been passed when SDelta became positive
' Begin Bisection Method
'
            TempExpLast = TempExpand
            TempExpand = TempExpand + 50
            k% = 1
blow2:
        If TempExpand > 1600 Then
          Call Equilibrium(TempExpand, PressExpand, 0)
        End If
        SDelta = SProducts - EntropyProducts(TempExpand, PressExpand)
        If SDelta < 0 Then
          TempExpLast = TempExpand
          TempExpand = TempExpand - 50! / 2 ^ k%
        If Abs(TempExpand - TempExpLast) < 1 GoTo blow3     'Convergance
          k% = k% + 1
          GoTo blow2
        Else
          TempExpLast = TempExpand
          TempExpand = TempExpand + 50! / 2 ^ k%
        If Abs(TempExpand - TempExpLast) < 1 GoTo blow3     'Convergance
          k% = k% + 1
          GoTo blow2
        End If
blow3:
        TempState5 = TempExpand
        MolesExhaustProduct = MolesProduct
        PressState5 = PressExpand
        VolumeBlowdown = MolesProduct * TempExpand * 8.314 / PressExpand / 101
        BlowdownFactor = VolumeState5 / VolumeBlowdown
        Residual = VolumeState3 / VolumeBlowdown
        For i = 1 To 10
          NumberMolesExhaust(i) = (1 - Residual) * NumberMoles(i)
        Next
        MolesExhaustProduct = (1 - Residual) * MolesExhaustProduct
        MolesProduct = 0
        For i = 1 To 10
           NumberMoles(i) = BlowdownFactor * NumberMoles(i)
           MolesProduct = MolesProduct + NumberMoles(i)
        Next
        BlowdownData$ = " 5 " + "     " + Format$(TempExpand, "####") + "     " +
Format$(PressExpand, "##.00")
        For A% = 1 To 10
'
'for mole fraction of product, comment out the next line
'
'     BlowdownData$ = BlowdownData$ + "     " + FORMAT$(NumberMoles(A%), "00.000")
'
'for mole numbers of product, comment out the next line
'
        BlowdownData$ = BlowdownData$ + "     " + Format$((NumberMoles(A%) /
MolesProduct), "0.000")
'
'
        Next
End Sub

Sub cmdCalcAFT_Click ()
    VolumetricEfficiency = Val(txtVolumetricEfficiency.Text)
    EngineSpeed = Val(txtEngineSpeed.Text)
    PressManifold = Val(txtPressManifold.Text)
    Call FourStroke(VolumetricEfficiency, EngineSpeed, PressManifold)
End Sub


Sub cmdCalcSinglePoint_Click ()
    fraInputs.Visible = False
    VolumetricEfficiency = Val(txtVolumetricEfficiency.Text)
    EngineSpeed = Val(txtEngineSpeed.Text)
    PressManifold = Val(txtPressManifold.Text)
    Call FourStroke(VolumetricEfficiency, EngineSpeed, PressManifold)
    mnuViewSinglePoint.Enabled = True
    mnuViewCombustor.Enabled = True
    mnuView.Enabled = True
End Sub


Sub cmdCalculatePerfMap_Click ()
  ReDim PerformanceMapData$(100, 21)
  gagPercentComplete.Visible = True
  On Error GoTo usercancel2
```

```
        VolumetricEfficiency = .58
        EngineSpeed = 800
        PressManifold = .55
        Call FourStroke(VolumetricEfficiency, EngineSpeed, PressManifold)
        PerformanceMapData$(EngineSpeed / 100, PressManifold * 20) =
Format(BrakeHorsepower, "000.0") + " " + Format(FuelFlowRate, "00.00") + " " +
Format(VolumetricEfficiency, "0.0000")
        For spec1% = 1 To 10
          PerformanceMapData$(EngineSpeed / 100, PressManifold * 20) =
PerformanceMapData$(EngineSpeed / 100, PressManifold * 20) + " " +
Format(NumberMolesExhaust(spec1%) / MolesExhaustProduct, "0.0000")
        Next

  NumberPoints = 0
  PointNumber = 0
  CMDialog1.Filter = "Vol. Eff. Files *.eff|*.eff|Text Files *.txt|*.txt|All Files *.*|*.*"
  CMDialog1.FilterIndex = 1
  CMDialog1.DialogTitle = "OPEN VOLUMETRIC EFFICIENCY FILE"
  CMDialog1.CancelError = True
  CMDialog1.Action = 1
  If CMDialog1.Filename = "" Then Exit Sub
  frmAdFlameTemperature.Refresh
  Open CMDialog1.Filename For Input As 10

  Do While Not EOF(10)
        Line Input #10, EngineData$
        NumberPoints = NumberPoints + 1
  Loop
  Close 10
  Open CMDialog1.Filename For Input As 10
  Do While Not EOF(10)
        Line Input #10, EngineData$
        PointNumber = PointNumber + 1
        VolumetricEfficiency = Val(Mid$(EngineData$, 19, 6))
        EngineSpeed = Val(Mid$(EngineData$, 5, 4))
        PressManifold = Val(Mid$(EngineData$, 14, 3)) / 100
        Call FourStroke(VolumetricEfficiency, EngineSpeed, PressManifold)
        PerformanceMapData$(EngineSpeed / 100, PressManifold * 20) =
Format(BrakeHorsepower, "000.0") + " " + Format(FuelFlowRate, "00.00") + " " +
Format(VolumetricEfficiency, "0.0000")
        For spec1% = 1 To 10
          PerformanceMapData$(EngineSpeed / 100, PressManifold * 20) =
PerformanceMapData$(EngineSpeed / 100, PressManifold * 20) + " " +
Format(NumberMolesExhaust(spec1%) / MolesExhaustProduct, "0.0000")
        Next
        gagPercentComplete.Value = 100 * PointNumber / NumberPoints
  Loop
```

```
   Close 10
   mnuView.Enabled = True
   mnuViewEngineOutput.Enabled = True
   cmdDrivingCycle.Enabled = True
   mnuFileSave.Enabled = True
   mnuSavePerfData.Enabled = True
   gagPercentComplete.Visible = False

   If frmEngineParameters.optRichCombustor.Value = True Then
          mnuViewCombustor.Enabled = True
   End If


usercancel2:
   Exit Sub
   Resume

End Sub
```

Sub cmdCancel_Click ()

```
   txtOutputFile.Text = ""
   fraOutputFile.Visible = False
   cmdSaveData.SetFocus
End Sub
```

Sub cmdCancelSinglePoint_Click ()

```
   fraInputs.Visible = 0
End Sub
```

Sub cmdClear_Click ()

```
   txtPressManifold.Text = ""
   frmEngineParameters.txtCompRatio.Text = ""
End Sub
```

Sub cmdCombustorOutputReturn_Click ()

```
   TotalItems = lstCombustorOutput.ListCount
   For i = 1 To TotalItems
          lstCombustorOutput.RemoveItem 0
   Next i
   fraCombustorOutput.Visible = False

End Sub
```

```
Sub cmdDrivingCycle_Click ()
  ReDim VehicleSpeed(505), RequiredBrakePower(505), CycleEngineSpeed(505),
CycleFuelFlowrate(505), CycleBrakePower(505)
  gagPercentComplete.Visible = True
  AxleRatio = Val(frmChassisParameters.txtAxleRatio.Text)
  IdleHorsepower = Val(Left$(PerformanceMapData$(8, 11), 4))
  IdleFuelRate = Val(Mid$(PerformanceMapData$(8, 11), 6, 5))

  Open "fuds.txt" For Input As 1
        Line Input #1, CycleStep$
        VehicleSpeed(0) = Val(Right$(CycleStep$, 6))
        CycleEngineSpeed(i) = 800
        RequiredBrakePower(0) = IdleHorsepower
        For j = 1 To 505
         Line Input #1, CycleStep$
         i = CInt(Left$(CycleStep$, 4))
         VehicleSpeed(i) = Val(Right$(CycleStep$, 6))
         If VehicleSpeed(i) = 0 Then
          CycleEngineSpeed(i) = 800
          CycleFuelFlowrate(i) = IdleFuelRate
          RequiredBrakePower(i) = IdleHorsepower
          GoTo Idle
         End If
         VehicleAcceleration = (VehicleSpeed(j) - VehicleSpeed(j - 1)) * 88 / 60
         If VehicleAcceleration < 0 Then
          CycleEngineSpeed(i) = 800
          CycleFuelFlowrate(i) = IdleFuelRate
          RequiredBrakePower(i) = IdleHorsepower
          GoTo Idle
         End If
         If VehicleSpeed(i) < 15 Then
          TransRatio = Val(frmChassisParameters.txtTransmissionRatio1.Text)
         ElseIf VehicleSpeed(i) < 25 Then
          TransRatio = Val(frmChassisParameters.txtTransmissionRatio2.Text)
         ElseIf VehicleSpeed(i) < 40 Then
          TransRatio = Val( frmChassisParameters.txtTransmissionRatio3.Text)
         Else
          TransRatio = Val(frmChassisParameters.txtTransmissionRatio4.Text)
         End If


         CycleEngineSpeed(i) = (VehicleSpeed(i) * 5280 * 12 / 60 / 3.1416 /
Val(frmChassisParameters.txtTireDiameter.Text)) * AxleRatio * TransRatio
         RequiredBrakePower(i) = ((VehicleMass * VehicleAcceleration * VehicleSpeed(i) *
88 / 60 / 550) + .1263 * VehicleSpeed(i) + .00008448 * VehicleSpeed(i) ^ 3) / .7

' Determine the operating point from the performance map data
```

```
        LowerRPMPoint = Int(CycleEngineSpeed(i) / 100)
        UpperRPMPoint = LowerRPMPoint + 1
        For k = 13 To 20
         If RequiredBrakePower(i) < Val(Left$(PerformanceMapData$(LowerRPMPoint, k),
4)) Then
                LowerRPMUpperHP = k
                LowerRPMLowerHP = k - 1
                GoTo UpperRPM
         End If

        Next

UpperRPM:

        For m = 13 To 20
         If RequiredBrakePower(i) < Val(Left$(PerformanceMapData$(UpperRPMPoint, m),
5)) Then
                UpperRPMUpperHP = m
                UpperRPMLowerHP = m - 1
                GoTo FoundPoint
         End If

        Next

FoundPoint:

        XRPM = CycleEngineSpeed(i) / 100 - LowerRPMPoint
        FuelRateLL = Val(Mid$(PerformanceMapData$(LowerRPMPoint,
LowerRPMLowerHP), 6, 5))
        FuelRateLU = Val(Mid$(PerformanceMapData$(LowerRPMPoint,
LowerRPMUpperHP), 6, 5))
        FuelRateUL = Val(Mid$(PerformanceMapData$(UpperRPMPoint,
UpperRPMLowerHP), 6, 5))
        FuelRateUU = Val(Mid$(PerformanceMapData$(UpperRPMPoint,
UpperRPMUpperHP), 6, 5))
        PowerLL = Val(Left$(PerformanceMapData$(LowerRPMPoint,
LowerRPMLowerHP), 5))
        PowerLU = Val(Left$(PerformanceMapData$(LowerRPMPoint,
LowerRPMUpperHP), 5))
        PowerUL = Val(Left$(PerformanceMapData$(UpperRPMPoint,
UpperRPMLowerHP), 5))
        PowerUU = Val(Left$(PerformanceMapData$(UpperRPMPoint,
UpperRPMUpperHP), 5))
        PowerUpper = (PowerUU - PowerLU) * XRPM + PowerLU
        PowerLower = (PowerUL - PowerLL) * XRPM + PowerLL
        FuelRateUpper = (FuelRateUU - FuelRateLU) * XRPM + FuelRateLU
```

```
          FuelRateLower = (FuelRateUL - FuelRateLL) * XRPM + FuelRateLU
          XPower = (RequiredBrakePower(i) - PowerLower) / (PowerUpper - PowerLower)
          CycleFuelFlowrate(i) = (1 - XPower) * FuelRateLower + XPower * FuelRateUpper


Idle:   'This point is at idle conditions
NextPoint:
        gagPercentComplete.Value = j / 5.05
        Next
        mnuView.Enabled = True
        mnuViewDrivingCycle.Enabled = True
        mnuFileSave.Enabled = True
        mnuSaveDrivCycData.Enabled = True
        gagPercentComplete.Visible = False


        Close 1

End Sub


Sub cmdDrivingCycleOK_Click ()
   fraDrivingCycle.Visible = False
End Sub


Sub cmdExhaustCompReturn_Click ()
   TotalItems = lstExhaustComposition.ListCount
   For i = 1 To TotalItems
        lstExhaustComposition.RemoveItem 0
   Next i
   fraExhaustComposition.Visible = False
End Sub


Sub cmdExit_Click ()
   End
End Sub


Sub cmdOtto_Click ()
   Open "Otto.txt" For Output As 1
   For i = 1 To 300
        Print #1, Format$(i, "000"), Format$(TempVolume(i) / VolumeState1, "0.000"),
Format$(TempPressure(i), "###.00")
   Next
   Close 1
```

```
End Sub


Sub cmdPerformanceDataOK_Click ()
   TotalItems = lstPerformanceData.ListCount
   For i = 1 To TotalItems
          lstPerformanceData.RemoveItem 0
   Next i
   fraPerformanceData.Visible = False


End Sub


Sub cmdPerformanceMapOK_Click ()
   fraPerformanceMap.Visible = False
End Sub


Sub cmdPrintData_Click ()
On Local Error GoTo error1
            lblFuelName.Caption = Mid$(frmEngineParameters.lstFuel.Text, 10, 5) + " with " +
frmEngineParameters.txtEquivRatio.Text + " theoretical air."
            Printer.Print lblFuelName.Caption
            Printer.Print ""
            Printer.Print "State    Temp    Press   CO2     CO      O2      O       NO      N2
H       H2      OH      H2O"
            Printer.Print ""
                   Printer.Print IntakeData$
                   Printer.Print IsentropicCompressionData$
                   Printer.Print AdiabaticFlameData$
                   Printer.Print " init    " + Format$(TempInitial, "####") + "   " +
Format$(PressInitial, "##.##")
                   Printer.Print " final  " + Format$(TempFinal, "####") + "   " +
Format$(PressFinal, "##.##")
                   Printer.Print IsentropicExpansionData$
                   Printer.Print BlowdownData$
                   Printer.Print ExhaustData$
                   Printer.Print "Residual = " + Format$(Residual, "0.000")
                   Printer.Print "IMEP = " + Format$(IMEP, "##.00")
   Printer.NewPage
   Printer.EndDoc
Exit Sub
error1: MsgBox "check the printer"
Resume
End Sub
```

```
Sub cmdReturn_Click ()
   TotalItems = lstFlameTemperature.ListCount
   For i = 1 To TotalItems
         lstFlameTemperature.RemoveItem 0
   Next i
   fraOutput.Visible = False
End Sub


Sub cmdSaveData_Click ()
   lblFileExtension.Caption = ".WOT"
   txtOutputFile.Text = Left$(frmEngineParameters.lstFuel.Text, 8)
   fraOutputFile.Visible = True
   txtOutputFile.SetFocus
   cmdWrite.Default = -1
End Sub


Sub cmdViewData_Click ()
   lblFuelName.Caption = Mid$(frmEngineParameters.lstFuel.Text, 10, 5) + " with " +
frmEngineParameters.txtEquivRatio.Text + " theoretical air."
   lstFlameTemperature.AddItem IntakeData$
   lstFlameTemperature.AddItem IsentropicCompressionData$
   lstFlameTemperature.AddItem " init    " + Format$(TempInitial, "####") + "    " +
Format$(PressInitial, "##.##")
   lstFlameTemperature.AddItem AdiabaticFlameData$
   lstFlameTemperature.AddItem " final  " + Format$(TempFinal, "####") + "    " +
Format$(PressFinal, "##.##")
   lstFlameTemperature.AddItem IsentropicExpansionData$
   lstFlameTemperature.AddItem BlowdownData$
   lstFlameTemperature.AddItem ExhaustData$
   lstFlameTemperature.AddItem "Residual = " + Format$(Residual, "0.000")
   lstFlameTemperature.AddItem "IMEP = " + Format$(IMEP, "##.00") + " atm"
   lstFlameTemperature.AddItem "POWER = " + Format$(Horsepower, "##.00") + " hp"
   fraInputs.Visible = False
   fraOutput.Visible = True
   frmEngineParameters.fraDimensions.Visible = False
   cmdReturn.SetFocus
End Sub


Sub cmdWrite_Click ()
On Local Error GoTo error2
   If txtOutputFile.Text = "" Then
         MsgBox "PLEASE SPECIFY AN OUTPUT FILENAME"
```

```
        GoTo 5000
    End If
        lblFuelName.Caption = Mid$(frmEngineParameters.lstFuel.Text, 10, 5) + " with " +
frmEngineParameters.txtEquivRatio.Text + " theoretical air."
        OutputFile$ = Left$(txtOutputFile.Text, 8) + ".WOT"
        Open OutputFile$ For Output As 1
            Print #1, lblFuelName.Caption
            Print #1, ""
            Print #1, "State    Temp    Press   CO2     CO      O2      O       NO      N2
H       H2      OH      H2O"
            Print #1, ""
                    Print #1, IntakeData$
                    Print #1, IsentropicCompressionData$
                    Print #1, AdiabaticFlameData$
                    Print #1, IsentropicExpansionData$
                    Print #1, BlowdownData$
                    Print #1, ExhaustData$
                    Print #1, "Residual = " + Format$(Residual, "0.000")
                    Print #1, "IMEP = " + Format$(IMEP, "##.00")
        Close 1
    fraOutputFile.Visible = False
Exit Sub
error2:
    Select Case Err
    Case 68, 71
        MsgBox "Make sure the disk is available for writing!"
    Case 59
        response% = MsgBox("Existing file will be overwritten!", 257, "WARNING")
        If response% = 2 Then Exit Sub
    Case Else
        MsgBox "File in use by another application"
        txtOutputFile.SetFocus
        Exit Sub
    End Select
Resume
5000 End Sub
```

Sub Combustion (UReactants, MolesReactant)

```
    TempProducts = 1600 'Initial Value for iteration


'   Increase TempProducts until UDelta becomes negative
'
60  A = PressState2 * TempProducts / (MolesReactant * TempState2)
    Call Equilibrium(TempProducts, 0, A)
    UDelta = UReactants - (HProducts(TempProducts) - 8.314 * MolesProduct * TempProducts)
```

```
   If UDelta > 0 Then
          TempProducts = TempProducts + 100
          GoTo 60
   End If
'
'  The Adiabatic Flame Temperature has just been passed when UDelta became negative
'  Begin Bisection Method
'
   TempProdLast = TempProducts
   TempProducts = TempProducts - 50
   k% = 1
70  A = PressState2 * TempProducts / (MolesReactant * TempState2)
   Call Equilibrium(TempProducts, 0, A)
   UDelta = UReactants - (HProducts(TempProducts) - 8.314 * MolesProduct * TempProducts)
   If UDelta > 0 Then
          TempProdLast = TempProducts
          TempProducts = TempProducts + 50! / 2 ^ k%
   If Abs(TempProducts - TempProdLast) < 1 GoTo 80     'Convergance
          k% = k% + 1
          GoTo 70
   Else
          TempProdLast = TempProducts
          TempProducts = TempProducts - 50! / 2 ^ k%
   If Abs(TempProducts - TempProdLast) < 1 GoTo 80     'Convergance
             k% = k% + 1
             GoTo 70
   End If
'PressState3 =PressState2 * MolesProduct * TempProducts / (MolesReactant * TempState2)
80  PressState3 = 8.314 * MolesProduct * TempProducts / VolumeState3 / 101
   TempState3 = TempProducts
   AdiabaticFlameData$ = " 3 " + "     " + Format$(TempState3, "####") + "   " +
Format$(PressState3, "###.##")
   For A1% = 1 To 10
'
'for mole fraction of product, comment out the next line
'
'  AdiabaticFlameData$ = AdiabaticFlameData$ + "    " + FORMAT$(NumberMoles(A%),
"00.000")
'
'for mole numbers of product, comment out the next line
'
   AdiabaticFlameData$ = AdiabaticFlameData$ + "    " + Format$((NumberMoles(A1%) /
MolesProduct), "0.000")
'
'
          Next
End Sub
```

```
Sub Command2_Click ()
   fraPerformanceMap.Visible = False
End Sub


Sub Compression (VolumeReactants)
   TempState2 = TempState1
Comp1:
   SDelta = EntropyReactant(TempState2, VolumeReactants) - EntropyReactant(TempState1,
VolumeIntakeCloses)
'
'  Increase TempState2 until SDelta becomes positive
'
   If SDelta < 0 Then
          TempState2 = TempState2 + 100
          GoTo Comp1
   End If
'
'  The Isentropic Compression Temperature has just been passed when SDelta became positive
'  Begin Bisection Method
'
   TempState2Last = TempState2
   TempState2 = TempState2 - 50
   k% = 1
comp3:
   SDelta = EntropyReactant(TempState2, VolumeReactants) - EntropyReactant(TempState1,
VolumeIntakeCloses)
   If SDelta < 0 Then
          TempState2Last = TempState2
          TempState2 = TempState2 + 50! / 2 ^ k%
   If Abs(TempState2 - TempState2Last) < 1 GoTo comp2     'Convergance
          k% = k% + 1
          GoTo comp3
   Else
          TempState2Last = TempState2
          TempState2 = TempState2 - 50! / 2 ^ k%
   If Abs(TempState2 - TempState2Last) < 1 GoTo comp2     'Convergance
          k% = k% + 1
          GoTo comp3
   End If
comp2:
   PressState2 = PressState1 * VolumeIntakeCloses / VolumeReactants * TempState2 /
TempState1
   If VolumeReactants = VolumeState2 Then   'Piston at TDC
   IsentropicCompressionData$ = " 2  " + "     " + Format$(TempState2, "####") + "   " +
Format$(PressState2, "##.##")
         For A% = 1 To 10
'
'for mole fraction of product, comment out the next line
'
'     IsentropicCompressionData$ = IsentropicCompressionData$ + "    " +
FORMAT$(NumberMoles(A%), "00.000")
'
'for mole numbers of product, comment out the next line
'
         IsentropicCompressionData$ = IsentropicCompressionData$ + "   " +
Format$((NumberMoles(A%) / MolesState1), "0.000")
'
'
         Next
   End If
End Sub


