An Integrated Geometric and Thermodynamic Performance Model of the 2.670 Stirling Engine

By
Munhee Sohn

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2004

Signature of Author: ______________________________
Department of Mechanical Engineering
May 7, 2004

Certified by: ______________________________
David Wallace
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by: ______________________________
Ernest G. Cravalho
Professor of Mechanical Engineering
Chairman, Undergraduate Thesis Committee

An Integrated Geometric and Thermodynamic Performance Model of the 2.670 Stirling Engine

By
Munhee Sohn

Submitted to the Department of Mechanical Engineering
On May 7, 2004 in Partial Fulfillment of the
Requirements for the Degree of Bachelor of Science in
Mechanical Engineering

Abstract

2.670 is a required mechanical engineering class taught during the Independent Activities Period (IAP) at MIT in which each student constructs a Stirling Engine. For the most part, all of the engine parts are uniform, but if so desired, students are allowed to make design changes to certain parts in order to compete for the fastest engine at the end of the class. The research team in the MIT CADlab is working on an environment, called DOME, which makes it easy to link together simulations in different packages to perform integrated analysis and make them operable over the Internet.

An integration environment has been created as a DOME project in which students can analyze and optimize the design of the 2.670 Stirling Engine. A thermodynamics model of the engine was created in Matlab and a parametric solid model was created in SolidWorks. Then, DOME was used to link the Matlab thermodynamic models to the Solidworks cad model so that when geometric parameters are changed one can see how this will affect engine performance. Students will be allowed to change the diameter and length of the displacer piston and see how it affects the work per cycle of the engine.

In general, DOME was easy to learn how to use and the capabilities of web accessibility and the speed of design analysis and optimization was impressive. The future intention is that 2.670 students could use this integration environment to better analyze the 2.670 Stirling Engine.

Thesis Supervisor: Dave Wallace
Title: Professor of Mechanical Engineering

Acknowledgements

I would like to thank all the people who made this thesis possible. First, I thank my thesis advisor, Professor Wallace, for providing me with this project in his CADlab and for his continued support and guidance throughout my endeavors. I would like to thank the graduate students in the CADlab: Sittha Sukkasi for teaching me how to build a matlab wrapper and answering any questions I had regarding DOME, Wei Mao for helping me with the Solidworks wrapper and final integration project and solving problems associated with it, Elaine Yang for debugging my Matlab file and teaching me the inner-workings of DOME, Qing Cao for setting up my computer so that it was updated with all the newest programs and setting up an account for me, Jacob Wronski for answering any general questions I had about DOME and SolidWorks. I would also like to thank Bill Fienup for helping me debug my parametric SolidWorks model and answering any questions I had about SolidWorks. Finally, I would like to thank my advisor, Professor Sang Kim, for his continued support and encouragement in both my academic and personal goals at MIT.

Index

List of Figures

1.0 Goal

The goal of this thesis is to develop geometric and thermodynamic models of the 2.670 Stirling engine and link them together in a integrated manner so that parametric changes to the engine geometry seamlessly drives affected performance simulations. The work will involve reusing, building, and linking Matlab and SolidWorks models describing different aspects of the engine. The future goal is to make these models available online to be used by future 2.670 students.

2.0 Introduction

Design in engineering is a cooperative effort which involves the integration of numerous designers and various tools. Thus, in an un-integrated system model, the product design cycles can be very long. Because the system is un-integrated, any changes made in the design chain take a long time to propagate and slow down the evaluation of the effects of the changes. Oftentimes flaws are not discovered until the end of the design cycle and are very costly to fix. Therefore, predictive integrated system modeling is becoming a more important and urgent issue as companies seek to produce high-quality complex products while using fewer resources [1,2,3]. In the case of 2.670, a predictive integrated system will allow students to quickly explore parametric changes and see how they affect the engine efficiency.

Integrated design can reduce the design evaluation cycle time dramatically, allowing the effects of the design changes to be seen easily and quickly. This permits hundreds of design cycles and furthermore, errors caused by design changes can be caught immediately [1].

3.0 Background Information

3.1 DOME

DOME stands for distributed objected-based modeling environment. DOME is a system integration solution that has been produced by the research team in the MIT CADlab. It is currently a software prototype that allows experts to integrate the services of models they have created using familiar tools to form large distributed simulations [1].