Function EntropyProducts (Temperature, Pressure)
    TempEntropy = 0
    If Temperature < 1600 Then
           For j = 1 To 10
             TempEntropy = TempEntropy + NumberMoles(j) * (BL(j) * Log(Temperature) -
CL(j) / Temperature + DL(j) - 8.314 * Log(Pressure))
             If NumberMoles(j) > 0 Then
               TempEntropy = TempEntropy - 8.314 * NumberMoles(j) * Log(NumberMoles(j) /
MolesProduct)
             End If
           Next
    Else
           For j = 1 To 10
             TempEntropy = TempEntropy + NumberMoles(j) * (BH(j) * Log(Temperature) -
CH(j) / Temperature + DH(j) - 8.314 * Log(Pressure))
             If NumberMoles(j) > 0 Then
               TempEntropy = TempEntropy - 8.314 * NumberMoles(j) * Log(NumberMoles(j) /
MolesProduct)
             End If
           Next
    End If
    EntropyProducts = TempEntropy
End Function


Function EntropyReactant (Temperature, Volume)
   FuelACP = Val(Mid$(frmEngineParameters.lstFuel.Text, 53, 6))
```

```
 FuelBCP = Val(Mid$(frmEngineParameters.lstFuel.Text, 59, 8))
 EntropyFuel = VolumetricEfficiency * (1 - Residual) * ((FuelACP - 8.314) *
Log(Temperature) + FuelBCP * Temperature + 8.314 * Log(Volume))
 EntropyAir_Residual = 0
 For j = 1 To 10
     EntropyAir_Residual = EntropyAir_Residual + NumberMoles(j) * ((ACP(j) - 8.314) *
Log(Temperature) + BCP(j) * Temperature + 8.314 * Log(Volume))
 Next
 EntropyReactant = EntropyAir_Residual
 If frmEngineParameters.optRichCombustor.Value = 0 Then
     EntropyReactant = EntropyFuel + EntropyAir_Residual
 End If
End Function


Sub Equilibrium (Temperature, Pressure, A)
 Static KReaction(6)
 For React% = 1 To 6
     KReaction(React%) = Exp(KA(React%) / Temperature + (KB(React%) + KC(React%)
/ Temperature) * Log(Temperature) + KD(React%))
 Next
     If Pressure = 0 GoTo calcW
calcA:
 A = Pressure / MolesProduct
calcW:
 W1 = KReaction(1) * Sqr(A)
 W2 = KReaction(2) * A
 W3 = KReaction(3)
 W4 = KReaction(4) * A
 W5 = KReaction(5) * A
 W6 = KReaction(6) * Sqr(A)
 N7max = (Sqr(1 + 8 * AbundancyHydrogen * W4) - 1) / (4 * W4)
 N7min = 0
iterate:
     NumberMoles(7) = (N7max + N7min) / 2!
     NumberMoles(9) = (AbundancyHydrogen - NumberMoles(7) * (1 + 2 * W4 *
NumberMoles(7))) / (1 + 2 * W4 * W6 * NumberMoles(7) * Sqr(W2) / W5)
     NumberMoles(8) = W4 * NumberMoles(7) ^ 2
     NumberMoles(10) = NumberMoles(9) * NumberMoles(7) * W4 * W6 * Sqr(W2) / W5
     NumberMoles(4) = NumberMoles(9) / (W5 * NumberMoles(7))
     NumberMoles(3) = W2 * NumberMoles(4) ^ 2
     NumberMoles(1) = AbundancyCarbon / (1 + 1 / (W1 * Sqr(NumberMoles(3))))
     NumberMoles(2) = AbundancyCarbon - NumberMoles(1)
     NumberMoles(5) = W3 * NumberMoles(3) * (Sqr(1 + 8 * AbundancyNitrogen / (W3 *
NumberMoles(3))) - 1) / 4!
     NumberMoles(6) = (AbundancyNitrogen - NumberMoles(5)) / 2
```

```
        AOP = 2 * (NumberMoles(1) + NumberMoles(3)) + NumberMoles(2) +
NumberMoles(4) + NumberMoles(5) + NumberMoles(9) + NumberMoles(10)
        If Abs(AOP - AbundancyOxygen) / AbundancyOxygen < .001 Then
          GoTo converge1
        Else
          If AbundancyOxygen > AOP Then
                N7max = NumberMoles(7)
                GoTo iterate
          Else
                N7min = NumberMoles(7)
                GoTo iterate
          End If
        End If

converge1:
        If Pressure = 0 GoTo done1
        MolesProductPrime = 0
        For spec% = 1 To 10
          MolesProductPrime = MolesProductPrime + NumberMoles(spec%)
        Next
        If Abs(MolesProductPrime - MolesProduct) / MolesProduct > .001 Then
          MolesProduct = MolesProductPrime
          GoTo calcA
        End If

done1:
        MolesProduct = 0
        For spec% = 1 To 10
          MolesProduct = MolesProduct + NumberMoles(spec%)
        Next
End Sub


Sub Exhaust ()
        MolesProduct = 0
        For i = 1 To 10
          NumberMoles(i) = NumberMoles(i) / CompRatio
          MolesProduct = MolesProduct + NumberMoles(i)
        Next
        ExhaustData$ = " 6 " + "     " + Format$(TempState6, "####") + "    " +
Format$(PressState6, "##.00")
        For A% = 1 To 10
'
'for mole fraction of product, comment out the next line
'
'     ExhaustData$ = ExhaustData$ + "     " + FORMAT$(NumberMoles(A%), "00.000")
'
```

```
'for mole numbers of product, comment out the next line
'
        ExhaustData$ = ExhaustData$ + "    " + Format$((NumberMoles(A%) / MolesProduct),
"0.000")
'
'
        Next

End Sub


Sub ExpandResidual ()
  PressExpand = PressManifold
  TempExpand = TempState6
  Volume5 = VolumeState1
  SProducts = EntropyProducts(TempState6, AmbientPressure)
residual1:
  If TempExpand > 1600 Then
       Call Equilibrium(TempExpand, PressExpand, 0)
  End If
  SDelta = SProducts - EntropyProducts(TempExpand, PressExpand)
'
'  Decrease TempExpand until SDelta becomes positive
'
          If SDelta < 0 Then
               TempExpand = TempExpand - 16
               GoTo residual1
          End If
'
'  The Isentropic Expansion Temperature has just been passed when SDelta became positive
'  Begin Bisection Method
'
          TempExpLast = TempExpand
          TempExpand = TempExpand + 8
          k% = 1
residual2:
        If TempExpand > 1600 Then
          Call Equilibrium(TempExpand, PressExpand, 0)
        End If
        SDelta = SProducts - EntropyProducts(TempExpand, PressExpand)
        If SDelta < 0 Then
          TempExpLast = TempExpand
          TempExpand = TempExpand - 8! / 2 ^ k%
        If Abs(TempExpand - TempExpLast) < 1 GoTo residual3     'Convergance
          k% = k% + 1
          GoTo residual2
        Else
```

```
          TempExpLast = TempExpand
          TempExpand = TempExpand + 8! / 2 ^ k%
        If Abs(TempExpand - TempExpLast) < 1 GoTo residual3      'Convergance
          k% = k% + 1
          GoTo residual2
        End If
residual3:
        TempState6Prime = TempExpand
        VolumeState6Prime = MolesProduct * TempExpand * 8.314 / PressExpand / 101
End Sub


Sub Expansion (TempProducts, PressProducts, SProducts, Ratio, MolesPassed)
        TempExpand = TempProducts
        VolumeState4 = VolumeState1
61  If TempExpand > 1600 Then
        A = 8.314 * TempExpand / (CompRatio * VolumeState3)
        Call Equilibrium(TempExpand, 0, A)
  End If
  PressExpand = PressProducts / Ratio * TempExpand / TempProducts * MolesProduct /
MolesPassed
  SDelta = SProducts - EntropyProducts(TempExpand, PressExpand)
'
'  Increase TempExpand until SDelta becomes positive
'
          If SDelta < 0 Then
                TempExpand = TempExpand - 100
                GoTo 61
          End If
'
'  The Isentropic Expansion Temperature has just been passed when SDelta became positive
'  Begin Bisection Method
'
          TempExpLast = TempExpand
          TempExpand = TempExpand + 50
          k% = 1
71     If TempExpand > 1600 Then
        A = 8.314 * TempExpand / (CompRatio * VolumeState3)
          Call Equilibrium(TempExpand, 0, A)
        End If
        PressExpand = PressProducts / Ratio * TempExpand / TempProducts * MolesProduct /
MolesPassed
        SDelta = SProducts - EntropyProducts(TempExpand, PressExpand)
        If SDelta < 0 Then
          TempExpLast = TempExpand
          TempExpand = TempExpand - 50! / 2 ^ k%
          If Abs(TempExpand - TempExpLast) < 1 GoTo 81      'Convergance
```

```
                k% = k% + 1
                GoTo 71
          Else
                TempExpLast = TempExpand
                TempExpand = TempExpand + 50! / 2 ^ k%
                If Abs(TempExpand - TempExpLast) < 1 GoTo 81     'Convergance
                k% = k% + 1
                GoTo 71
          End If
81    PressExpand = PressProducts / Ratio * TempExpand / TempProducts * MolesProduct /
MolesPassed
        TempState4 = TempExpand
        PressState4 = PressExpand
        IsentropicExpansionData$ = " 4 " + "     " + Format$(TempState4, "####") + "    " +
Format$(PressState4, "##.00")
        For A2% = 1 To 10
'
'for mole fraction of product, comment out the next line
'
'      IsentropicExpansionData$ = IsentropicExpansionData$ + "     " +
FORMAT$(NumberMoles(A%), "00.000")
'
'for mole numbers of product, comment out the next line
'
        IsentropicExpansionData$ = IsentropicExpansionData$ + "    " +
Format$((NumberMoles(A2%) / MolesProduct), "0.000")
'
'

        Next
End Sub


Sub Form_Load ()
'  ON LOCAL ERROR GOTO error3
   ReDim NumberCombustorMoles(10), NumberMoles(10), ACP(10), BCP(10), Hzero(10),
SpeciesName$(10), AH(10), BH(10), CH(10), DH(10), AL(10), BL(10), CL(10), DL(10),
KA(6), KB(6), KC(6), KD(6)
   frmAdFlameTemperature.Caption = "Thermodynamic Engine Model with Rich Combustor"
   Open "products.txt" For Input As 1
   For Species% = 1 To 10
         Line Input #1, ProductStuff$
         SpeciesName$(Species%) = Left$(ProductStuff$, 6)
         Hzero(Species%) = Val(Mid$(ProductStuff$, 7, 8))
         ACP(Species%) = Val(Mid$(ProductStuff$, 15, 6))
         BCP(Species%) = Val(Mid$(ProductStuff$, 21, 8))
         AL(Species%) = Val(Mid$(ProductStuff$, 29, 10))
         BL(Species%) = Val(Mid$(ProductStuff$, 39, 7))
```

```
       CL(Species%) = Val(Mid$(ProductStuff$, 46, 10))
       DL(Species%) = Val(Mid$(ProductStuff$, 56, 8))
       AH(Species%) = Val(Mid$(ProductStuff$, 64, 10))
       BH(Species%) = Val(Mid$(ProductStuff$, 74, 7))
       CH(Species%) = Val(Mid$(ProductStuff$, 81, 10))
       DH(Species%) = Val(Mid$(ProductStuff$, 91, 8))
Next
Close 1
Open "table_c2.txt" For Input As 1
      For React% = 1 To 6
        Line Input #1, ReactionData$
        KA(React%) = Val(Mid$(ReactionData$, 1, 10))
        KB(React%) = Val(Mid$(ReactionData$, 11, 9))
        KC(React%) = Val(Mid$(ReactionData$, 20, 9))
        KD(React%) = Val(Mid$(ReactionData$, 29, 8))
      Next
Close 1
fraCombustorOutput.Visible = False
txtPressManifold.Text = "1"
fraOutput.Visible = False
fraOutputFile.Visible = False
fraOutput.Left = 3320
fraOutput.Top = 120
fraOutput.Width = 4215
fraOutput.Height = 4335
fraOutputFile.Left = 13
fraOutputFile.Top = 5
fraOutputFile.Width = 18
fraOutputFile.Height = 8
txtVolumetricEfficiency.Text = "1"
txtEngineSpeed.Text = "3000"
txtCombustorTemp = "243"
fraInputs.Visible = False
mnuViewSinglePoint.Enabled = False
mnuViewCombustor.Enabled = False
mnuViewEngineOutput.Enabled = False
fraExhaustComposition.Visible = False
fraPerformanceData.Visible = False
mnuView.Enabled = False
fraPerformanceMap.Visible = False
cmdSaveData.Visible = False
cmdOtto.Visible = False
VehicleMass = Val(frmChassisParameters.txtVehicleWeight.Text) / 32.2
fraDrivingCycle.Visible = False
cmdDrivingCycle.Enabled = False
mnuViewDrivingCycle.Enabled = False
mnuFileSave.Enabled = False
```

```
   gagPercentComplete.Visible = False



Exit Sub
error3: errorMsg$ = "Make sure the files FUELS.TXT and PRODUCTS.TXT are" + Chr$(13)
+ Chr$(10)
        errorMsg$ = errorMsg$ + "in the directory where the program was started"
        End
End Sub


Sub FourStroke (VolumetricEfficiency, EngineSpeed, PressManifold)
ReDim NumberMolesCombustor(10), NumberMoles(10), NumberMolesExhaust(10)
ReDim TempPressure(300), TempVolume(300)

    If txtPressManifold.Text = "" Then
          MsgBox "PLEASE SPECIFY AN INTAKE MANIFOLD PRESSURE"
          txtPressManifold.SetFocus
          Exit Sub
        End If




        TempManifold = Val(frmEngineParameters.txtAmbientTemperature.Text)
        TempCombustorFuel = Val(txtCombustorTemp.Text)
        PressExhaust = Val(frmEngineParameters.txtAmbientPressure.Text)
        EquivRatio = Val(frmEngineParameters.txtEquivRatio.Text)
        CombustorEquivRatio = Val(frmEngineParameters.txtCombustorEquivRatio.Text)
        YStoich = MC + MH / 4! - MO / 2!
        PressState1 = PressManifold
        TempState1 = TempManifold
        Residual = 0
        cmdSaveData.Visible = False
'
'Calculate Manifold Conditions for Normal Operation
'
   If frmEngineParameters.optEnrichment.Value = -1 Then

        Y = YStoich / EquivRatio
        HFuel = MC * Hzero(1) + MH / 2 * Hzero(10) - YStoich * Hzero(3) - Hrp + FuelACP
* (TempManifold - 298.16) + FuelBCP * (TempManifold ^ 2 - 298.16 ^ 2) / 2
        HOxygen = Y * (Hzero(3) + ACP(3) * (TempManifold - 298.16) + BCP(3) *
(TempManifold ^ 2 - 298.16 ^ 2) / 2)
        HNitrogen = 3.76 * Y * (Hzero(6) + ACP(6) * (TempManifold - 298.16) + BCP(6) *
(TempManifold ^ 2 - 298.16 ^ 2) / 2)
```

```
        HManifold = HFuel + HNitrogen + HOxygen
        HFreshState1 = HManifold
        VolumeState1 = (1 + 4.76 * Y) * 8.314 * TempState1 / PressManifold / 101
        MolesState1 = 1 + 4.76 * Y
        AbundancyCarbon = MC
        AbundancyHydrogen = MH
        AbundancyOxygen = (MO + Y * 2)
        AbundancyNitrogen = 7.52 * Y
        NumberMoles(3) = Y
        NumberMoles(6) = 3.76 * Y
        EnthalpyState1 = HManifold
  Else
'
'
' Calculate Manifold Conditions for Rich Combustor Operation
'
'
        Call RichCombustor

        YStoich = (1 - (1 / CombustorEquivRatio)) * YStoich
        Y = YStoich / EquivRatio
        MolesCombustorProduct = MolesProduct
        MolesState1 = MolesCombustorProduct + 4.76 * Y
        VolumeState1 = MolesState1 * 8.314 * TempState1 / PressManifold / 101
        'from RichCombustor AbundancyCarbon = NumberMoles(1) + NumberMoles(2)
        'from RichCombustor AbundancyHydrogen = NumberMoles(7) + NumberMoles(8) * 2
+ NumberMoles(10) * 2 + NumberMoles(9)
        AbundancyOxygen = (Y * 2) + NumberMoles(1) * 2 + NumberMoles(2) +
NumberMoles(10) + NumberMoles(4) + NumberMoles(3) * 2 + NumberMoles(9) +
NumberMoles(5)
        AbundancyNitrogen = 7.52 * Y + NumberMoles(5) + NumberMoles(6) * 2
        HOxygen = Y * (Hzero(3) + ACP(3) * (TempManifold - 298.16) + BCP(3) *
(TempManifold ^ 2 - 298.16 ^ 2) / 2)
        HNitrogen = 3.76 * Y * (Hzero(6) + ACP(6) * (TempManifold - 298.16) + BCP(6) *
(TempManifold ^ 2 - 298.16 ^ 2) / 2)
        'HFuel = HProducts(TempCombustorProducts)
        HFuel = HProducts(TempCombustorFuel)
        YCombustorStoich = MC + MH / 4! - MO / 2! / CombustorEquivRatio
        HCombustorFuel = MC * Hzero(1) + MH / 2 * Hzero(10) - YCombustorStoich *
Hzero(3) - Hrp + FuelACP * (TempManifold - 298.16) + FuelBCP * (TempManifold ^ 2 -
298.16 ^ 2) / 2 + HProducts(TempCombustorFuel) - HProducts(TempCombustorProducts)

        HManifold = HFuel + HNitrogen + HOxygen
        NumberMoles(3) = NumberMoles(3) + Y
        NumberMoles(6) = NumberMoles(6) + 3.76 * Y
        EnthalpyState1 = HOxygen + HNitrogen + HFuel
```

```
    End If

          TempState1Last = TempState1
'
'    Begin Iteration on TempState1
'
          pass% = 1

CompressionStroke:
'    Calculate the Amount of Fuel and Fresh Charge
'    after the Intake closes (volumetric efficiency effect)
'
          EfficiencyRatio = (VolumetricEfficiency + (1 - VolumetricEfficiency) / CompRatio)
          VolumeIntakeCloses = VolumeState1 * EfficiencyRatio
'
'
'    Calculate the amount of residual and fuel if burner is operating
'
          For i = 1 To 10
             NumberMoles(i) = NumberMoles(i) * VolumetricEfficiency
             NumberMolesCombustor(i) = NumberMoles(i) * VolumetricEfficiency
          Next

          MolesState1 = MolesState1 * VolumetricEfficiency
          AbundancyCarbon = AbundancyCarbon * VolumetricEfficiency
          AbundancyHydrogen = AbundancyHydrogen * VolumetricEfficiency
          AbundancyOxygen = AbundancyOxygen * VolumetricEfficiency
          AbundancyNitrogen = AbundancyNitrogen * VolumetricEfficiency
          EnthalpyState1 = EnthalpyState1 * VolumetricEfficiency
          UState1 = UState1 * VolumetricEfficiency
'
'
'  The sharp cornered Otto Cycle pressure values for
'  p2 and p3 are needed for the progressive combustion
'  calculations.  They will be obtained first.
'
'
'
'  Isentropic Compression of Residual + Fresh Charge
'  To State 2....State 1 conditions are known from
'  the Intake Stroke
'
             stepp$ = "Calculating Iteration Number " + Format$(pass%, "##")
   VolumeState2 = VolumeState1 / CompRatio
   Call Compression(VolumeState2)
   Temperature2 = TempState2
   Pressure2 = PressState2
   MolesState2 = MolesState1
   EnthalpyState2 = EnthalpyState1
   For i = 1 To 10
         EnthalpyState2 = EnthalpyState2 + (ACP(i) * (TempState2 - TempState1) + BCP(i) *
(TempState2 ^ 2 - TempState1 ^ 2) / 2) * NumberMoles(i)
   Next
   If frmEngineParameters.optRichCombustor.Value = 0 Then
         EnthalpyState2 = EnthalpyState2 + (FuelACP * (TempState2 - TempState1) +
FuelBCP * (TempState2 ^ 2 - TempState1 ^ 2) / 2) * (1 - Residual) * VolumetricEfficiency
   End If
   UState2 = EnthalpyState2 - MolesState2 * 8.314 * TempState2
'
'  Constant Volume Combustion Process
'  at VolumeState2
'
'
   VolumeState3 = VolumeState2
   Call Combustion(UState2, MolesState2)
   MolesState3 = MolesProduct
   UState3 = HProducts(TempState3) - 8.314 * MolesState3 * TempState3
   EntropyState3 = EntropyProducts(TempState3, PressState3)
'
'
'
'  The sharp cornered pressures are now known for
'  use in the progressive combustion calculations
'
'
'
'
'
'  Progressive Combustion Process
'
'  Calculate the Properties of the Fresh Charge/Residual
'  mixture at the Start of Cumbustion
'
   VolumeInitial = (VolumeState1 - VolumeState2) * (CompRatio / (CompRatio - 1) - (1 -
Cos(3.1416 - SparkAdvance * 3.1416 / 180)) / 2 + StrokeRatio - .5 * Sqr((2 * StrokeRatio) ^ 2
- (Sin(3.1416 - SparkAdvance * 3.1416 / 180)) ^ 2))
'
'  Recalculate the Composition of the Fresh Charge + Residual
'
         For i = 1 To 10
            NumberMoles(i) = NumberMolesCombustor(i)
         Next
```

```
' Calculate 100 P-V points on compression up to ignition

'  For i = 1 To 76
'     TempVolume(i) = VolumeState1 + Sin((i - 1) * 3.1416 / 180) * (VolumeState2 -
VolumeState1)
'     Call Compression(TempVolume(i))
'     TempPressure(i) = PressState2
'  Next

   Call Compression(VolumeInitial)
   TempInitial = TempState2
   TempState2 = Temperature2
   PressInitial = PressState2
   PressState2 = Pressure2


   EnthalpyInitial = EnthalpyState1
   For i = 1 To 10
         EnthalpyInitial = EnthalpyInitial + (ACP(i) * (TempInitial - TempState1) + BCP(i) *
(TempInitial ^ 2 - TempState1 ^ 2) / 2) * NumberMoles(i)
   Next
   EnthalpyInitial = EnthalpyInitial + (FuelACP * (TempInitial - TempState1) + FuelBCP *
(TempInitial ^ 2 - TempState1 ^ 2) / 2) * (1 - Residual)
   UInitial = EnthalpyInitial - MolesState1 * 8.314 * TempInitial
   kReactants = Log(PressInitial / PressState1) / Log(VolumeIntakeCloses / VolumeInitial)

   Call ProgressiveCombustion(VolumeInitial, TempInitial, PressInitial, kReactants,
SparkAdvance, CombustionDuration, MolesState1)
   EntropyFinal = EntropyProducts(TempFinal, PressFinal)
   MolesFinal = MolesProduct
   UFinal = HProducts(TempFinal) - 8.314 * MolesFinal * TempFinal


'
' Isentropic Expansion from VolumeFinal to
' VolumeState4
'
   VolumeState4 = VolumeState1

' Calculate 100 P-V points on expansion from VolumeFinal

'  For i = 1 To 75
'     TempVolume(106 + i) = VolumeState1 + Sin((75 - i) * 3.1416 / 180) * (VolumeState2 -
VolumeState1)
'     Call Expansion(TempFinal, PressFinal, EntropyFinal, TempVolume(106 + i) /
VolumeFinal, MolesFinal)
'     TempPressure(106 + i) = PressState4
'  Next

   Call Expansion(TempFinal, PressFinal, EntropyFinal, VolumeState1 / VolumeFinal,
MolesFinal)
   MolesState4 = MolesProduct
   PressState4 = PressState4
   UState4 = HProducts(TempState4) - 8.314 * MolesState4 * TempState4
'
'
'    Blowdown Process
'
'
   VolumeState5 = VolumeState1
   Call Blowdown(TempFinal, PressFinal, VolumeFinal, EntropyFinal)
'
' Exhaust Stroke
'
   VolumeState6 = VolumeState2
   TempState6 = TempState5
   PressState6 = PressExhaust
   Call Exhaust
   EnthalpyState6 = HProducts(TempState6)
'
' Intake Stroke
'
   Call Intake(HFuel, HManifold, EnthalpyState6)

   If Abs(TempState1 - TempState1Last) < 20 Then
        GoTo FullConverge
   Else
        If frmEngineParameters.optRichCombustor.Value = 0 Then
           TempState1Last = TempState1
           VolumeState1 = (MolesProduct + (1 - Residual) * (1 + 4.76 * Y)) * 8.314 *
TempState1 / PressState1 / 101
           MolesState1 = MolesProduct + (1 - Residual) * (1 + 4.76 * Y)
           AbundancyCarbon = (1 - Residual) * MC + NumberMoles(1) + NumberMoles(2)
           AbundancyHydrogen = (1 - Residual) * MH + NumberMoles(7) + NumberMoles(8)
* 2 + NumberMoles(10) * 2 + NumberMoles(9)
           AbundancyOxygen = (1 - Residual) * (MO + Y * 2) + NumberMoles(1) * 2 +
NumberMoles(2) + NumberMoles(10) + NumberMoles(4) + NumberMoles(3) * 2 +
NumberMoles(9) + NumberMoles(5)
           AbundancyNitrogen = (1 - Residual) * 7.52 * Y + NumberMoles(5) +
NumberMoles(6) * 2
           NumberMoles(3) = NumberMoles(3) + (1 - Residual) * Y
           NumberMoles(6) = NumberMoles(6) + (1 - Residual) * 3.76 * Y
           UState1 = EnthalpyState1 - MolesState1 * 8.314 * TempState1
```

```
          Else
             UState1 = EnthalpyState1 - MolesState1 * 8.314 * TempState1
             TempState1Last = TempState1
             VolumeState1 = (MolesProduct + (1 - Residual) * (MolesCombustorProduct + 4.76
* Y)) * 8.314 * TempState1 / PressState1 / 101
             MolesState1 = MolesProduct + (1 - Residual) * (MolesCombustorProduct + 4.76 *
Y)
              For i = 1 To 10
                    NumberMoles(i) = NumberMoles(i) + (1 - Residual) *
NumberMolesCombustor(i)
              Next
             NumberMoles(3) = NumberMoles(3) + Y
             NumberMoles(6) = NumberMoles(6) + 3.76 * Y
             'AbundancyCarbon = NumberMoles(1) + NumberMoles(2)
             'AbundancyHydrogen = NumberMoles(7) + NumberMoles(8) * 2 +
NumberMoles(10) * 2 + NumberMoles(9)
             'AbundancyOxygen = NumberMoles(1) + NumberMoles(2) * 2 + NumberMoles(10)
+ NumberMoles(4) + NumberMoles(3) * 2 + NumberMoles(9) + NumberMoles(5)
             'AbundancyNitrogen = NumberMoles(5) + NumberMoles(6) * 2

          End If
          pass% = pass% + 1
          GoTo CompressionStroke
   End If

FullConverge:
   WorkIntakeLoop = (1 - PressManifold) * 101 * (VolumeState2 - VolumeState1)
   IMEP = (UFinal - UState4 - UInitial + UState1 + WorkCombustion + WorkIntakeLoop) /
(VolumeState1 - VolumeState2) / 101
   'IMEP = (UState3 - UState4 - (UState2 - UState1)) / (VolumeState1 - VolumeState2) / 101
   WorkPerCycle = EngineDisplacement / 1000000! * IMEP * 101 'kiloJoules
   Watts = WorkPerCycle * EngineSpeed / 2 / 60 * 1000
   IndicatedHorsepower = Watts / 745.7
   FuelFlowRate = (1 - Residual) * (12 * MC + 16 * MO + MH) * 1000 * EngineDisplacement
/ 1000000! * EngineSpeed / 2 / 60 / (VolumeState1 - VolumeState2)
   If frmEngineParameters.optRichCombustor.Value = True Then
          ThermalEfficiency = IMEP * 101 / UState1 * (VolumeState1 - VolumeState2)
   Else
          ThermalEfficiency = Watts / (FuelFlowRate / (12 * MC + 16 * MO + MH) / (1 -
Residual) * (HFuel))
   End If
   FrictionHorsepower = ThermalEfficiency * .00567 * EngineSpeed * EngineDisplacement /
1000
   BrakeHorsepower = IndicatedHorsepower - FrictionHorsepower
   NumberCylinders = Val(frmEngineParameters.txtNumberCylinders.Text)
```

```
End Sub


Function HProducts (TempProducts)
   EnthalpyProducts = 0
   If TempProducts < 400 Then
        For j = 1 To 10
           EnthalpyProducts = Hzero(j) + ACP(j) * (TempProducts - 298.16) + BCP(j) *
(TempProducts ^ 2 - 298.16 ^ 2) / 2
        Next
   ElseIf TempProducts < 1600 Then
         For j = 1 To 10
          EnthalpyProducts = EnthalpyProducts + NumberMoles(j) * (AL(j) + BL(j) *
TempProducts + CL(j) * Log(TempProducts))
         Next
   Else
         For j = 1 To 10
          EnthalpyProducts = EnthalpyProducts + NumberMoles(j) * (AH(j) + BH(j) *
TempProducts + CH(j) * Log(TempProducts))
         Next
   End If
   HProducts = EnthalpyProducts
End Function


Sub Intake (HFuel, HManifold, EnthalpyState6)
ReDim Preserve NumberMolesCombustor(10), NumberMolesPrime(10)

   FuelACP = Val(Mid$(frmEngineParameters.lstFuel.Text, 53, 6))
   FuelBCP = Val(Mid$(frmEngineParameters.lstFuel.Text, 59, 8))


'
'  Determine the TempState6Prime of the Residual after
'  expanding to PressManifold
'
   'Call ExpandResidual

   'EnthalpyState6Prime = HProducts(TempState6Prime)
   EnthalpyState6Prime = EnthalpyState6
'
'  Reset the Combustor Output
'

        For i = 1 To 10
           NumberMolesCombustor(i) = NumberMolesCombustor(i) / VolumetricEfficiency
        Next
```

```
'
'
int1:
   If frmEngineParameters.optRichCombustor.Value = 0 Then
         HFuelState1 = MC * Hzero(1) + MH / 2 * Hzero(10) - YStoich * Hzero(3) - Hrp +
FuelACP * (TempState1 - 298.16) + FuelBCP * (TempState1 ^ 2 - 298.16 ^ 2) / 2
         HOxygenState1 = Y * (Hzero(3) + ACP(3) * (TempState1 - 298.16) + BCP(3) *
(TempState1 ^ 2 - 298.16 ^ 2) / 2)
         HNitrogenState1 = 3.76 * Y * (Hzero(6) + ACP(6) * (TempState1 - 298.16) + BCP(6)
* (TempState1 ^ 2 - 298.16 ^ 2) / 2)
         HFreshState1 = HFuelState1 + HNitrogenState1 + HOxygenState1
   Else
         For i = 1 To 10
            NumberMolesPrime(i) = NumberMoles(i)
            NumberMoles(i) = NumberMolesCombustor(i)
         Next
         NumberMoles(3) = NumberMoles(3) + Y
         NumberMoles(6) = NumberMoles(6) + 3.76 * Y
         HFreshState1 = HProducts(TempState1)
         For i = 1 To 10
            NumberMoles(i) = NumberMolesPrime(i)
         Next
   End If
'
'  Calculate the Enthalpy of the Residual at State 1
'  considering only those species we have Cp data for
'  CO2, CO, 02, N2, H2, H20
'

   HResidualState1 = EnthalpyState6Prime
   For i = 1 To 10
         HResidualState1 = HResidualState1 + (ACP(i) * (TempState1 - TempState6) + BCP(i)
* (TempState1 ^ 2 - TempState6 ^ 2) / 2) * NumberMoles(i)
   Next
   EnthalpyState1 = (1 - Residual) * HFreshState1 + HResidualState1
   HDelta = EnthalpyState1 - EnthalpyState6Prime - (1 - Residual) * HManifold
   If HDelta < 0 Then
         TempState1 = TempState1 + 16
         GoTo int1
   End If

   TempState1Last = TempState1
   TempState1 = TempState1 - 8
   k% = 1
int2:
   If frmEngineParameters.optRichCombustor.Value = 0 Then
```

```
        HFuelState1 = MC * Hzero(1) + MH / 2 * Hzero(10) - YStoich * Hzero(3) - Hrp +
FuelACP * (TempState1 - 298.16) + FuelBCP * (TempState1 ^ 2 - 298.16 ^ 2) / 2
        HOxygenState1 = Y * (Hzero(3) + ACP(3) * (TempState1 - 298.16) + BCP(3) *
(TempState1 ^ 2 - 298.16 ^ 2) / 2)
        HNitrogenState1 = 3.76 * Y * (Hzero(6) + ACP(6) * (TempState1 - 298.16) + BCP(6)
* (TempState1 ^ 2 - 298.16 ^ 2) / 2)
        HFreshState1 = HFuelState1 + HNitrogenState1 + HOxygenState1
  Else
        For i = 1 To 10
           NumberMolesPrime(i) = NumberMoles(i)
           NumberMoles(i) = NumberMolesCombustor(i)
        Next
        NumberMoles(3) = NumberMoles(3) + Y
        NumberMoles(6) = NumberMoles(6) + 3.76 * Y
        HFreshState1 = HProducts(TempState1)
        For i = 1 To 10
           NumberMoles(i) = NumberMolesPrime(i)
        Next
  End If
'
'  Calculate the Enthalpy of the Residual at State 1
'  considering only those species we have Cp data for
'  CO2, CO, O2, N2, H2, H20
'
  HResidualState1 = EnthalpyState6Prime
  For i = 1 To 10
        HResidualState1 = HResidualState1 + (ACP(i) * (TempState1 - TempState6) + BCP(i)
* (TempState1 ^ 2 - TempState6 ^ 2) / 2) * NumberMoles(i)
  Next
  EnthalpyState1 = (1 - Residual) * HFreshState1 + HResidualState1
  HDelta = EnthalpyState1 - EnthalpyState6Prime - (1 - Residual) * HManifold
  If HDelta > 0 Then
        TempState1Last = TempState1
        TempState1 = TempState1 - 8 / 2 ^ k%
  If Abs(TempState1 - TempState1Last) < 1 GoTo int3     'Convergance
        k% = k% + 1
        GoTo int2
  Else
        TempState1Last = TempState1
        TempState1 = TempState1 + 8 / 2 ^ k%
  If Abs(TempState1 - TempState1Last) < 1 GoTo int3     'Convergance
           k% = k% + 1
           GoTo int2
  End If
int3:
  If frmEngineParameters.optRichCombustor.Value = 0 Then
        MolesState1 = MolesProduct + (1 - Residual) * (MolesCombustorProduct + 4.76 * Y)
```

```
   Else
         MolesState1 = MolesProduct + (1 - Residual) * (1 + 4.76 * Y)
   End If
   IntakeData$ = " 1  " + "     " + Format$(TempState1, "####") + "     " + Format$(PressState1,
"##.##")
   For A% = 1 To 10
'
'for mole fraction of product, comment out the next line
'
'   IntakeDataData$ = IntakeDataData$ + "     " + FORMAT$(NumberMoles(A%), "00.000")
'
'for mole numbers of product, comment out the next line
'
   IntakeDataData$ = IntakeDataData$ + "  " + Format$((NumberMoles(A%) / MolesState1),
"0.000")
'
'
   Next
End Sub


Sub mnuChassisParameters_Click ()
   frmChassisParameters.Visible = True

End Sub


Sub mnuEngineParameters_Click ()
   frmEngineParameters.Show
End Sub


Sub mnuExit_Click ()
   End
End Sub


Sub mnuPrintCombustor_Click ()
   CMDialog1.DialogTitle = "PRINT COMBUSTOR PERFORMANCE TABLE"
   CMDialog1.PrinterDefault = True
   CMDialog1.CancelError = True
   On Error GoTo UserCancel3
   CMDialog1.Action = 5
   Printer.Print "Rich Combustor Operation"
   Printer.Print Mid$(frmEngineParameters.lstFuel.Text, 10, 5) + " with " + Format(1 /
Val(frmEngineParameters.txtCombustorEquivRatio.Text), "##.##") + " theoretical air."
   Printer.Print "Percents on Molar Basis"
   Printer.Print "CO2 --> " + Format(100 * NumberMolesCombustor(1) /
MolesCombustorProduct, "00.00") + " %"
   Printer.Print "CO  --> " + Format(100 * NumberMolesCombustor(2) /
MolesCombustorProduct, "00.00") + " %"
   Printer.Print "O2  --> " + Format(100 * NumberMolesCombustor(3) /
MolesCombustorProduct, "00.00") + " %"
   Printer.Print "O   --> " + Format(100 * NumberMolesCombustor(4) /
MolesCombustorProduct, "00.00") + " %"
   Printer.Print "NO  --> " + Format(100 * NumberMolesCombustor(5) /
MolesCombustorProduct, "00.00") + " %"
   Printer.Print "N2  --> " + Format(100 * NumberMolesCombustor(6) /
MolesCombustorProduct, "00.00") + " %"
   Printer.Print "H   --> " + Format(100 * NumberMolesCombustor(7) /
MolesCombustorProduct, "00.00") + " %"
   Printer.Print "H2  --> " + Format(100 * NumberMolesCombustor(8) /
MolesCombustorProduct, "00.00") + " %"
   Printer.Print "OH  --> " + Format(100 * NumberMolesCombustor(9) /
MolesCombustorProduct, "00.00") + " %"
   Printer.Print "H2O --> " + Format(100 * NumberMolesCombustor(10) /
MolesCombustorProduct, "00.00") + " %"
   Printer.Print "Temp = " + Format(TempCombustorProducts, "0000.00") + " K"
   Printer.EndDoc
UserCancel3:

Exit Sub
Resume



End Sub


Sub mnuPrintEngineInputs_Click ()
   CMDialog1.Action = 5
End Sub


Sub mnuPrintEngineOut_Click ()
   CMDialog1.DialogTitle = "PRINT SINGLE POINT ENGINE OUTPUT"
   CMDialog1.PrinterDefault = True
   CMDialog1.CancelError = True
   On Error GoTo UserCancel1
   CMDialog1.Action = 5
            lblFuelName.Caption = Mid$(frmEngineParameters.lstFuel.Text, 10, 5) + " with " +
frmEngineParameters.txtEquivRatio.Text + " theoretical air."
            Printer.Print lblFuelName.Caption
            Printer.Print ""
```

```
        Printer.Print "State  Temp  Press   CO2    CO     O2    O     NO     N2    H
H2    OH    H2O"
        Printer.Print ""
            Printer.Print IntakeData$
            Printer.Print IsentropicCompressionData$
            Printer.Print AdiabaticFlameData$
            Printer.Print " init    " + Format$(TempInitial, "####") + "    " +
Format$(PressInitial, "##.##")
            Printer.Print " final  " + Format$(TempFinal, "####") + "    " +
Format$(PressFinal, "##.##")
            Printer.Print IsentropicExpansionData$
            Printer.Print BlowdownData$
            Printer.Print ExhaustData$
            Printer.Print "Residual = " + Format$(Residual, "0.000")
            Printer.Print "IMEP = " + Format$(IMEP, "##.00") + " atm"
            Printer.Print "Indicated Power = " + Format$(IndicatedHorsepower, "###.00")
+ " hp"
            Printer.Print "Brake Power = " + Format$(BrakeHorsepower, "###.00") + " hp"
            Printer.Print "Fuel Flow = " + Format$(FuelFlowRate, "##.00") + " gm/sec"
            Printer.Print "MAP = " + Format$(PressManifold, "##.00") + " atm"
            Printer.Print "Engine Speed = " + Format$(EngineSpeed, "####") + " rpm"
            Printer.Print "Engine Displacement = " + Format$(EngineDisplacement / 1000,
"##.0") + " litre"
  Printer.NewPage
  Printer.EndDoc



UserCancel1:

Exit Sub

Resume

End Sub
```

Sub mnuPrintPerformanceTable_Click ()

```
  On Error GoTo UserCancel
  CMDialog1.DialogTitle = "PRINT ENGINE PERFORMANCE TABLE"
  CMDialog1.PrinterDefault = True
  CMDialog1.CancelError = True

  CMDialog1.Action = 5
        lblFuelName.Caption = Mid$(frmEngineParameters.lstFuel.Text, 10, 5) + " with " +
frmEngineParameters.txtEquivRatio.Text + " theoretical air."
        Printer.Print lblFuelName.Caption
        Printer.Print ""
        Printer.Print "Eng  MAP  Brake  Fuel  Vol.  CO2  CO   O2   O   NO   N2
H   H2   OH   H2O"
        Printer.Print "Spd        Power  Flow   Eff."
        Printer.Print " rpm  atm    hp   g/s"
        Printer.Print ""
        For i = 1 To 100
            For j = 1 To 21
              If PerformanceMapData$(i, j) = "" Then
                  GoTo NoData
              End If
            Printer.Print Format(i * 100, "00000") + "   " + Format((j - 1) / 20, "#.00") + "   "
+ PerformanceMapData$(i, j)
NoData:     Next
      Next

        Printer.Print "Engine Displacement " + Format(EngineDisplacement / 1000,
"###.#") + " litre"
        Printer.Print "Ambient Temperature " + Format(AmbientTemperature, "###") + " K"
        Printer.Print "Compression Ratio " + Format(CompRatio, "##")
        Printer.EndDoc
UserCancel:

Exit Sub



End Sub


Sub mnuSaveDrivCycData_Click ()
  Open "DrivCycl.txt" For Output As 1

  For i = 1 To 506
      Print #1, Format(i, "####") + "   " + Format(VehicleSpeed(i), "00.0") + "   " +
Format(CycleEngineSpeed(i), "0000") + "   " + Format(RequiredBrakePower(i), "00.0") + "   "
+ Format(CycleFuelFlowrate(i), "00.0")

  Next
  Close 1

End Sub


Sub mnuSavePerfData_Click ()
  Open "perfrmnc.txt" For Output As 1
  For i = 1 To 100
        For j = 0 To 20
          If PerformanceMapData$(i, j) = "" Then
                GoTo empty2
          End If
        Print #1, Format(i * 100, "####") + "   " + Format((j - 1) / 20, "0.00") + "   " +
Left$(PerformanceMapData$(i, j), 5) + "   " + Mid$(PerformanceMapData$(i, j), 7, 5) + "   " +
Mid$(PerformanceMapData$(i, j), 12, 6)

empty2:
        Next
  Next
  Close 1

End Sub


Sub mnuSingleOperatingPoint_Click ()
  If frmEngineParameters.optEnrichment.Value = True Then
    lblCombustorProductsTemp.Visible = False
    lblCombTempUnits.Visible = False
    txtCombustorTemp.Visible = False
  End If

  fraInputs.Visible = True
End Sub


Sub mnuViewCombustor_Click ()
  If frmEngineParameters.optEnrichment.Value = True Then
        lblCombustor.Caption = "The Rich Combustor option was not selected in Engine
Parameters."
        fraCombustorOutput.Visible = True
        Exit Sub
  End If

  Call RichCombustor
  fraCombustorOutput.Visible = True
  lblCombustor.Caption = Mid$(frmEngineParameters.lstFuel.Text, 10, 5) + " with " +
Format(1 / Val(frmEngineParameters.txtCombustorEquivRatio.Text), "##.##") + " theoretical
air."
  lstCombustorOutput.AddItem "Percents on Molar Basis"
  lstCombustorOutput.AddItem "CO2 --> " + Format(100 * NumberMolesCombustor(1) /
MolesCombustorProduct, "00.00") + " %"
  lstCombustorOutput.AddItem "CO  --> " + Format(100 * NumberMolesCombustor(2) /
MolesCombustorProduct, "00.00") + " %"
```

```
   lstCombustorOutput.AddItem "O2  --> " + Format(100 * NumberMolesCombustor(3) /
MolesCombustorProduct, "00.00") + " %"
   lstCombustorOutput.AddItem "O   --> " + Format(100 * NumberMolesCombustor(4) /
MolesCombustorProduct, "00.00") + " %"
   lstCombustorOutput.AddItem "NO  --> " + Format(100 * NumberMolesCombustor(5) /
MolesCombustorProduct, "00.00") + " %"
   lstCombustorOutput.AddItem "N2  --> " + Format(100 * NumberMolesCombustor(6) /
MolesCombustorProduct, "00.00") + " %"
   lstCombustorOutput.AddItem "H   --> " + Format(100 * NumberMolesCombustor(7) /
MolesCombustorProduct, "00.00") + " %"
   lstCombustorOutput.AddItem "H2  --> " + Format(100 * NumberMolesCombustor(8) /
MolesCombustorProduct, "00.00") + " %"
   lstCombustorOutput.AddItem "OH  --> " + Format(100 * NumberMolesCombustor(9) /
MolesCombustorProduct, "00.00") + " %"
   lstCombustorOutput.AddItem "H2O --> " + Format(100 * NumberMolesCombustor(10) /
MolesCombustorProduct, "00.00") + " %"
   lstCombustorOutput.AddItem "Temp = " + Format(TempCombustorProducts, "0000.00") +
" K"
End Sub


Sub mnuViewDrivingCycle_Click ()
   grphDrivingCycle.GraphTitle = "Vehicle Speed"
   grphDrivingCycle.AutoInc = 0
   grphDrivingCycle.NumSets = 1
   grphDrivingCycle.FontUse = 4
   grphDrivingCycle.FontSize = 100

   grphDrivingCycle.NumPoints = 506
   grphDrivingCycle.LeftTitle = "Vehicle Speed (mph)"
   grphDrivingCycle.BottomTitle = "Elapsed Time (sec)"
   For i = 1 To 506
           grphDrivingCycle.ThisSet = 1
           grphDrivingCycle.ThisPoint = i
           grphDrivingCycle.GraphData = VehicleSpeed(i - 1)
           grphDrivingCycle.XPosData = i - 1
   Next
   grphDrivingCycle.DrawMode = 2
   fraDrivingCycle.Visible = True
   optVehicleSpeed.Value = True