3.1.1 The Solution

In addition to a number of other unique characteristics, DOME provides the integration functionality and multiple user web-like access issues that are needed for this application, as mentioned in section 2.2.

3.1.2 How DOME Works

The way DOME works can be explained by 3 sets of users: the experts who define the interface, the administrators who publish DOME models, and the users who run the design simulations. The experts create and design the models, determine user access, and specify input and output options of the model [1]. They log into a DOME model server by a web browser and use special wrapper objects to make their published services accessible via the internet. A wrapper, written as a software plug-in to DOME for third party applications, interprets the metadata generated by a corresponding publisher in order to create a front-end service interface constructed [2]. The administrators take these models and make them available through the DOME server which then allows models to be accessed anywhere by users via the internet. Users can then run design simulations from anywhere. And they can connect models together and perform design trade-off analysis and optimization of these models [1].

3.2 Mechanical Engineering Tools Class (2.670)

2.670 is a required mechanical engineering class taught during the Independent Activities Period (IAP) at MIT. The objective of this class is to introduce fundamental machine tool and computer tool use. This is done through the construction of a Stirling Engine (see **Fig 1**) by each student.

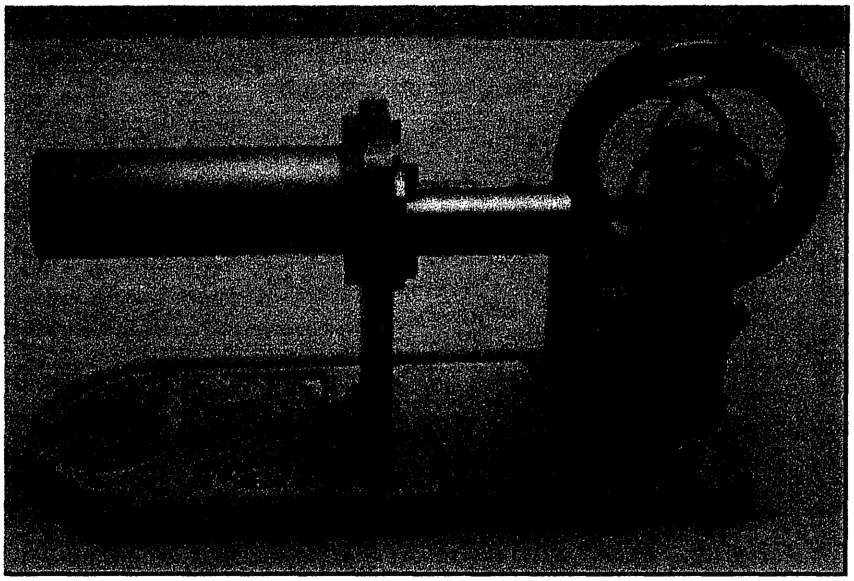

Figure 1: 2.670 Stirling Engine

The Stirling Engine operates on a closed regenerative thermodynamic cycle also known as the Stirling cycle. The 2.670 engine, unlike an internal combustion engine, has no valves and does not intake or exhaust gas. Instead the air is trapped and heat is converted to mechanical work by pushing air from the cold to the hot side of the engine in alternation. Through the use of thermodynamic regenerative processes, the efficiency of the Stirling engine is improved over more conventional engine cycles. In theory, a Stirling engine can operate with an almost ideal efficiency (Carnot cycle). However, the 2.670 engine loses energy from friction, poor heat transfer between the flame and the engine, and from the heat conduction between the hot and cold side of the engine through the walls of the displacer piston and cylinder [6].

Students learn to use machine tools to build parts for the engine, a CAD program called Solid Works to design solid models of parts, and other tools such as MATLAB to study the thermodynamics of the engine. For the most part, all of the engine parts are uniform. However, if so desired, students are allowed to make design changes to certain parts in order to compete for the fastest engine at the end of the class.