End Sub


Sub mnuViewHorsepowerMap_Click ()
   grphPerformanceMap.GraphTitle = "Horsepower Map"
   grphPerformanceMap.NumSets = 8
```

```
  grphPerformanceMap.NumPoints = 40
  grphPerformanceMap.AutoInc = 0
  For j = 20 To 13 Step -1
       For i = 1 To 40
         grphPerformanceMap.ThisSet = 21 - j
         grphPerformanceMap.ThisPoint = i
         grphPerformanceMap.GraphData = Val(Left$(PerformanceMapData$(i, j), 5))
         grphPerformanceMap.XPosData = i * 100
       Next
  Next
  grphPerformanceMap.DrawMode = 2
  fraPerformanceMap.Visible = True
End Sub


Sub mnuViewPerformanceTable_Click ()
  lblPerformanceData.Caption = Mid$(frmEngineParameters.lstFuel.Text, 10, 5) + " with " +
frmEngineParameters.txtEquivRatio.Text + " theoretical air."
  fraPerformanceData.Visible = True
  For i = 1 To 100
       For j = 1 To 21
         If PerformanceMapData$(i, j) = "" Then
              GoTo empty1
         End If
       lstPerformanceData.AddItem Format(i * 100, "####") + "   " + Format((j - 1) / 20,
"0.00") + "   " + Left$(PerformanceMapData$(i, j), 5) + "   " + Mid$(PerformanceMapData$(i,
j), 7, 5) + "   " + Mid$(PerformanceMapData$(i, j), 12, 6)

empty1:
       Next
  Next
End Sub


Sub mnuViewSinglePointExhaust_Click ()
  fraExhaustComposition.Visible = True
  lblExhaustComposition.Caption = Mid$(frmEngineParameters.lstFuel.Text, 10, 5) + " with "
+ Format(1 / Val(frmEngineParameters.txtEquivRatio.Text), "##.##") + " theoretical air."
  lstExhaustComposition.AddItem "Percents on Molar Basis"
  lstExhaustComposition.AddItem "CO2 --> " + Format(100 * NumberMolesExhaust(1) /
MolesExhaustProduct, "00.000000") + " %"
  lstExhaustComposition.AddItem "CO  --> " + Format(100 * NumberMolesExhaust(2) /
MolesExhaustProduct, "00.000000") + " %"
  lstExhaustComposition.AddItem "O2  --> " + Format(100 * NumberMolesExhaust(3) /
MolesExhaustProduct, "00.000000") + " %"
  lstExhaustComposition.AddItem "O   --> " + Format(100 * NumberMolesExhaust(4) /
MolesExhaustProduct, "00.000000") + " %"
```

```
   lstExhaustComposition.AddItem "NO  --> " + Format(100 * NumberMolesExhaust(5) /
MolesExhaustProduct, "00.000000") + " %"
   lstExhaustComposition.AddItem "N2  --> " + Format(100 * NumberMolesExhaust(6) /
MolesExhaustProduct, "00.000000") + " %"
   lstExhaustComposition.AddItem "H   --> " + Format(100 * NumberMolesExhaust(7) /
MolesExhaustProduct, "00.000000") + " %"
   lstExhaustComposition.AddItem "H2  --> " + Format(100 * NumberMolesExhaust(8) /
MolesExhaustProduct, "00.000000") + " %"
   lstExhaustComposition.AddItem "OH  --> " + Format(100 * NumberMolesExhaust(9) /
MolesExhaustProduct, "00.000000") + " %"
   lstExhaustComposition.AddItem "H2O --> " + Format(100 * NumberMolesExhaust(10) /
MolesExhaustProduct, "00.000000") + " %"

End Sub


Sub mnuViewStatePoints_Click ()
   lblFuelName.Caption = Mid$(frmEngineParameters.lstFuel.Text, 10, 5) + " with " +
frmEngineParameters.txtEquivRatio.Text + " theoretical air."
   lstFlameTemperature.AddItem Left$(IntakeData$, 21)
   lstFlameTemperature.AddItem Left$(IsentropicCompressionData$, 21)
   lstFlameTemperature.AddItem " init    " + Format$(TempInitial, "####") + "    " +
Format$(PressInitial, "##.##")
   lstFlameTemperature.AddItem Left$(AdiabaticFlameData$, 21)
   lstFlameTemperature.AddItem " final  " + Format$(TempFinal, "####") + "    " +
Format$(PressFinal, "##.##")
   lstFlameTemperature.AddItem Left$(IsentropicExpansionData$, 21)
   lstFlameTemperature.AddItem Left$(BlowdownData$, 21)
   lstFlameTemperature.AddItem Left$(ExhaustData$, 21)
   lstFlameTemperature.AddItem "Residual = " + Format$(Residual, "0.000")
   lstFlameTemperature.AddItem "IMEP = " + Format$(IMEP, "##.00") + " atm"
   lstFlameTemperature.AddItem "INDICATED POWER = " + Format$(IndicatedHorsepower,
"##.00") + " hp"
   lstFlameTemperature.AddItem "BRAKE POWER = " + Format$(BrakeHorsepower,
"##.00") + " hp"
   FuelFlowRate = (1 - Residual) * (12 * MC + 16 * MO + MH) * 1000 * EngineDisplacement
/ 1000000! * EngineSpeed / 2 / 60 / (VolumeState1 - VolumeState2)
   lstFlameTemperature.AddItem "FUEL FLOW = " + Format$(FuelFlowRate, "##.00") + "
gm/sec"

   fraInputs.Visible = False
   fraOutput.Visible = True
   cmdReturn.SetFocus
End Sub


Sub optEngineSpeed_Click ()
```

```
    grphDrivingCycle.GraphTitle = "Engine Speed"
    grphDrivingCycle.AutoInc = 0
    grphDrivingCycle.NumSets = 1
    grphDrivingCycle.NumPoints = 506
    grphDrivingCycle.LeftTitle = "Engine Speed (rpm)"
    grphDrivingCycle.BottomTitle = "Elapsed Time (sec)"
    For i = 1 To 506
            grphDrivingCycle.ThisSet = 1
            grphDrivingCycle.ThisPoint = i
            grphDrivingCycle.GraphData = CycleEngineSpeed(i - 1)
            grphDrivingCycle.XPosData = i - 1
            grphDrivingCycle.XPosData = i - 1
    Next
    grphDrivingCycle.DrawMode = 2
    fraDrivingCycle.Visible = True

End Sub


Sub optFuelFlowrate_Click ()
    grphDrivingCycle.GraphTitle = "Fuel Consumption Map"
    grphDrivingCycle.AutoInc = 0
    grphDrivingCycle.NumSets = 1
    grphDrivingCycle.NumPoints = 506
    grphDrivingCycle.LeftTitle = "Fuel Flow Rate (gps)"
    grphDrivingCycle.BottomTitle = "Elapsed Time (sec)"
    For i = 1 To 506
            grphDrivingCycle.ThisSet = 1
            grphDrivingCycle.ThisPoint = i
            grphDrivingCycle.GraphData = CycleFuelFlowrate(i - 1)
            grphDrivingCycle.XPosData = i - 1
    Next
    grphDrivingCycle.DrawMode = 2
    fraDrivingCycle.Visible = True

End Sub


Sub optRequiredPower_Click ()
    grphDrivingCycle.GraphTitle = "Horsepower Map"
    grphDrivingCycle.AutoInc = 0
    grphDrivingCycle.NumSets = 1
    grphDrivingCycle.NumPoints = 506
    grphDrivingCycle.LeftTitle = "Required Engine Power (hp)"
    grphDrivingCycle.BottomTitle = "Elapsed Time (sec)"
    For i = 1 To 506
            grphDrivingCycle.ThisSet = 1
            grphDrivingCycle.ThisPoint = i
            grphDrivingCycle.GraphData = RequiredBrakePower(i - 1)
            grphDrivingCycle.XPosData = i - 1
   Next
   grphDrivingCycle.DrawMode = 2
   fraDrivingCycle.Visible = True

End Sub


Sub optVehicleSpeed_Click ()
   grphDrivingCycle.GraphTitle = "Vehicle Speed"
   grphDrivingCycle.AutoInc = 0
   grphDrivingCycle.NumSets = 1
   grphDrivingCycle.NumPoints = 506
   grphDrivingCycle.LeftTitle = "Vehicle Speed (mph)"
   grphDrivingCycle.BottomTitle = "Elapsed Time (sec)"
   For i = 1 To 506
            grphDrivingCycle.ThisSet = 1
            grphDrivingCycle.ThisPoint = i
            grphDrivingCycle.GraphData = VehicleSpeed(i - 1)
            grphDrivingCycle.XPosData = i - 1
   Next
   grphDrivingCycle.DrawMode = 2
   fraDrivingCycle.Visible = True

End Sub


Sub ProgressiveCombustion (VolumeInitial, TempInitial, PressInitial, kReactants,
SparkAdvance, CombustionDuration, MolesReactant)
    Static PressProgressive(101), TempProgressive(101), VolumeProgressive(101)
   CompRatio = Val(frmEngineParameters.txtCompRatio.Text)
   WorkCombustion = 0
   FractionBurned = 0
   PressProgressive(1) = PressInitial
   VolumeProgressive(1) = VolumeInitial
   TempProgressive(1) = TempInitial
   CrankIncrement = CombustionDuration / 100
   DeltaN = 1 / 30  'Use linear relation
   For i = 2 To 31

'
'  Evaluate k for the Products of Combustion
'

         CpProducts = 0
         If TempProgressive(i - 1) > 400 Then
```

```
             If TempProgressive(i - 1) > 1600 Then
               A = PressInitial * TempProgressive(i - 1) / (MolesReactant * TempInitial)
               Call Equilibrium(TempProgressive(i - 1), PressProgressive(i - 1), A)
               For j = 1 To 10
                    CpProducts = CpProducts + NumberMoles(j) / MolesProduct * (BH(j)
+ CH(j) / TempProgressive(i - 1))
               Next
             Else
               For j = 1 To 10
                    CpProducts = CpProducts + NumberMoles(j) / MolesProduct * (BL(j)
+ CL(j) / TempProgressive(i - 1))
               Next

             End If
         kProducts = CpProducts / (CpProducts - 8.314)
       Else
         For j = 1 To 10
         kProducts = kReactants
         Next
       End If
       FractionBurned = FractionBurned + DeltaN
       kMixture = kReactants + (kProducts - kReactants) * FractionBurned
       VolumeProgressive(i) = (VolumeState1 - VolumeState2) * (CompRatio / (CompRatio -
1) - (1 - Cos(3.1416 - (SparkAdvance - CrankIncrement * (i - 1)) * 3.1416 / 180)) / 2 +
StrokeRatio - .5 * Sqr((2 * StrokeRatio) ^ 2 - (Sin(3.1416 - (SparkAdvance - CrankIncrement *
(i - 1)) * 3.1416 / 180)) ^ 2))
       PressProgressive(i) = PressProgressive(i - 1) - (PressProgressive(i - 1) * kMixture *
(VolumeProgressive(i) - VolumeProgressive(i - 1)) / VolumeProgressive(i)) + (PressState3 -
PressState2) * VolumeState2 / VolumeProgressive(i) * DeltaN
       TempProgressive(i) = 101 * PressProgressive(i) * VolumeProgressive(i) / (8.314 *
(FractionBurned * MolesProduct + (1 - FractionBurned) * (1 - Residual) * (1 + 4.76 * Y)))
       WorkCombustion = WorkCombustion + (PressProgressive(i) + PressProgressive(i - 1))
* 50.5 * (VolumeProgressive(i) - VolumeProgressive(i - 1))
       TempVolume(i + 75) = VolumeProgressive(i)
       TempPressure(i + 75) = PressProgressive(i)
  Next
  TempFinal = TempProgressive(31)
  PressFinal = PressProgressive(31)
  VolumeFinal = VolumeProgressive(31)
End Sub


Sub RichCombustor ()
  FuelACP = Val(Mid$(frmEngineParameters.lstFuel.Text, 53, 6))
  FuelBCP = Val(Mid$(frmEngineParameters.lstFuel.Text, 59, 8))
  FuelTemp = Val(frmEngineParameters.txtAmbientTemperature.Text)
  OxidizerTemp = FuelTemp
  PressReactants = Val(frmEngineParameters.txtAmbientPressure.Text)
  YCombustorStoich = MC + MH / 4! + -MO / 2!
10  Y = YCombustorStoich / Val(frmEngineParameters.txtCombustorEquivRatio.Text)
         AO = MO + Y * 2
         AN = 7.52 * Y
         AbundancyCarbon = MC
         AbundancyHydrogen = MH
         AbundancyOxygen = (MO + Y * 2)
         AbundancyNitrogen = 7.52 * Y
         HFuel = MC * Hzero(1) + MH / 2 * Hzero(10) - YStoich * Hzero(3) - Hrp +
FuelACP * (FuelTemp - 298.16) + FuelBCP * (FuelTemp ^ 2 - 298.16 ^ 2) / 2
         HOxygen = Y * (Hzero(3) + ACP(3) * (OxidizerTemp - 298.16) + BCP(3) *
(OxidizerTemp ^ 2 - 298.16 ^ 2) / 2)
         HNitrogen = 3.76 * Y * (Hzero(6) + ACP(6) * (OxidizerTemp - 298.16) + BCP(6) *
(OxidizerTemp ^ 2 - 298.16 ^ 2) / 2)
         HReactants = HFuel + HNitrogen + HOxygen
             NumberMoles(1) = MC              'CO2
             NumberMoles(2) = 0              'CO
             NumberMoles(3) = Y - YStoich        'O2
             NumberMoles(6) = 3.76 * Y       'N2
             NumberMoles(10) = MH / 2          'H2O
         MolesProduct = 0
         For I% = 1 To 10
             MolesProduct = MolesProduct + NumberMoles(I%)
         Next
         MolesProductSimple = MolesProduct
         TempProducts = 1600 'Initial Value for iteration
'
' Constant Pressure Combustion
'
         PressProducts = PressReactants
'
'  Increase TempProducts until HDelta becomes negative
'
20        If TempProducts >= 1600 Then
             Call Equilibrium(TempProducts, PressProducts, 0)
         Else
             MolesProduct = MolesProductSimple
         End If
         HDelta = HReactants - HProducts(TempProducts)
         If HDelta > 0 Then
             TempProducts = TempProducts + 100
             GoTo 20
         End If
'
'  The Adiabatic Flame Temperature has just been passed when HDelta became negative
'  Begin Bisection Method
```

```
'
            TempProdLast = TempProducts
            TempProducts = TempProducts - 50
            k% = 1
50        If TempProducts > 1600 Then
                  Call Equilibrium(TempProducts, PressProducts, 0)
            Else
                  MolesProduct = MolesProductSimple
            End If
            HDelta = HReactants - HProducts(TempProducts)
            If HDelta > 0 Then
                  TempProdLast = TempProducts
                  TempProducts = TempProducts + 50! / 2 ^ k%
                  If Abs(TempProducts - TempProdLast) < 1 GoTo 54      'Convergance to 1
Kelvin
                  k% = k% + 1
                  GoTo 50
            Else
                  TempProdLast = TempProducts
                  TempProducts = TempProducts - 50! / 2 ^ k%
                  If Abs(TempProducts - TempProdLast) < 1 GoTo 54      'Convergance to 1
Kelvin
                  k% = k% + 1
                  GoTo 50
            End If
'  Temperature and Composition of Combustor Products has
'  been determined for Ambient Pressure ---
'     TempProducts  NumberMoles(i)  PressProducts
'
'
'  The Combustor Products will now be expanded isentropically
'  to the pressure in the intake manifold
'
54        TempExpand = TempProducts
            PressExpand = PressManifold


'
'  Increase TempExpand until SDelta becomes negative
'
55        SDelta = EntropyProducts(TempProducts, PressProducts) -
EntropyProducts(TempExpand, PressExpand)
            If SDelta < 0 Then
                  TempExpand = TempExpand - 100
                  GoTo 55
            End If
'
'  The Isentropic Expansion Temperature has just been passed when SDelta became negative
'  Begin Bisection Method
'
           TempExpLast = TempExpand
           TempExpand = TempExpand + 50
           k% = 1
56       If TempProducts > 1600 Then
                 Call Equilibrium(TempProducts, PressProducts, 0)
           End If
           SDelta = EntropyProducts(TempProducts, PressProducts) -
EntropyProducts(TempExpand, PressExpand)
           If SDelta < 0 Then
                 TempExpLast = TempExpand
                 TempExpand = TempExpand - 50! / 2 ^ k%
                 If Abs(TempExpand - TempExpLast) < 1 GoTo 57      'Convergance
                 k% = k% + 1
                 GoTo 56
           Else
                 TempExpLast = TempExpand
                 TempExpand = TempExpand + 50! / 2 ^ k%
                 If Abs(TempExpand - TempExpLast) < 1 GoTo 57      'Convergance
                 k% = k% + 1
                 GoTo 56
           End If
57     TempCombustorProducts = TempExpand
        MolesCombustorProduct = MolesProduct
        For i = 1 To 10
          NumberMolesCombustor(i) = NumberMoles(i)
        Next

End Sub


  ClientWidth    =   5160
  ForeColor      =   &H00000000&
  Height         =   4620
  Left           =   3615
  ScaleHeight    =   4215
  ScaleWidth     =   5160
  Top            =   1050
  Width          =   5280
  Begin CommandButton cmdChassisParametersCancel
    Caption         =   "Cancel"
    Height          =   375
    Left            =   3840
    TabIndex        =   9
    Top             =   3720
    Width           =   1095
```

```
End
Begin CommandButton cmdChassisParametersOK
  Caption      =  "OK"
  Height       =  375
  Left         =  2760
  TabIndex     =  10
  Top          =  3720
  Width        =  975
End
Begin Frame Frame1
  BackColor    =  &H00FFFFFF&
  Caption      =  "Vehicle Weight"
  ForeColor    =  &H00000000&
  Height       =  1095
  Left         =  120
  TabIndex     =  24
  Top          =  360
  Width        =  2175
  Begin CommandButton cmdSelectVehicleWeight
    BackColor    =  &H00FFFFFF&
    Caption      =  "Select..."
    Height       =  375
    Left         =  1200
    TabIndex     =  26
    Top          =  480
    Width        =  855
  End
  Begin TextBox txtVehicleWeight
    BackColor    =  &H00FFFFFF&
    ForeColor    =  &H00000000&
    Height       =  375
    Left         =  120
    TabIndex     =  25
    Top          =  480
    Width        =  600
  End
  Begin Label Label7
    BackColor    =  &H00FFFFFF&
    Caption      =  "lbs"
    ForeColor    =  &H00000000&
    Height       =  240
    Left         =  800
    TabIndex     =  27
    Top          =  600
    Width        =  360
  End
End
```

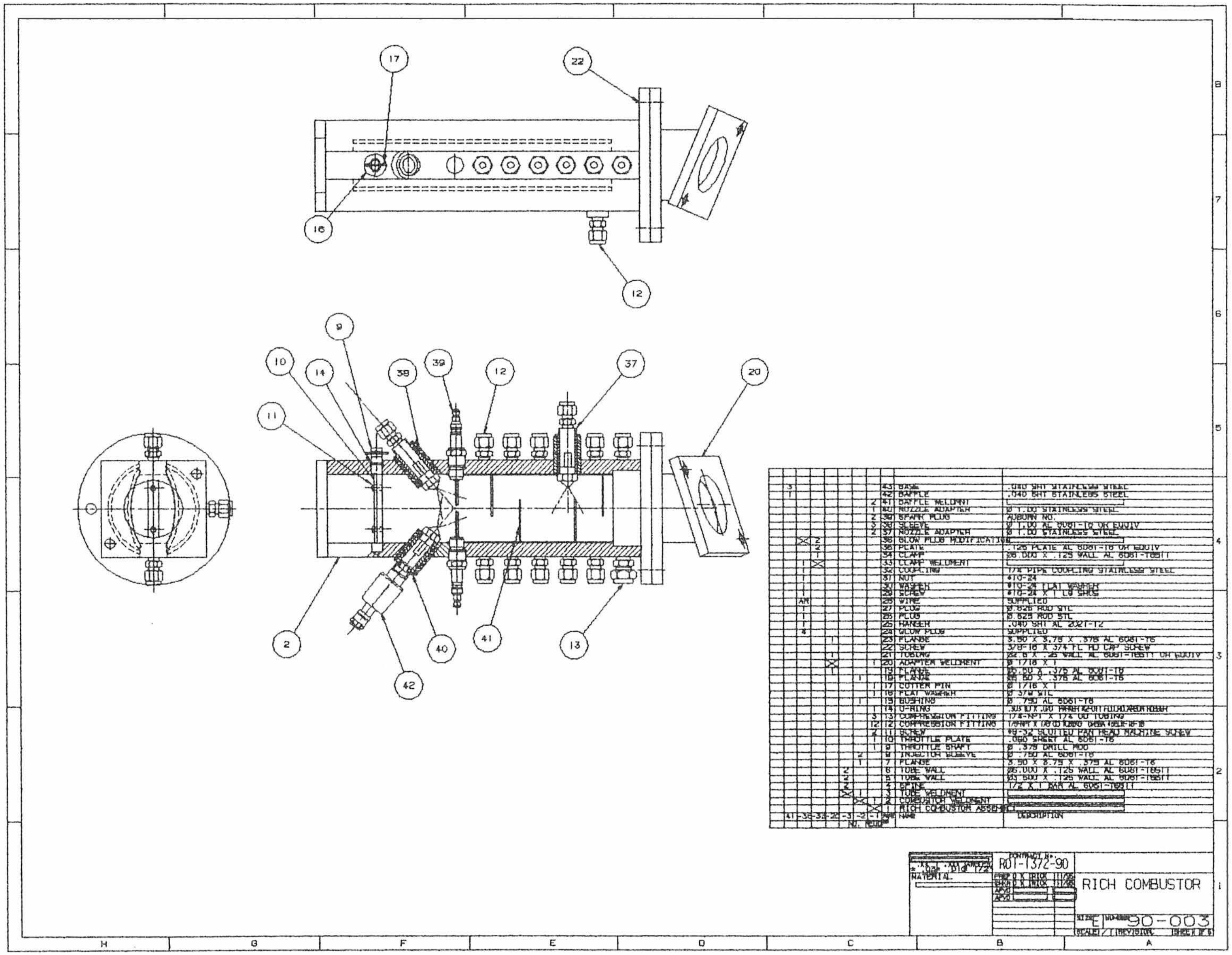
```
Begin Frame fraTireSize
  BackColor    =  &H00FFFFFF&
  Caption      =  "Tire Size"
  ForeColor    =  &H00000000&
  Height       =  1095
  Left         =  120
  TabIndex     =  0
  Top          =  1680
  Width        =  2175
  Begin TextBox txtTireDiameter
    BackColor    =  &H00FFFFFF&
    ForeColor    =  &H00000000&
    Height       =  375
    Left         =  120
    TabIndex     =  4
    Top          =  480
    Width        =  600
  End
  Begin CommandButton cmdSelectTireSize
    BackColor    =  &H00C0C0C0&
    Caption      =  "Select..."
    Height       =  375
    Left         =  1200
    TabIndex     =  5
    Top          =  480
    Width        =  840
  End
  Begin Label lblTireSizeUnits
    BackColor    =  &H00FFFFFF&
    Caption      =  "in"
    ForeColor    =  &H00000000&
    Height       =  240
    Left         =  795
    TabIndex     =  7
    Top          =  600
    Width        =  255
  End
End
Begin Frame fraAxleRatio
  BackColor    =  &H00FFFFFF&
  Caption      =  "Axle Ratio"
  ForeColor    =  &H00000000&
  Height       =  1095
  Left         =  120
  TabIndex     =  6
  Top          =  3000
  Width        =  2175
  Begin TextBox txtAxleRatio
    BackColor     =  &H00FFFFFF&
    ForeColor     =  &H00000000&
    Height        =  375
    Left          =  120
    TabIndex      =  1
    Top           =  480
    Width         =  600
  End
  Begin CommandButton cmdSelectAxleRatio
    BackColor     =  &H00FFFFFF&
    Caption       =  "Select..."
    Height        =  375
    Left          =  1200
    TabIndex      =  2
    Top           =  480
    Width         =  855
  End
  Begin Label Label2
    BackColor     =  &H00FFFFFF&
    Caption       =  ":1"
    ForeColor     =  &H00000000&
    Height        =  240
    Left          =  800
    TabIndex      =  13
    Top           =  600
    Width         =  360
  End
End
Begin Frame fraOverdrive
  BackColor     =  &H00FFFFFF&
  Caption       =  "Transmission Ratios"
  ForeColor     =  &H00000000&
  Height        =  2895
  Left          =  2760
  TabIndex      =  8
  Top           =  360
  Width         =  2175
  Begin TextBox txtTransmissionRatio2
    BackColor     =  &H00FFFFFF&
    ForeColor     =  &H00000000&
    Height        =  375
    Left          =  120
    TabIndex      =  19
    Top           =  840
    Width         =  600
  End
```

```
Begin TextBox txtTransmissionRatio3
  BackColor     =  &H00FFFFFF&
  ForeColor     =  &H00000000&
  Height        =  375
  Left          =  120
  TabIndex      =  18
  Top           =  1320
  Width         =  600
End
Begin TextBox txtTransmissionRatio4
  BackColor     =  &H00FFFFFF&
  ForeColor     =  &H00000000&
  Height        =  375
  Left          =  120
  TabIndex      =  17
  Top           =  1800
  Width         =  600
End
Begin TextBox txtTransmissionRatio5
  BackColor     =  &H00FFFFFF&
  ForeColor     =  &H00000000&
  Height        =  375
  Left          =  120
  TabIndex      =  16
  Top           =  2280
  Width         =  600
End
Begin TextBox txtTransmissionRatio1
  BackColor     =  &H00FFFFFF&
  ForeColor     =  &H00000000&
  Height        =  375
  Left          =  120
  TabIndex      =  3
  Top           =  360
  Width         =  600
End
Begin CommandButton cmdSelectTransRatio
  BackColor     =  &H00FFFFFF&
  Caption       =  "Select..."
  Height        =  375
  Left          =  1200
  TabIndex      =  14
  Top           =  2280
  Width         =  840
End
Begin Label Label6
  BackColor     =  &H00FFFFFF&
  Caption      = "2nd"
  ForeColor    = &H00000000&
  Height       = 240
  Left         = 750
  TabIndex     = 23
  Top          = 960
  Width        = 360
End
Begin Label Label5
  BackColor    = &H00FFFFFF&
  Caption      = "3rd"
  ForeColor    = &H00000000&
  Height       = 240
  Left         = 750
  TabIndex     = 22
  Top          = 1440
  Width        = 360
End
Begin Label Label4
  BackColor    = &H00FFFFFF&
  Caption      = "4th"
  ForeColor    = &H00000000&
  Height       = 240
  Left         = 750
  TabIndex     = 21
  Top          = 1920
  Width        = 360
End
Begin Label Label1
  BackColor    = &H00FFFFFF&
  Caption      = "5th"
  ForeColor    = &H00000000&
  Height       = 240
  Left         = 750
  TabIndex     = 20
  Top          = 2400
  Width        = 360
End
Begin Label Label3
  BackColor    = &H00FFFFFF&
  Caption      = "1st"
  ForeColor    = &H00000000&
  Height       = 240
  Left         = 750
  TabIndex     = 15
  Top          = 480
  Width        = 360
```

```
    End
  End
  Begin OptionButton optEnglish
    BackColor     =  &H00FFFFFF&
    Caption       =  "English"
    ForeColor     =  &H00000000&
    Height        =  240
    Left          =  120
    TabIndex      =  11
    Top           =  0
    Value         =  -1  'True
    Width         =  975
  End
  Begin OptionButton optMetric
    BackColor     =  &H00FFFFFF&
    Caption       =  "Metric"
    ForeColor     =  &H00000000&
    Height        =  240
    Left          =  1200
    TabIndex      =  12
    TabStop       =  0  'False
    Top           =  0
    Width         =  855
  End
End


Sub cmdChassisParametersOK_Click ()
  TireDiameter = Val(txtTireDiameter.Text)
  AxleRatio = Val(txtAxleRatio.Text)
  TransmissionRatio1 = Val(txtTransmissionRatio1.Text)
  TransmissionRatio2 = Val(txtTransmissionRatio2.Text)
  TransmissionRatio3 = Val(txtTransmissionRatio3.Text)
  TransmissionRatio4 = Val(txtTransmissionRatio4.Text)
  TransmissionRatio5 = Val(txtTransmissionRatio5.Text)
  VehicleMass = Val(txtVehicleWeight.Text) / 32.2
   If optMetric.Value = -1 Then
     TireDiameter = TireDiameter / 25.4
   End If
  frmChassisParameters.Hide
End Sub


Sub cmdSelectAxleRatio_Click ()
  txtAxleRatio.SetFocus
  frmAxleRatio.Show
End Sub
```

```
Sub cmdSelectTireSize_Click ()
 txtTireDiameter.SetFocus
 frmTires.Show
End Sub


Sub Form_Load ()
 txtTransmissionRatio1.Text = "3.50"
 txtTransmissionRatio2.Text = "2.05"
 txtTransmissionRatio3.Text = "1.38"
 txtTransmissionRatio4.Text = "0.94"
 txtTransmissionRatio5.Text = "0.72"
 txtAxleRatio.Text = "3.61"
 txtVehicleWeight.Text = "3000"
 txtTireDiameter.Text = "24.75"
 optMetric.Enabled = False
 cmdSelectVehicleWeight.Enabled = False
 cmdSelectAxleRatio.Enabled = False
 cmdSelectTransRatio.Enabled = False
End Sub


Sub optEnglish_Click ()
 lblTireSizeUnits.Caption = "in"
End Sub


Sub optMetric_Click ()
 lblTireSizeUnits.Caption = "mm"
End Sub


Sub txtTireDiameter_GotFocus ()
  txtTireDiameter.Text = clipboard.GetText()
  clipboard.Clear
End Sub