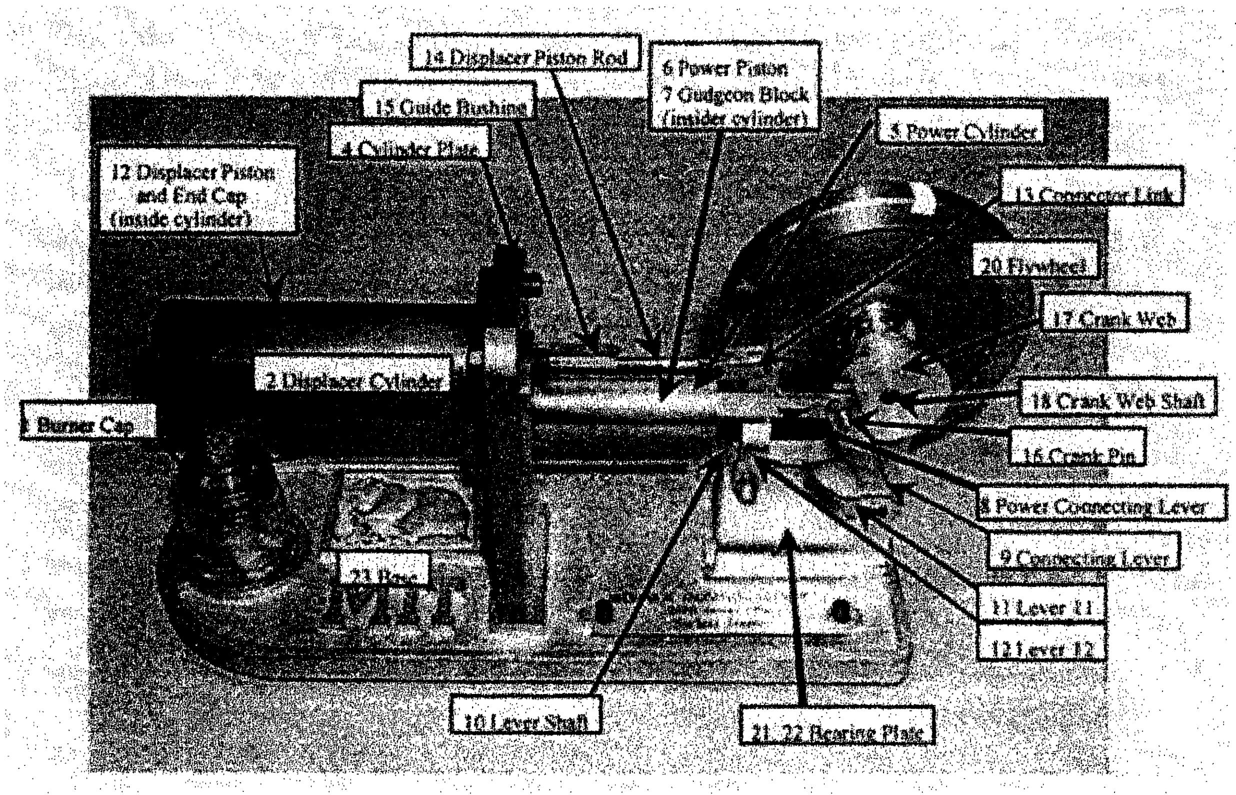

Figure 2: Stirling Engine with Labeled Parts [6]

4.0 Stirling Engine Model Parameters

There are many parameters of the 2.670 stirling engine that students in the past have considered modifying, but very few have resulted in a significant improvement in performance. The ones that have had an effect on performance have been changes made to dramatically lower friction. These changes increase speed by reducing the energy per cycle lost by friction, but they do not increase the power output of the engine. The analysis done here will study the effects on the work per cycle of the engine and thus the engine power. According to Professor Doug Hart, most modifications are too costly. However, because the displacer piston is one of the most expensive parts, any reasonable modification to this part will likely have little impact on cost [4]. Thus, the displacer piston will be the focus of the 2.670 stirling engine that will be examined for modification and optimization. In particular, two aspects of the displacer piston will be analyzed.

4.1 Material of the Displacer Piston

Currently, the displacer piston is made by hollowing out a solid piece of brass. Analysis previously done has demonstrated that the thermal resistance of the displacer piston is dominated by the convective heat transfer term and thus it matters little what the displacer material is. The best material would therefore be something inexpensive, light, and easily machined [4].

4.2 Dimension of the Displacer Piston

Modifications of the dimensions of the displacer piston are key because they affect the average volume of the engine hot and cold side, along with the change in hot and cold engine volume, all of which are important parameters that effect the work per cycle of the engine. The current outer diameter of the displacer piston is 1.063 inches and its length is 1.960 inches. Due to restrictions of the size of the base, there is a limit of 1.935 inches of increase in diameter of the current displacer piston, and due to restrictions of the heat transfer cylinder there is a limit of 1.127 inches in its current length.