Sub txtTransmissionRatio_GotFocus ()
End Sub
```

Appendix B



	-41	-36	-34	-20	-3	-2	-1	PART NO.	NAME	DESCRIPTION
	3							43	BASE	.040 SHT STAINLESS STEEL
	1							42	BAFFLE	.040 SHT STAINLESS STEEL
							2	41	BAFFLE WELDMNT	
							1	40	NOZZLE ADAPTER	Ø 1.00 STAINLESS STEEL
							2	39	SPARK PLUG	AUBURN NO.
							3	38	SLEEVE	Ø 1.00 AL 6061-T6 OR EQUIV
							2	37	NOZZLE ADAPTER	Ø 1.00 STAINLESS STEEL
		☒	2					36	GLOW PLUG MODIFICATION	
			2					35	PLATE	.125 PLATE AL 6061-T6 OR EQUIV
			1					34	CLAMP	Ø8.000 X .125 WALL AL 6061-T6511
		1	☒					33	CLAMP WELDMENT	
		1						32	COUPLING	1/4 PIPE COUPLING STAINLESS STEEL
		1						31	NUT	#10-24
		1						30	WASHER	#10-24 FLAT WASHER
		1						29	SCREW	#10-24 X 1 LG SHCS
		AR						28	WIRE	SUPPLIED
		1						27	PLUG	Ø.625 ROD STL
		1						26	PLUG	Ø.625 ROD STL
		1						25	HANGER	.040 SHT AL 2024-T3
		4						24	GLOW PLUG	SUPPLIED
				1				23	FLANGE	3.50 X 3.75 X .375 AL 6061-T6
								22	SCREW	3/8-16 X 3/4 FL HD CAP SCREW
				1				21	TUBING	Ø2.5 X .25 WALL AL 6061-T6511 OR EQUIV
				☒			1	20	ADAPTER WELDMENT	Ø 1/16 X 1
				1				19	FLANGE	Ø6.50 X .375 AL 6061-T6
						1		18	FLANGE	Ø8.50 X .375 AL 6061-T6
							1	17	COTTER PIN	Ø 1/16 X 1
							1	16	FLAT WASHER	Ø 3/8 STL
						1		15	BUSHING	Ø .750 AL 6061-T6
							1	14	O-RING	.303 ID X .070 PARKER #2-011 FLUOROCARBON RUBBER
							3	13	COMPRESSION FITTING	1/4-NPT X 1/4 OD TUBING
							12	12	COMPRESSION FITTING	1/8NPT X 1/8 OD TUBING [illegible]
							2	11	SCREW	#8-32 SLOTTED PAN HEAD MACHINE SCREW
							1	10	THROTTLE PLATE	.060 SHEET AL 6061-T6
							1	9	THROTTLE SHAFT	Ø .375 DRILL ROD
						2		8	INJECTOR SLEEVE	Ø .750 AL 6061-T6
						1		7	FLANGE	8.50 X 8.75 X .375 AL 6061-T6
					2			6	TUBE WALL	Ø6.000 X .125 WALL AL 6061-T6511
					2			5	TUBE WALL	Ø3.500 X .125 WALL AL 6061-T6511
					2			4	SPINE	1/2 X 1 BAR AL 6061-T6511
					☒	1		3	TUBE WELDMENT	
						☒	1	2	COMBUSTOR WELDMENT	
							☒	1	RICH COMBUSTOR ASSEMBLY	
	NO. REQD									

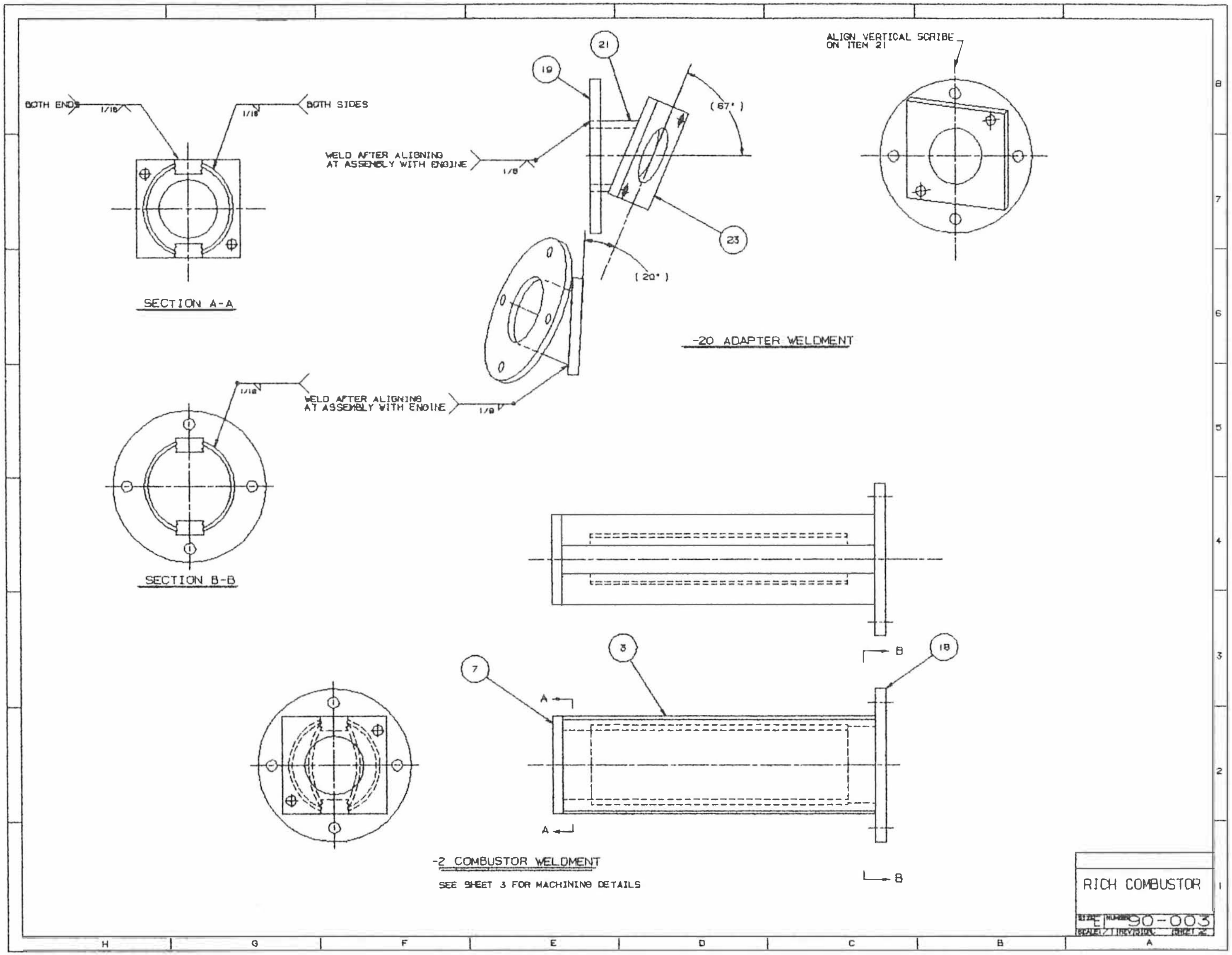
CONTRACT NO.
R01-1372-90

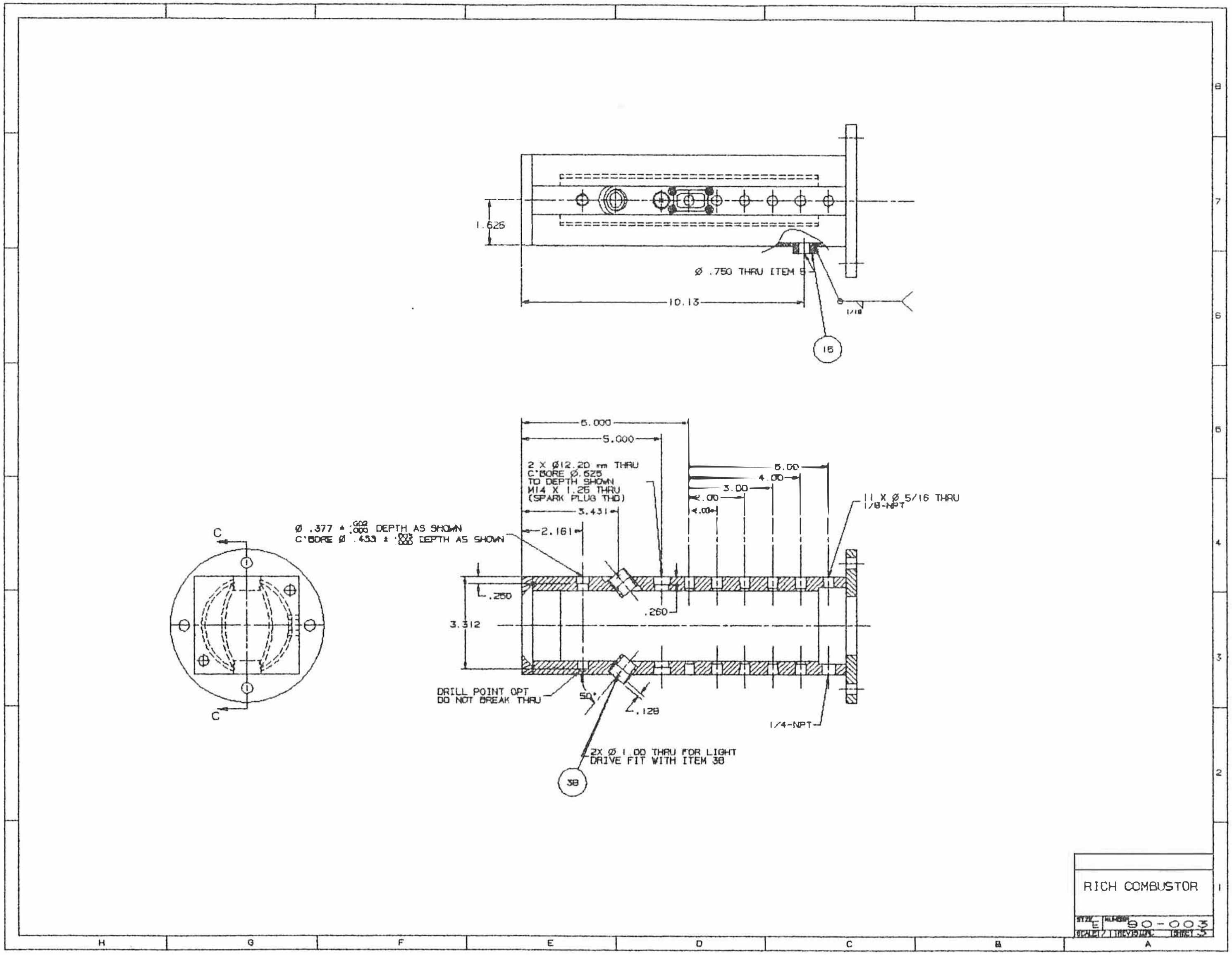
MATERIAL

PREP D K TRICK 11/88
CHKD D K TRICK 11/88
APVD
APVD

RICH COMBUSTOR

SIZE E DWG NUMBER 90-003
SCALE 1/1 REVISION SHEET 1 OF 5

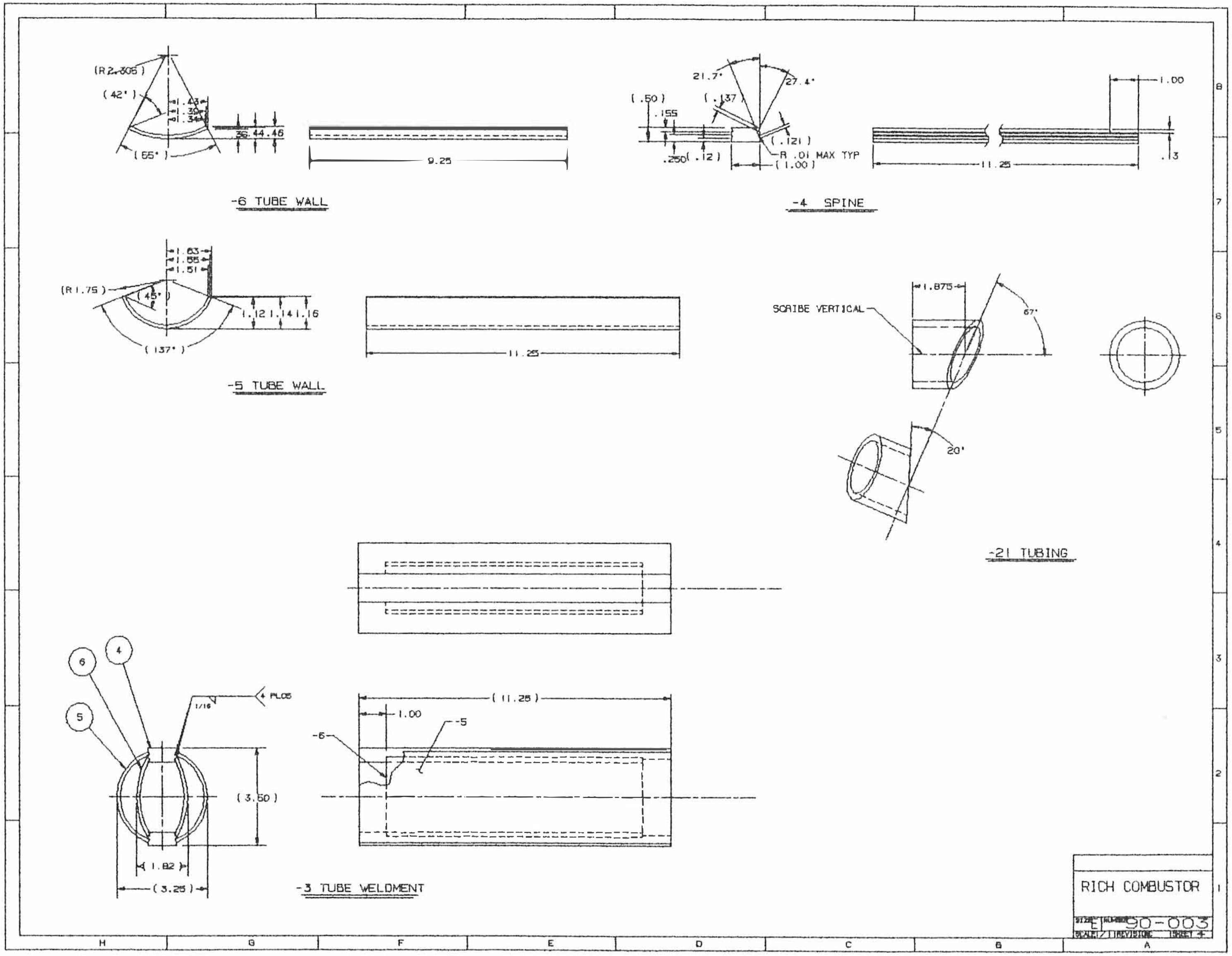

BOTH ENDS
1/16
1/16
BOTH SIDES
SECTION A-A
WELD AFTER ALIGNING AT ASSEMBLY WITH ENGINE
1/8
19
21
(67°)
23
(20°)
ALIGN VERTICAL SCRIBE ON ITEM 21
-20 ADAPTER WELDMENT
1/16
WELD AFTER ALIGNING AT ASSEMBLY WITH ENGINE
1/8
SECTION B-B
7
3
18
A
A
B
B
-2 COMBUSTOR WELDMENT
SEE SHEET 3 FOR MACHINING DETAILS
RICH COMBUSTOR
SIZE E
NUMBER 90-003
SHEET 2
H
G
F
E
D
C
B
A
8
7
6
5
4
3
2
1

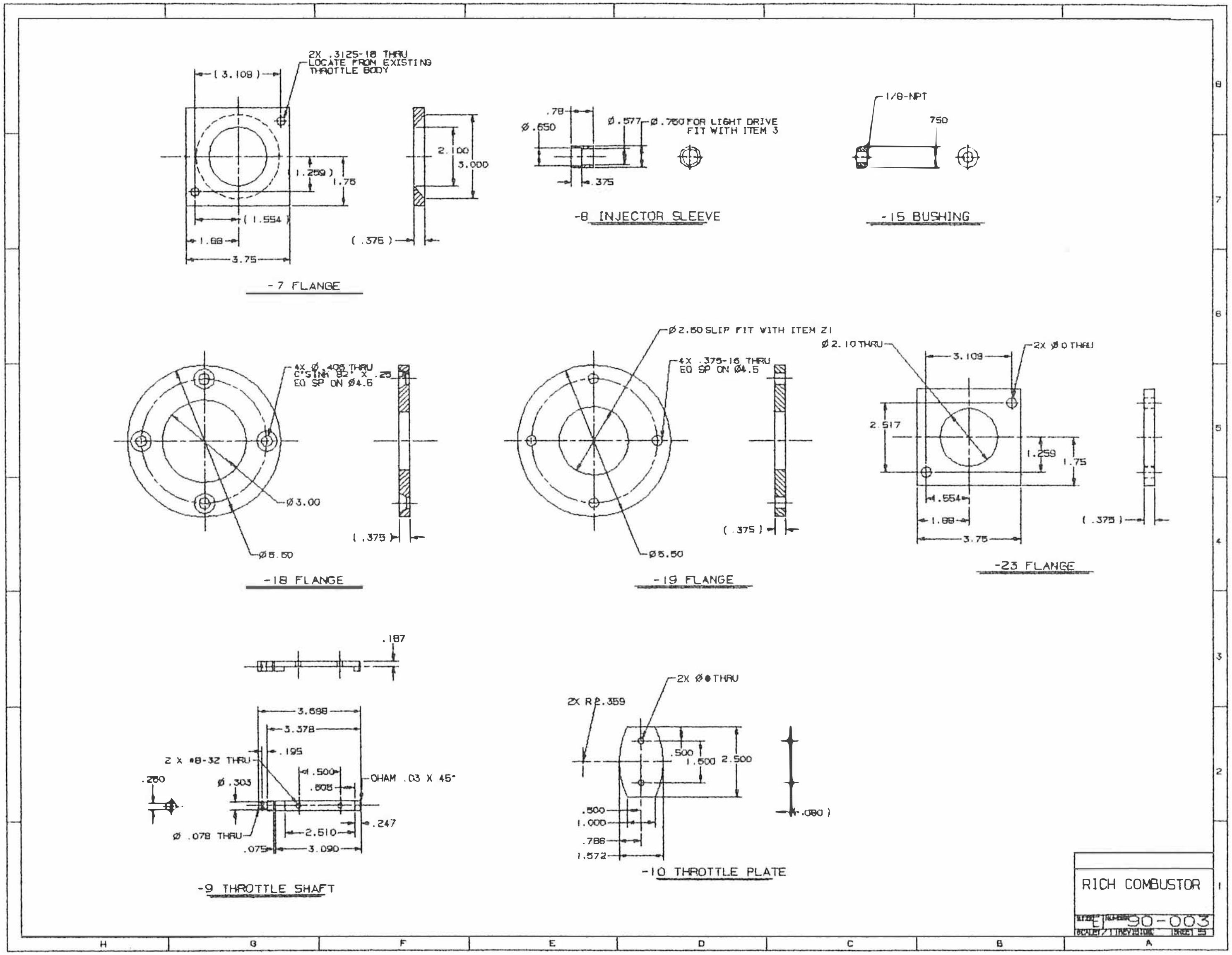


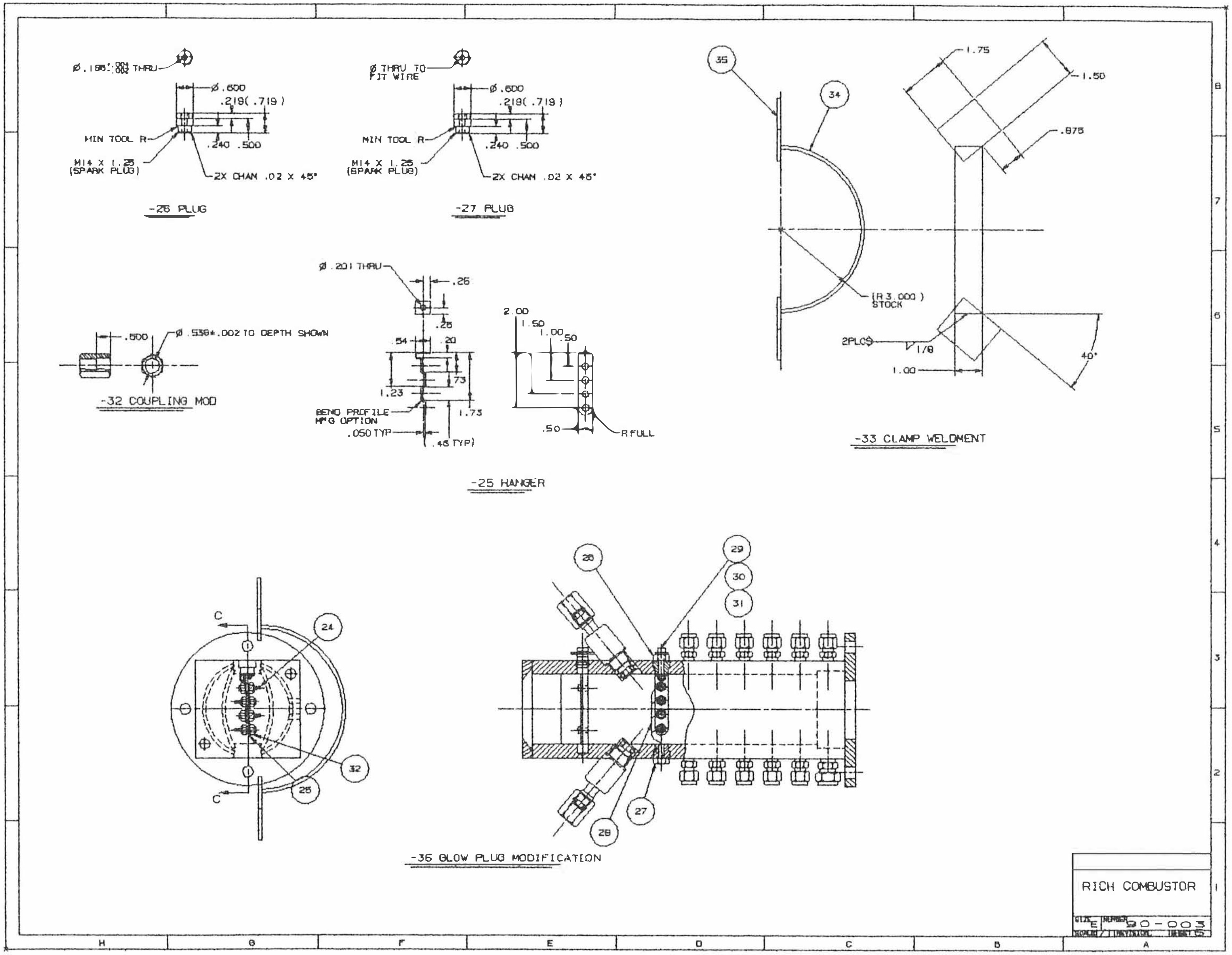
RICH COMBUSTOR

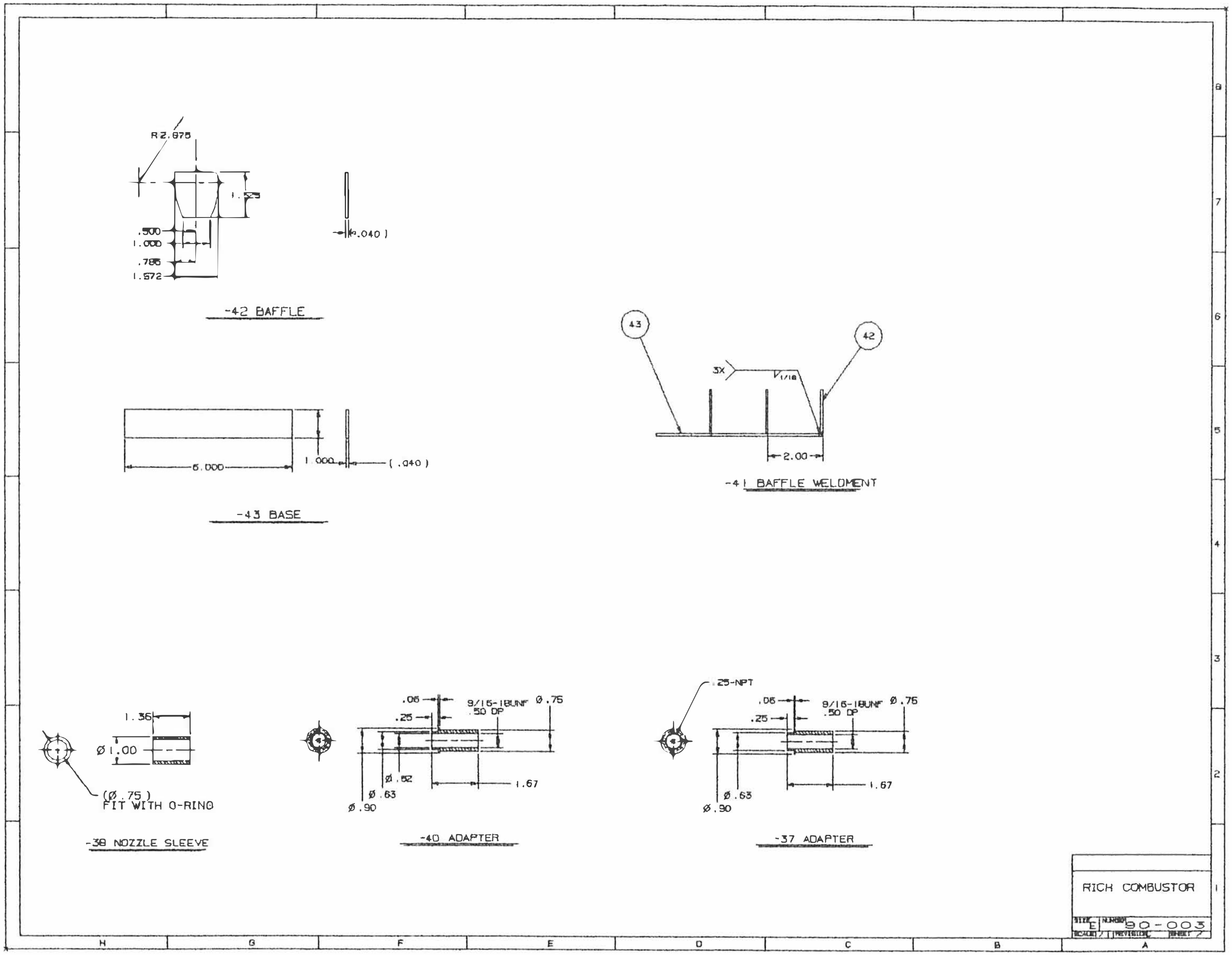
SIZE E NUMBER 90-003

SCALE 1 : 1 REVISION SHEET 3

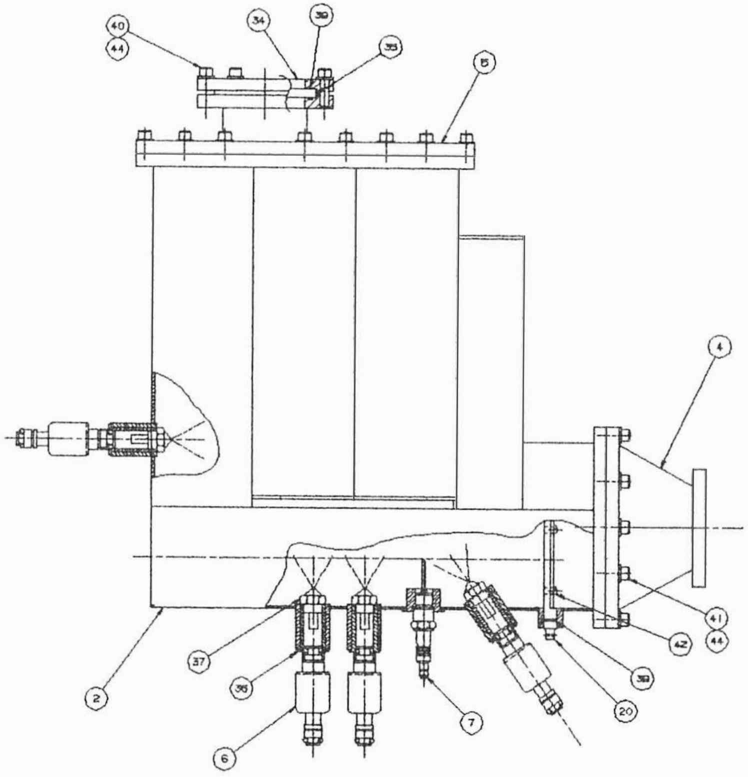

(R2.305)
(42°)
1.43
1.39
1.34
.36 .44 .46
(55°)
9.25
-6 TUBE WALL
21.7°
27.4°
(.50)
.155
(.137)
(.121)
.250
(.12)
R .01 MAX TYP
(1.00)
1.00
.13
11.25
-4 SPINE
1.63
1.55
1.51
(R1.75)
(45°)
1.12 1.14 1.16
(137°)
11.25
-5 TUBE WALL
1.875
SCRIBE VERTICAL
67°
20°
-21 TUBING
4
6
5
4 PLCS
1/16
(3.50)
1.82
(3.25)
(11.25)
1.00
-6
-5
-3 TUBE WELDMENT
RICH COMBUSTOR
SIZE E
NUMBER 90-003
SCALE 2/1
REVISION
SHEET 4
1 2 3 4 5 6 7 8
A B C D E F G H


2X .3125-18 THRU
LOCATE FROM EXISTING
THROTTLE BODY
(3.109)
(1.259)
1.75
(1.554)
1.88
3.75
2.100
3.000
(.375)
-7 FLANGE
.78
Ø.650
Ø.577
Ø.750 FOR LIGHT DRIVE
FIT WITH ITEM 3
.375
-8 INJECTOR SLEEVE
1/8-NPT
.750
-15 BUSHING
Ø2.50 SLIP FIT WITH ITEM 21
4X Ø.406 THRU
C'SINK 82° X .25
EQ SP ON Ø4.6
Ø3.00
Ø5.50
(.375)
-18 FLANGE
4X .375-16 THRU
EQ SP ON Ø4.5
Ø5.50
(.375)
-19 FLANGE
Ø2.10 THRU
3.109
2X Ø 0 THRU
2.517
1.259
1.75
1.554
1.88
3.75
(.375)
-23 FLANGE
.187
3.688
3.378
.195
2 X #8-32 THRU
.250
Ø.303
1.500
.808
CHAM .03 X 45°
.247
Ø .078 THRU
2.510
.075
3.090
-9 THROTTLE SHAFT
2X Ø THRU
2X R2.359
.500
1.500
2.500
.080
.500
1.000
.786
1.572
-10 THROTTLE PLATE
RICH COMBUSTOR
SIZE E
NUMBER 90-003
SCALE 2/1
REVISION
SHEET 5
H
G
F
E
D
C
B
A
1
2
3
4
5
6
7
8


Ø .188 +.001 -.002 THRU
Ø .600
.219(.719)
MIN TOOL R
.240 .500
M14 X 1.25
(SPARK PLUG)
2X CHAN .02 X 45°
-26 PLUG
Ø THRU TO
FIT WIRE
Ø .600
.219(.719)
MIN TOOL R
.240 .500
M14 X 1.25
(SPARK PLUG)
2X CHAN .02 X 45°
-27 PLUG
.500
Ø .538±.002 TO DEPTH SHOWN
-32 COUPLING MOD
Ø .201 THRU
.26
.26
.54
.20
.73
1.23
1.73
BEND PROFILE
MFG OPTION
.050 TYP
(.48 TYP)
2.00
1.50
1.00
.50
.50
R FULL
-25 HANGER
35
34
1.75
1.50
.875
(R 3.000)
STOCK
2PLCS
1/8
1.00
40°
-33 CLAMP WELDMENT
C
C
24
32
25
28
29
30
31
27
28
-36 GLOW PLUG MODIFICATION
RICH COMBUSTOR
SIZE E
NUMBER 90-003
SHEET 5
H
G
F
E
D
C
B
A
8
7
6
5
4
3
2
1


R 2.875
1.525
.500
1.000
.785
1.572
(.040)
-42 BAFFLE
6.000
1.000
(.040)
-43 BASE
43
42
3X
1/16
2.00
-41 BAFFLE WELDMENT
1.36
Ø 1.00
(Ø .75)
FIT WITH O-RING
-38 NOZZLE SLEEVE
.06
.25
9/16-18UNF
.50 DP
Ø .75
Ø .52
Ø .63
Ø .90
1.67
-40 ADAPTER
.25-NPT
.06
.25
9/16-18UNF
.50 DP
Ø .75
Ø .63
Ø .90
1.67
-37 ADAPTER
RICH COMBUSTOR
SIZE E
NUMBER 90-003
SCALE
REVISION
SHEET 7
H
G
F
E
D
C
B
A
8
7
6
5
4
3
2
1

Appendix C



-1 RICH COMBUSTOR ASSY

-5	-4	-3	-2	-1	PART NO	NAME	DESCRIPTION
				37	44	WASHER	1/4 FLAT WASHER STL
		2			43	DRAIN TUBE	3/8 OD X .08 WALL TUBING STL
				2	42	SCREW	#8 - 32 X 1/4 BUTTON HD SCREW STL
				31	41	SCREW	1/4 - 20 X 3/4 LG SHCS STL
				6	40	SCREW	1/4 - 20 X 7/8 LG SHCS STL
				2	39	O-RING	EXISTING
				1	38	O-RING	EXISTING
				4	37	NOZZLE	EXISTING
				1	36	ADAPTER	1" DIA BAR SST
				1	35	SIGHT GLASS	MCMASTER-CARR # 8483K23
				1	34	RETAINER	3/8 PLATE ALUM 6061-T6
				1	33	THROTTLE PLATE	.085 SHT STL AISI 1020 OR EQUIV
			1		32	SPACER	.085 SHT STL AISI 1020 OR EQUIV
			1		31	CAP	.085 SHT STL AISI 1020 OR EQUIV
			1		30	FLANGE	3/8 PLATE AISI 1020 OR EQUIV
			1		29	CAP	.085 SHT STL AISI 1020 OR EQUIV
			1		28	TUBE	2 X 2 X .085 WELDED TUBING STL
			1		27	TUBE	2 X 2 X .085 WELDED TUBING STL
			1		26	TUBE	2 X 3 X .085 WELDED TUBING STL
			1		25	TUBE	2 X 3 X .085 WELDED TUBING STL
			1		24	TUBE	2 X 3 X .085 WELDED TUBING STL
			1		23	TUBE	2 X 3 X .085 WELDED TUBING STL
			1		22	THROTTLE BUSHING	3/4 ROD AISI 1020 OR EQUIV
			1		21	THROTTLE BUSHING	3/4 ROD AISI 1020 OR EQUIV
				1	20	THROTTLE SHAFT	3/8 DRILL ROD
			1		19	PLUG BLOCK	3/4 PLATE AISI 1020 OR EQUIV
			4		18	NOZZLE SLEEVE	1" DIA BAR AISI 1020 OR EQUIV
			2		17	DRAIN CAP	.085 SHT STL AISI 1020 OR EQUIV
			2		16	DRAIN	2 X 2 X .085 WELDED TUBING STL
	2				15	TUBE HALF	2 X 3 X .085 WELDED TUBING STL
1					14	PIPE	2.5 OD X .085 WALL TUBING STL
1					13	FLANGE	3/8 PLATE AISI 1020 OR EQUIV
	1				12	FLANGE	3/8 PLATE AISI 1020 OR EQUIV
	1				11	FLANGE	3/8 PLATE AISI 1020 OR EQUIV
			1		10	FLANGE	3/8 PLATE AISI 1020 OR EQUIV
1					9	COVER PLATE	3/8 PLATE AISI 1020 OR EQUIV
			1		8	FLANGE	3/8 PLATE AISI 1020 OR EQUIV
				2	7	SPARK PLUG	EXISTING
				4	6	INJECTOR	EXISTING
☒				1	5	COVER WELDMENT	
	☒			1	4	ADAPTER WELDMENT	
		☒	1		3	TUBE WELDMENT	
			☒	1	2	COMBUSTOR WELDMENT	
				☒	1	RICH COMBUSTOR ASSEMBLY	
NO. REQD							

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		
.XX	.XXX	ANGLES
± .03	± .010	± 1/2°

MATERIAL

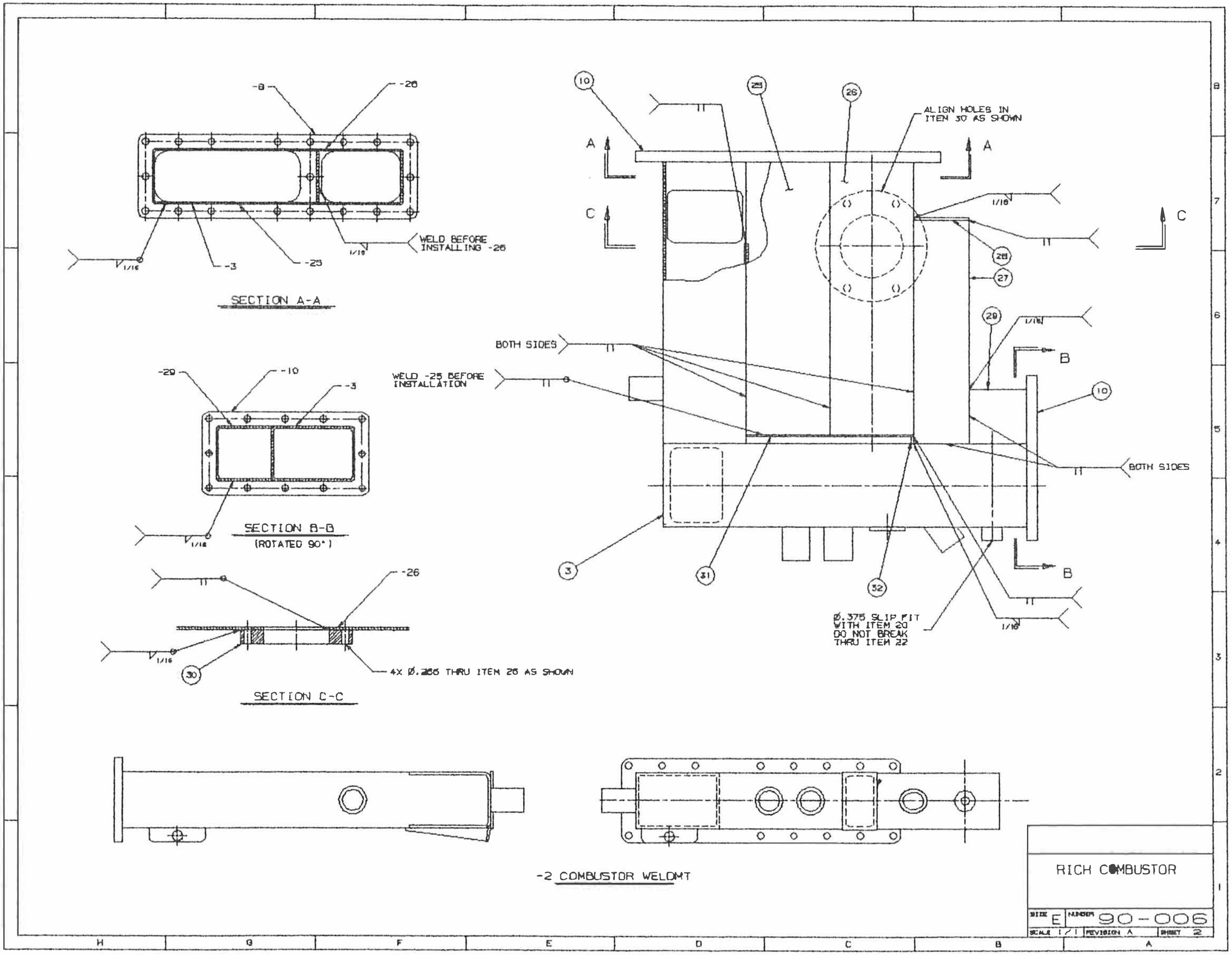
CONTRACT No.
R01-1372-90

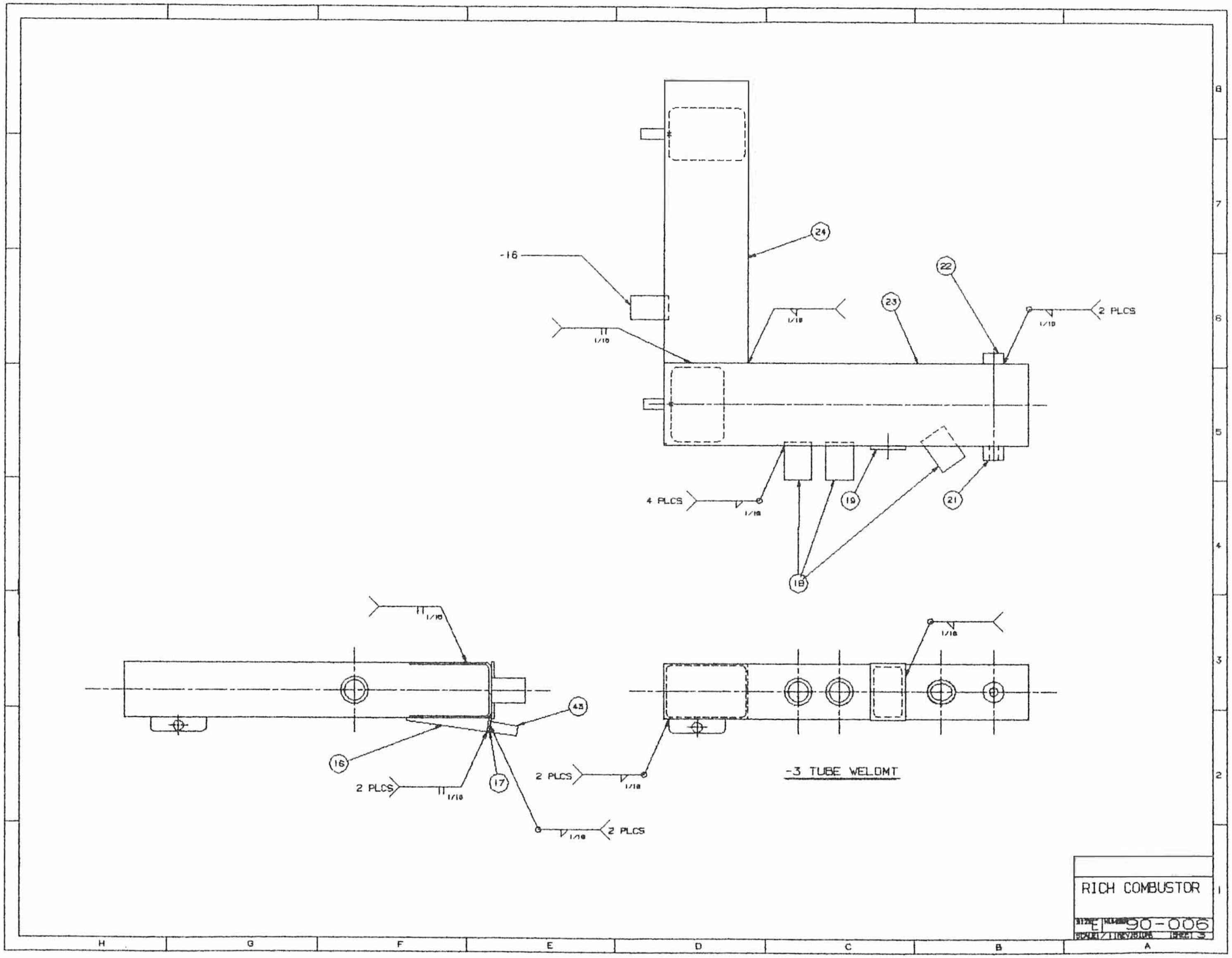
PRO#	D K IRICK	10/98
CHKR	DK IRICK	10/98
APVD		
APVD		

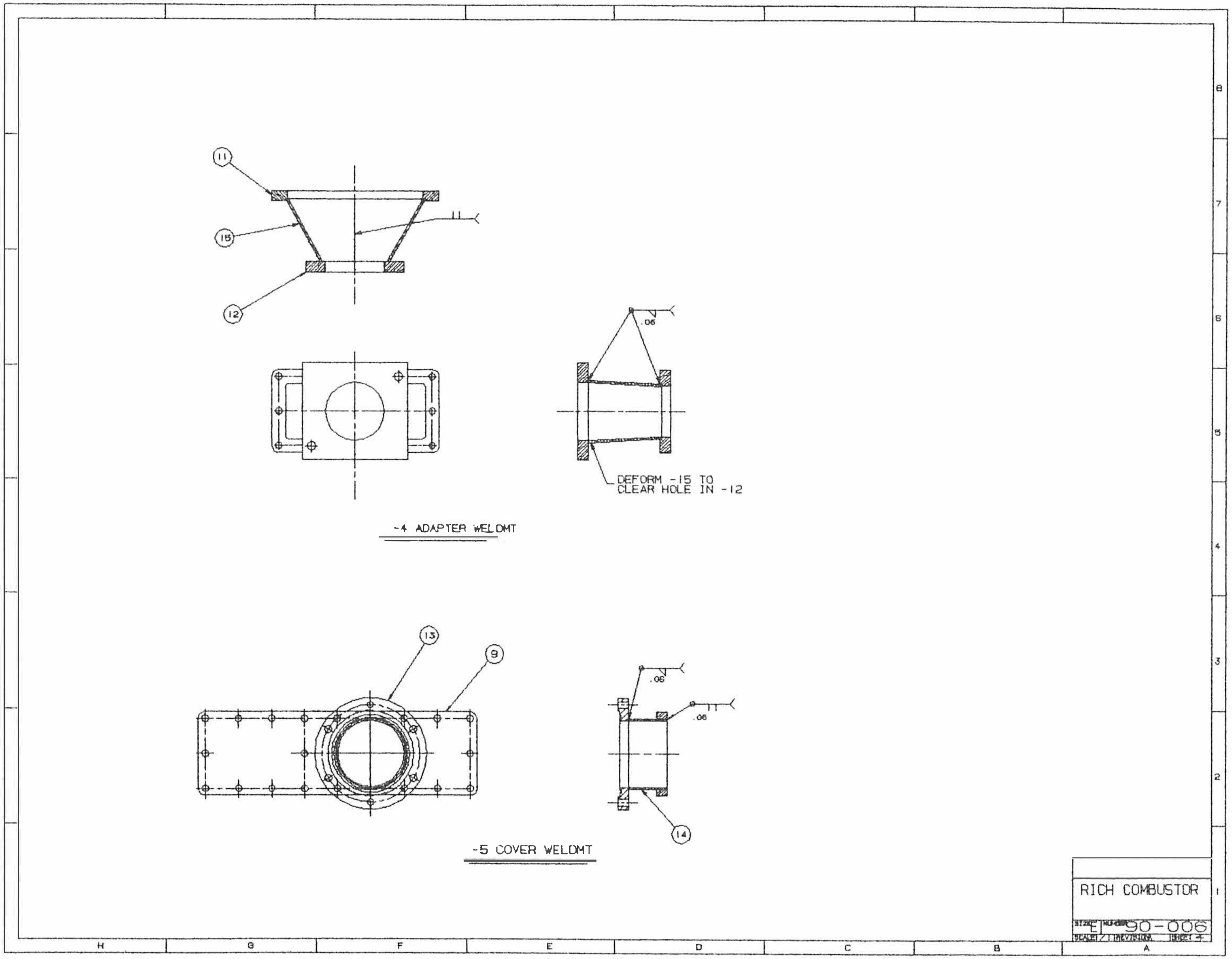
RICH COMBUSTOR

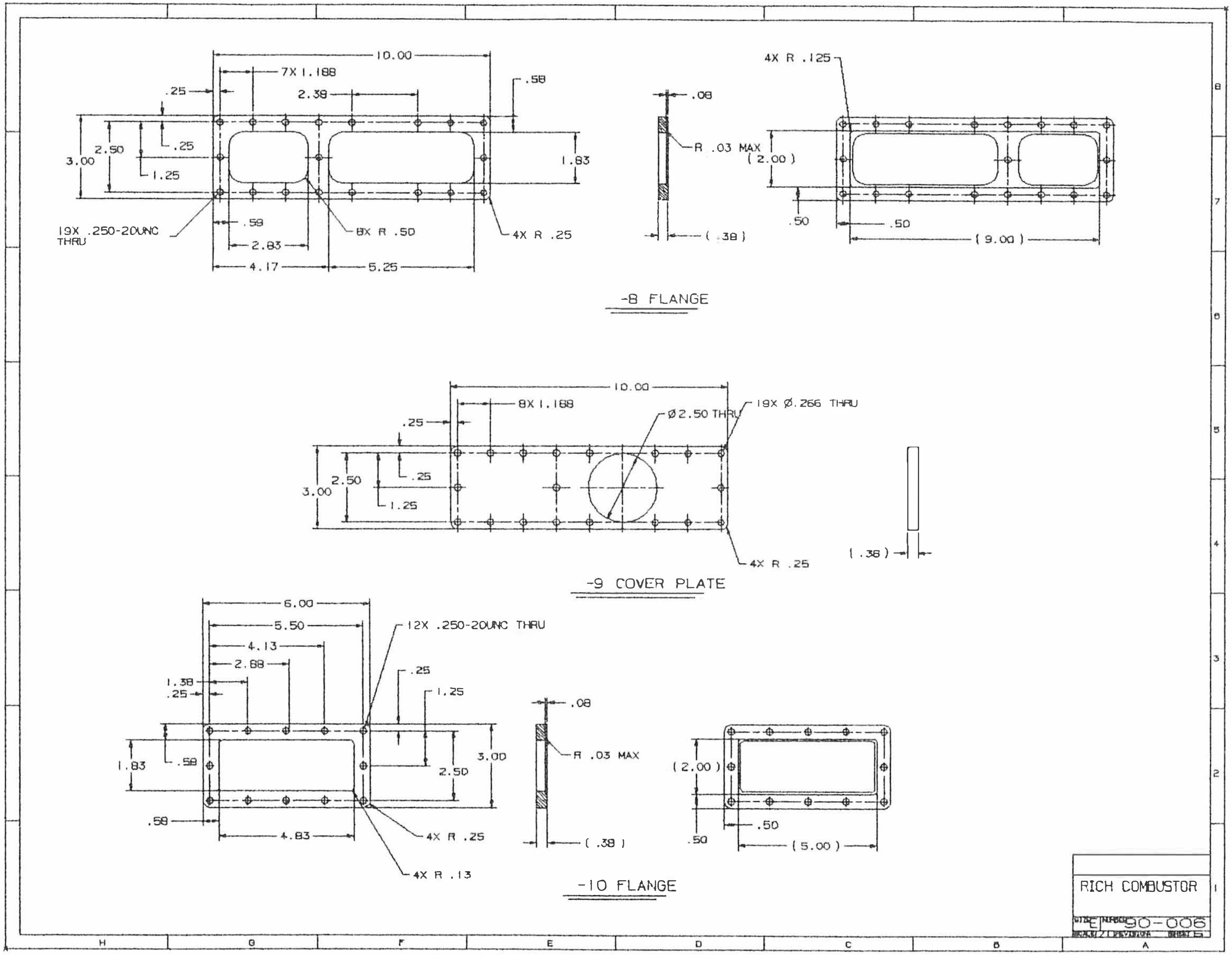
SIZE E NUMBER 90-006

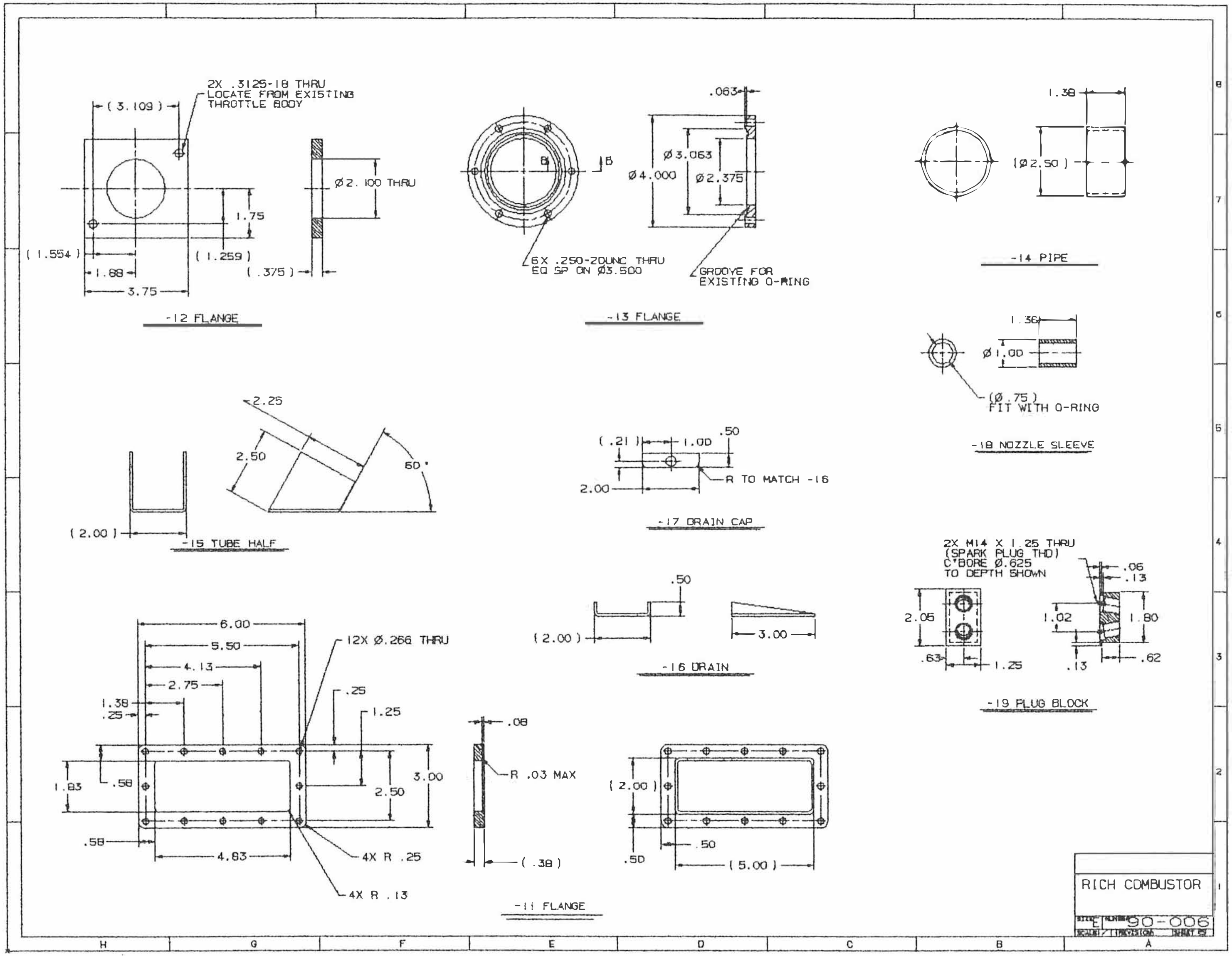
SCALE 1/1 REVISION A SHEET 1 OF 9

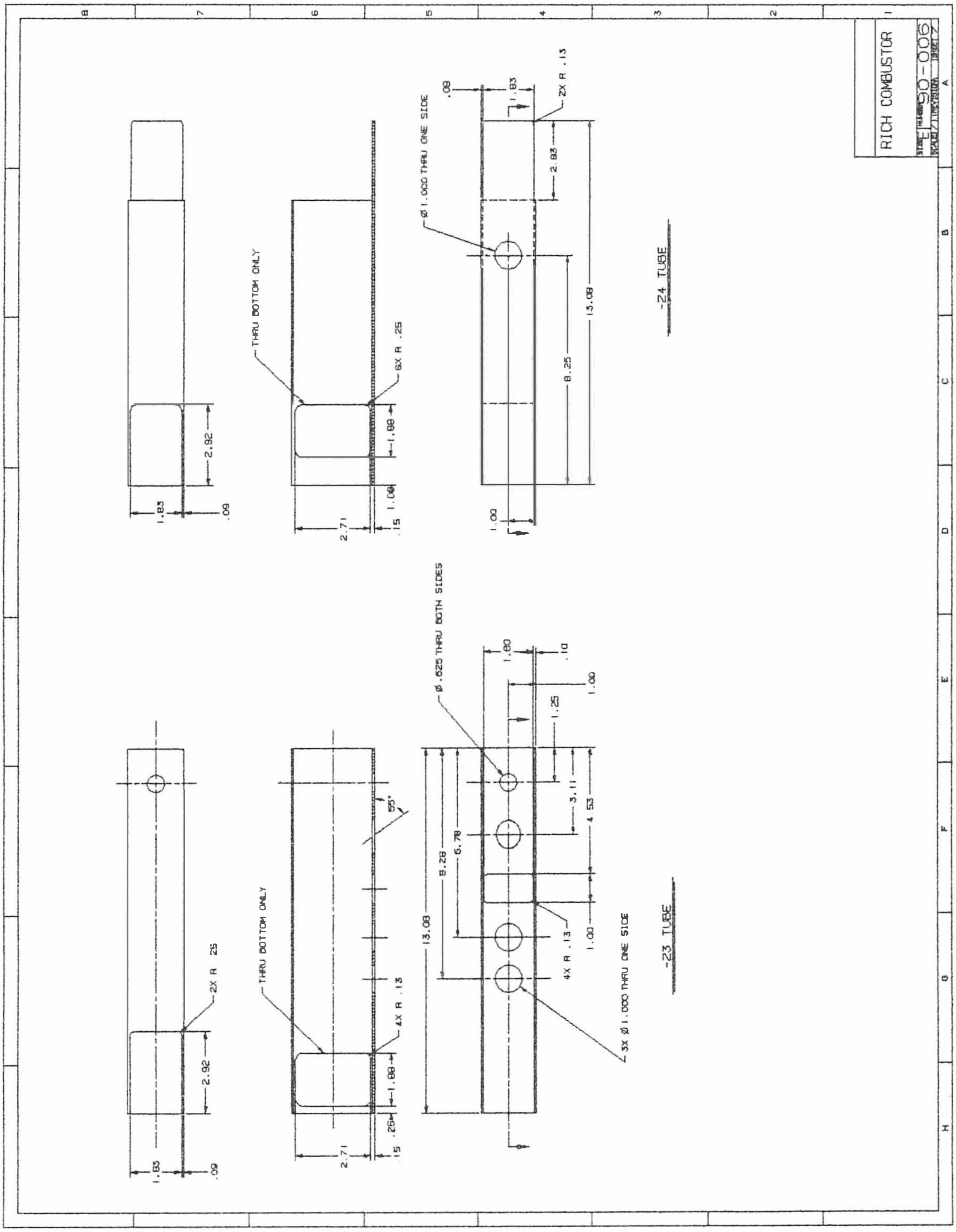

-8
-26
WELD BEFORE INSTALLING -26
1/16
1/16
-3
-25
SECTION A-A
-29
-10
-3
1/16
SECTION B-B
(ROTATED 90°)
-26
1/16
30
4X Ø.266 THRU ITEM 26 AS SHOWN
SECTION C-C
10
25
26
ALIGN HOLES IN ITEM 30 AS SHOWN
A
A
C
C
1/16
28
27
29
1/16
B
B
10
BOTH SIDES
BOTH SIDES
WELD -25 BEFORE INSTALLATION
3
31
32
Ø.375 SLIP FIT WITH ITEM 20 DO NOT BREAK THRU ITEM 22
1/16
-2 COMBUSTOR WELDMT
RICH COMBUSTOR
SIZE E
NUMBER 90-006
SCALE 1/1
REVISION A
SHEET 2
H
G
F
E
D
C
B
A
8
7
6
5
4
3
2
1

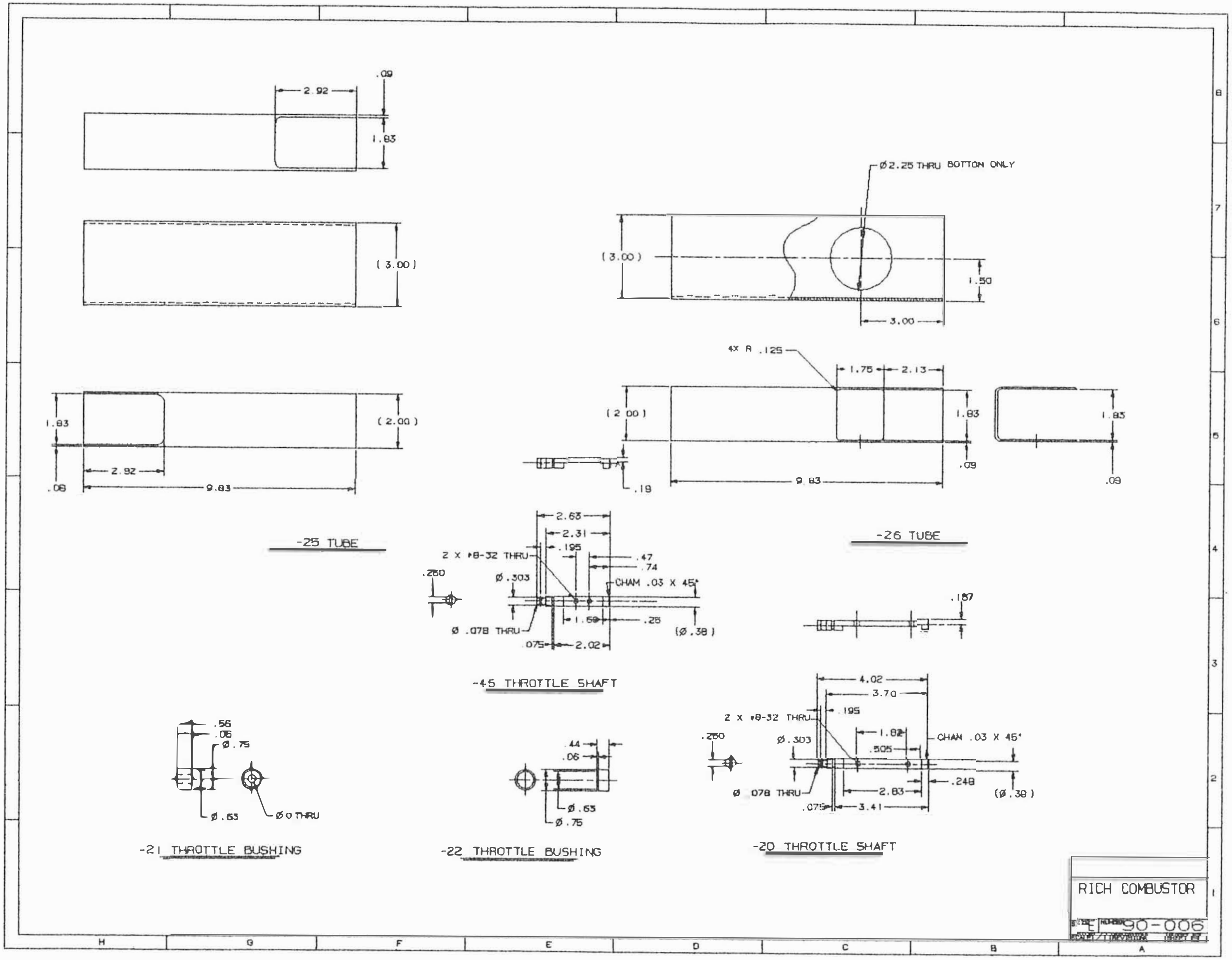

-16
24
22
23
2 PLCS
1/16
1/16
1/16
4 PLCS
1/16
19
21
18
1/16
1/16
-3 TUBE WELDMT
43
16
2 PLCS
1/16
17
2 PLCS
1/16
2 PLCS
1/16
RICH COMBUSTOR
SIZE E
NUMBER 90-006
SCALE 1/1 REVISION
SHEET 3
A B C D E F G H
1 2 3 4 5 6 7 8

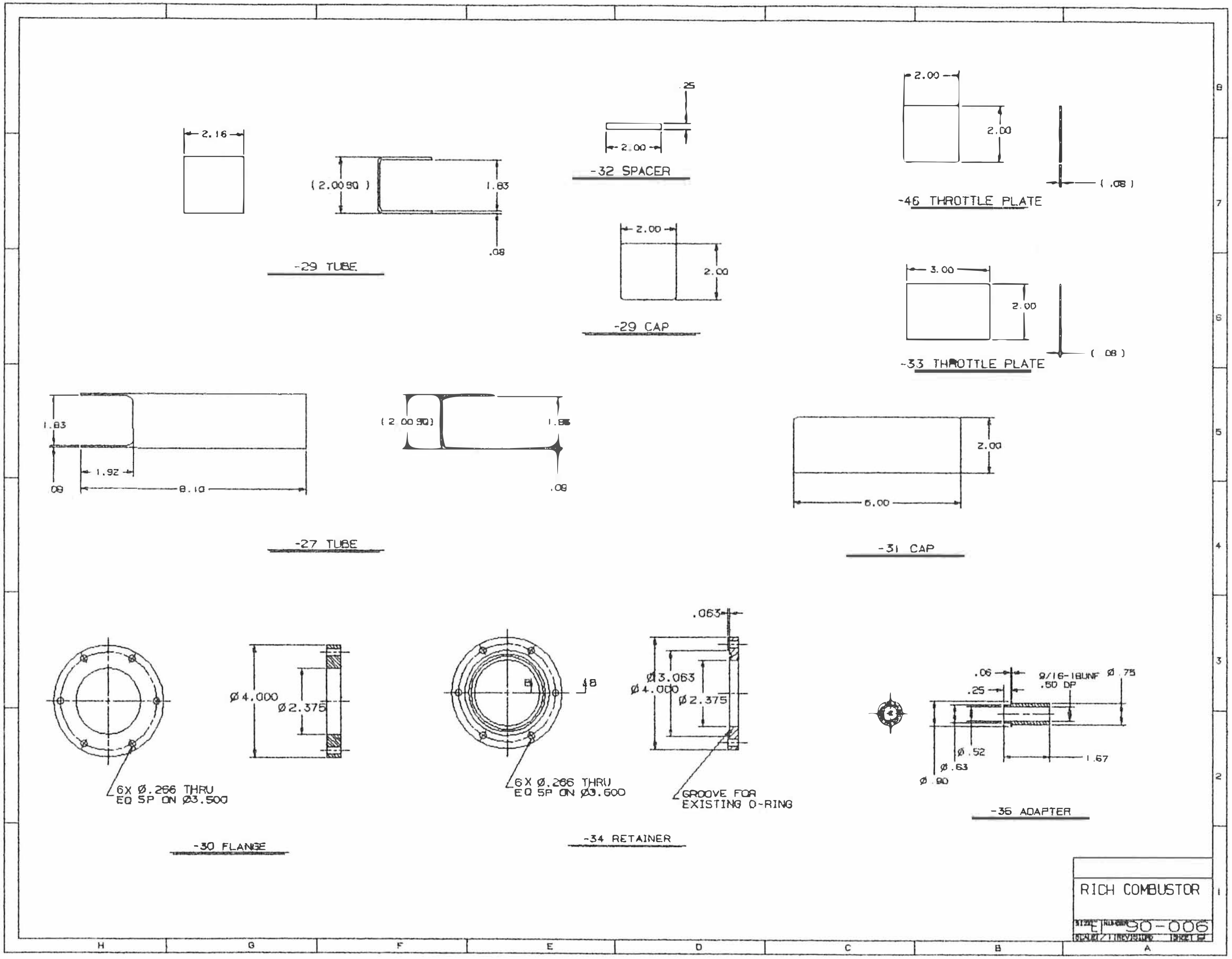

11
15
12
.06
DEFORM -15 TO
CLEAR HOLE IN -12
-4 ADAPTER WELDMT
13
9
.06
.06
14
-5 COVER WELDMT
RICH COMBUSTOR
SIZE E
NUMBER 90-006
SCALE 2:1
REVISION
SHEET 4
H
G
F
E
D
C
B
A
8
7
6
5
4
3
2
1

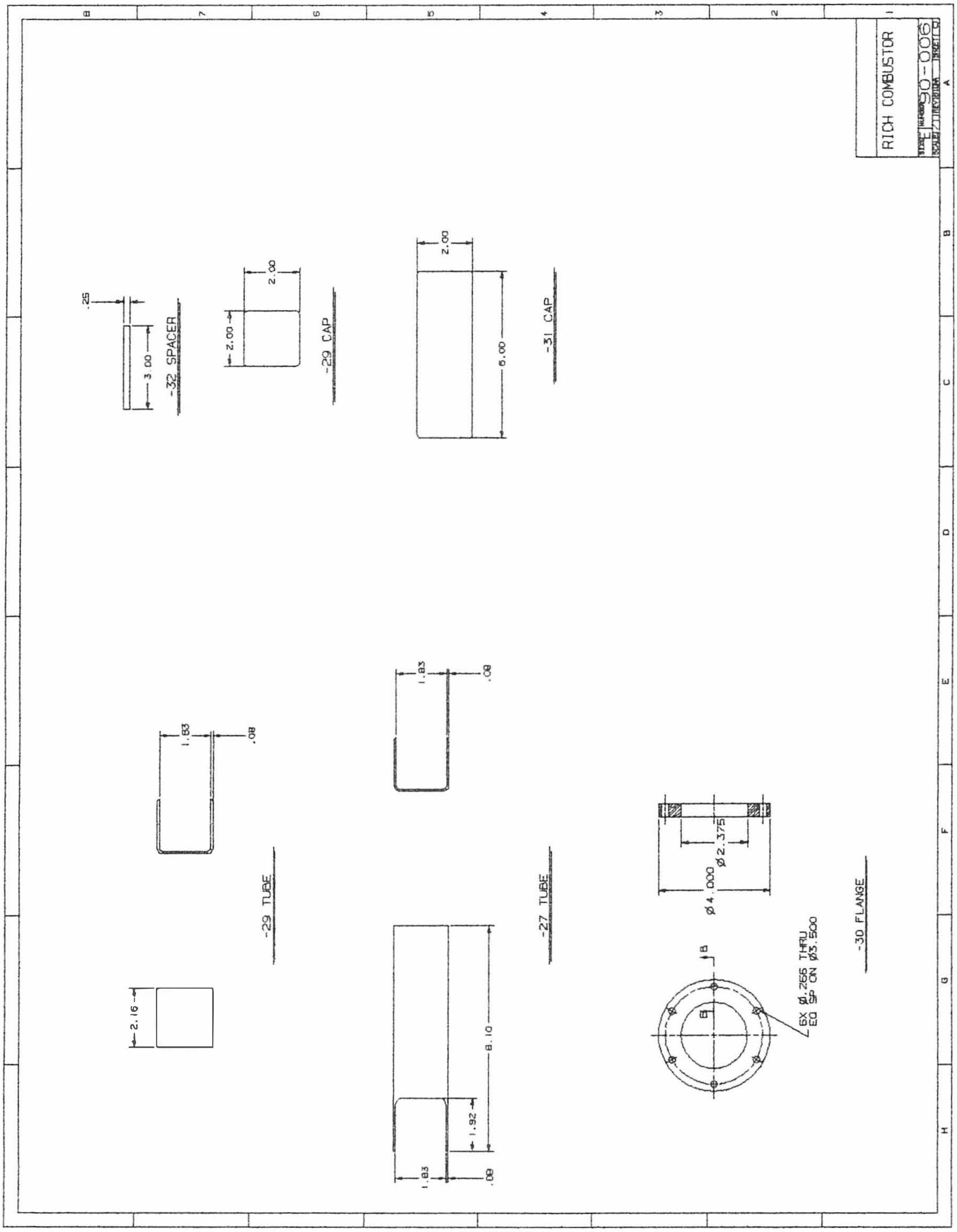

10.00
7X 1.188
.25
2.38
.58
.25
2.50
3.00
1.25
1.83
19X .250-20UNC THRU
.58
2.83
8X R .50
4X R .25
4.17
5.25
.08
R .03 MAX
(.38)
4X R .125
(2.00)
.50
.50
(9.00)
-8 FLANGE
10.00
8X 1.188
Ø2.50 THRU
19X Ø.266 THRU
.25
2.50
3.00
.25
1.25
4X R .25
(.38)
-9 COVER PLATE
6.00
5.50
12X .250-20UNC THRU
4.13
2.88
1.38
.25
.25
1.25
.08
1.83
.58
3.00
2.50
R .03 MAX
(2.00)
.58
4.83
4X R .25
(.38)
.50
.50
(5.00)
4X R .13
-10 FLANGE
RICH COMBUSTOR
SIZE E
NUMBER 90-006
SHEET 5
H
G
F
E
D
C
B
A
8
7
6
5
4
3
2
1


2X .3125-18 THRU
LOCATE FROM EXISTING
THROTTLE BODY
(3.109)
Ø2.100 THRU
1.75
(1.554)
(1.259)
1.88
3.75
(.375)
-12 FLANGE
.063
B
B
Ø3.063
Ø4.000
Ø2.375
6X .250-20UNC THRU
EQ SP ON Ø3.500
GROOVE FOR
EXISTING O-RING
-13 FLANGE
1.38
(Ø2.50)
-14 PIPE
1.36
Ø1.00
(Ø.75)
FIT WITH O-RING
-18 NOZZLE SLEEVE
2.25
2.50
60°
(2.00)
-15 TUBE HALF
(.21)
1.00
.50
R TO MATCH -16
2.00
-17 DRAIN CAP
.50
(2.00)
3.00
-16 DRAIN
2X M14 X 1.25 THRU
(SPARK PLUG THD)
C'BORE Ø.625
TO DEPTH SHOWN
.06
.13
2.06
1.02
1.80
.63
1.25
.13
.62
-19 PLUG BLOCK
6.00
5.50
4.13
2.75
1.38
.25
12X Ø.266 THRU
.25
1.25
1.83
.58
3.00
2.50
.58
4.83
4X R .25
4X R .13
.08
R .03 MAX
(.38)
(2.00)
.50
.50
(5.00)
-11 FLANGE
RICH COMBUSTOR
SIZE E
NUMBER 90-006
SCALE
REVISION
SHEET
8
7
6
5
4
3
2
1
H
G
F
E
D
C
B
A


RICH COMBUSTOR
E 90-006
-24 TUBE
-23 TUBE
Ø 1.000 THRU ONE SIDE
THRU BOTTOM ONLY
6X R .25
2X R .13
Ø .625 THRU BOTH SIDES
3X Ø 1.000 THRU ONE SIDE
4X R .13
2X R .25
55°
1.83
.09
2.92
2.71
.15
1.06
1.88
.26
.08
2.83
8.25
13.08
1.00
1.80
.10
1.25
3.11
4.53
6.78
9.28
A
B
C
D
E
F
G
H
1
2
3
4
5
6
7
8


2.92
.09
1.83
(3.00)
1.83
(2.00)
2.92
.08
9.83
-25 TUBE
Ø2.25 THRU BOTTOM ONLY
(3.00)
1.50
3.00
4X R .125
1.75
2.13
(2.00)
1.83
1.83
.09
.19
9.83
.09
-26 TUBE
2.63
2.31
.195
2 X #8-32 THRU
.47
.74
.280
Ø .303
CHAM .03 X 45°
1.59
.25
Ø .078 THRU
(Ø .38)
.075
2.02
-45 THROTTLE SHAFT
.187
4.02
3.70
.195
2 X #8-32 THRU
.280
Ø .303
1.82
.505
CHAM .03 X 45°
.248
Ø .078 THRU
2.83
(Ø .38)
.075
3.41
-20 THROTTLE SHAFT
.56
.06
Ø .75
Ø .63
Ø 0 THRU
-21 THROTTLE BUSHING
.44
.06
Ø .63
Ø .75
-22 THROTTLE BUSHING
RICH COMBUSTOR
E
90-006
H G F E D C B A
8 7 6 5 4 3 2 1


2.16
(2.00 SQ)
1.83
.08
-29 TUBE
.25
2.00
-32 SPACER
2.00
2.00
-29 CAP
2.00
2.00
(.08)
-46 THROTTLE PLATE
3.00
2.00
(.08)
-33 THROTTLE PLATE
1.83
.08
1.92
8.10
-27 TUBE
(2.00 SQ)
1.85
.08
2.00
6.00
-31 CAP
Ø4.000
Ø2.375
6X Ø.266 THRU
EQ SP ON Ø3.500
-30 FLANGE
.063
B
B
Ø3.063
Ø4.000
Ø2.375
6X Ø.266 THRU
EQ SP ON Ø3.500
GROOVE FOR
EXISTING O-RING
-34 RETAINER
.06
.25
9/16-18UNF Ø.75
.50 DP
Ø.52
Ø.63
Ø.90
1.67
-35 ADAPTER
RICH COMBUSTOR
SIZE E
NUMBER 90-006
SCALE 2
REVISION
SHEET
H
G
F
E
D
C
B
A
1
2
3
4
5
6
7
8


RICH COMBUSTOR
90-006
A
-32 SPACER
.25
3.00
-29 CAP
2.00
2.00
-31 CAP
2.00
8.00
-29 TUBE
1.83
.08
2.16
-27 TUBE
1.83
.08
8.10
1.92
1.83
.08
-30 FLANGE
Ø4.000
Ø2.375
6X Ø.266 THRU
EQ SP ON Ø3.500
B

Appendix D

Dynamometer

General Electric Type TLC-3544 DC Dynamometer
300 hp absorption
250 hp motoring
2500-5000 RPM
Model # 42G306aj
Serial # AN-2-6
General Electric Company
Schenectady, NY 12345

Gas Analyzer

PGA-5 Five Gas Infrared Analyzer
Range and Accuracy:

HC	0-5000 ppm	0-400 ± 12 ppm >400 ± 3% of reading
CO	0-10%	0-2 ± .06 >2 ± .03
CO_2	0-25%	0-16 ± .5 >16 ± .6% of reading
O_2	0-25%	0-16 ± .5 >16 ± .6% of reading
NO_x	0-4000 ppm	0-1000 ± 32 ppm 1000-2000 ± 60 ppm 2000-4000 ± 120 ppm

MicroProcessor Systems Inc
Sterling Heights, MI

Engine Controller

TEC-II PAF
Serial # 202776-34T2H-6CYL
Electromotive
Chantilly, VA

Refrigerated Enclosure

Refrigeration Unit

Kold Pack Low Temprature Split Sytsem
Model # 303FSSE
14000 Btu/hr
Kold Pack Refrigeration Systems
Hackson, MI 49201

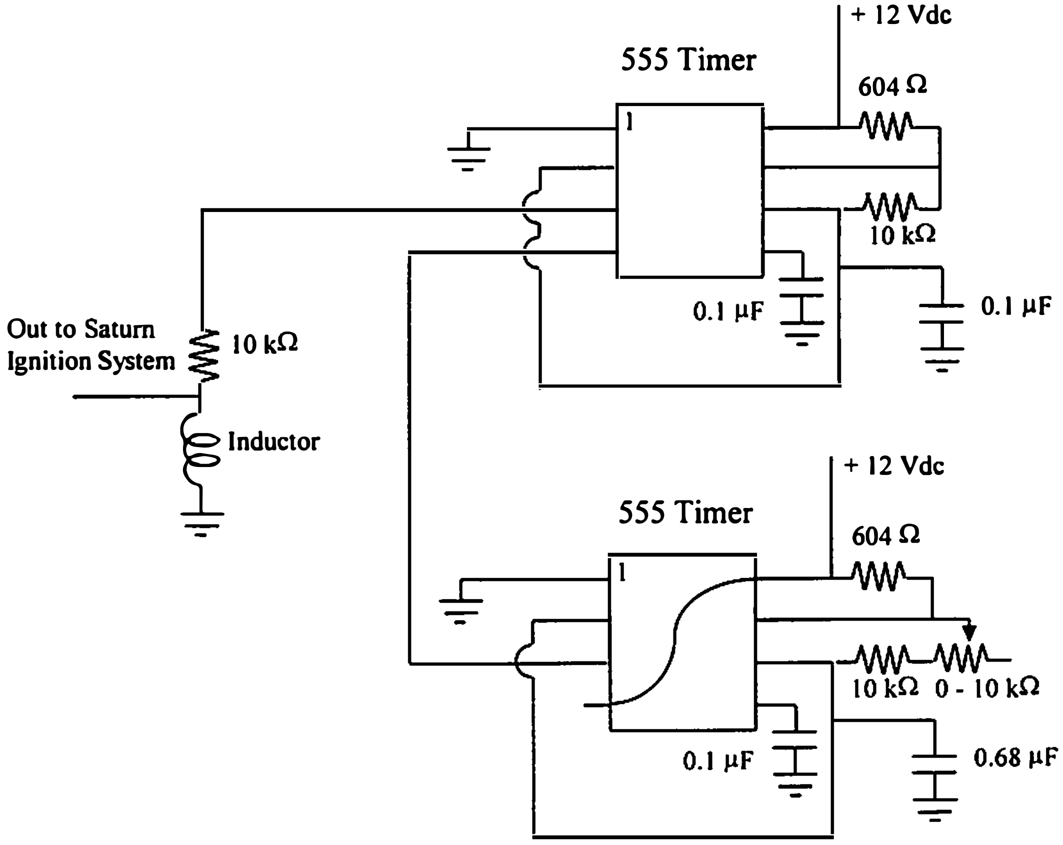
Enclosure

10' X 8' X 12' with 12" thick walls, floor and roof
Frame construction on site
Insulation: 12" Styrofoam (k=0.03 Btu/hr-ft-^{o}F)
Heated window
CO_2 fire supression system

Solid State Relay

Caydon Solid State Relay
Model D1D40
Line voltage: 0 - 100 Vdc
Load current range: 0 -40 Adc
Control voltage range: 3.5 - 32 Vdc
Control current @ 5Vdc: 1.6 mA
Must release voltage: 1.0 Vdc
Surge currend 10 ms: 106 A peak

In-house Developed Ignition Circuit



Data Acquisition System

Flight Recorder
In-Vehicle Data Acquisition System
Advance Electornic Diagnostics, Inc
10850 N. 24th Ave., Suite 101
Pheonix, AZ 85029

External Input Channels
Analog - 15 differential inputs. Cold junction compensation is handled internally. 12 bits resolution.
Serial Logging - 128 channels of RS-232 levels. Maximum baud rate of 131 kBaud.
Timer/Discrete Inputs - Frequency 0.2 Hz to 100 kHz. Pulsewidth - 10 microseconds to 5 seconds.
Sampling - 500 microseconds to 1 hour time between storage.

VITA

David Irick was born in Knoxville, Tennessee September 22, 1955. He attended public school in Knox County, Tennessee and graduated from Halls High School in June 1972. He enrolled at the University of Tennessee the following September and graduated in August 1980 with a Bachelor of Science degree in Mechanical Engineering. After working as an engineer in industry for twelve years, he returned to the University of Tennessee for graduate study while continuing his career as an engineer. David received the Master of Science degree in May 1995 and the Doctor of Philosophy degree in December 1997, both in Mechanical Engineering.

He is currently a Research Assistant Professor in the Department of Mechanical and Aerospace Engineering and Engineering Science at the University of Tennessee.