5.0 Design and Construction

5.1 Thermodynamics Model of Stirling Engine [5]

The mechanical work output by the stirling engine can be calculated by analyzing its thermodynamic cycle. Starting with the basic definition of work:

$$Work = \int F dx = \int PA dx = \int P dV \tag{1}$$

where P is the air pressure inside the engine and V is the volume of air in the engine, which is equal to the sum of the of the volume of air on the cold and hot side. The volume of the cold side is:

$$V_c = \overline{V_c} - v_d \sin(\theta + \phi) + v_p \sin\theta \tag{2}$$

and the volume of the hot side is:

$$V_H = \overline{V_H} + v_d \sin(\theta + \phi) \tag{3}$$

where v_d and v_p are the volumes of gas displaced by the displacer piston and power piston respectively. φ is the phase angle of the displacer piston relative to the crank angle θ and

$V_{c\ bar}$ and $V_{H\ bar}$ are the average volumes of the cold and hot side of the engine respectively.

Therefore, the total engine volume is defined as:

$$V = [\overline{V_c} - v_d \sin(\theta + \phi) + v_p \sin\theta] + [\overline{V_H} + v_d \sin(\theta + \phi)] \quad (4)$$

and $dV = v_p \cos\theta d\theta.$ (5)

The ideal gas law is used next to express the engine air pressure, P, as a function of the air temperature.

$$p = \frac{MRT}{V} \quad (6)$$

Therefore, the engine mechanical work per cycle is:

$$\frac{Work}{cycle} = \int_0^{2\pi} \frac{MRT}{V} v_p \cos\theta \cdot d\theta . \quad (7)$$

This expression is non-dimensionalized as follows:

$$'W = \frac{\frac{Work}{Cycle}}{P_{engine} \cdot v_p} = \frac{\int_0^{2\pi} \frac{MRT}{V} v_p \cos\theta \cdot d\theta}{\frac{v_p}{2\pi} \int_0^{2\pi} P d\theta} . \quad (8)$$

Using the ideal gas law, the average temperature of the air inside the engine can be approximated to be:

$$T \cong \frac{V_c T_c + V_c V_H}{V} , \quad (9)$$

where T_c and T_H are the temperatures of the air on the hot and cold side respectively. Because the 2.670 engines do not have perfect seals around the piston or guide bushing, as the average temperature increases from the heat, the air slowly leaks out of the engine. Thus in this case, the average pressure inside the engine can be set equal to the atmospheric pressure. Using this fact and the substitution for T from above, the non-dimensional engine work per cycle becomes:

$$'W = \frac{\frac{Work}{Cycle}}{P_{atm} \cdot v_p} = 2\pi \frac{\int_0^{2\pi} \left[\frac{V_c\left(\frac{T_c}{T_H}\right) + V_H}{V^2} \right] \cos\theta \cdot d\theta}{\int_0^{2\pi} \left[\frac{V_c\left(\frac{T_c}{T_H}\right) + V_H}{V^2} \right] d\theta} \qquad (10)$$

Substituting the previous equation for V_c, V_H, and V into the above equation, gives the final equation.

5.1.1 Matlab Analysis of Thermodynamics Model

Matlab was used to analyze the thermodynamic model previously discussed, specifically for the engine work per cycle as a function of the cold side to hot side temperature ratio, T_c/T_H, also known as T_i. Two m-files, enginenum2.m and engineden2.m, were written using the quad function to calculate the integrals of the numerator and denominator of equation 10 separately. The values of average V_c, V_H, v_p, and v_d were calculated using values of the diameters and lengths of both the displacer piston and power piston. The phase angle was a constant value of $\pi/2$. An m-file, enginefinal2.m was used to calculate the work per cycle and plot the function as a function of T_i, where T_i was declared a global variable that varied from 0 to 0.99 in increments of 0.01. A final m-file, main.m, was created to call the enginefinal2 file and declare the diameter and length of the displacer piston, which would be passed in from a user input, as global variables. (See **Appendix A** for MATlab files).

5.2 Solid Model of Stirling Engine

Below is the 2.670 Stirling engine that was created as a parametric model in SolidWorks (**Fig 3**). The diameter and length of the displacer piston are the two dimensions that are changeable. Because it is a parametric model, the dimensions of the displacer cylinder, cylinder plate, and base are driven by the dimensions of the displacer piston, and increase or decrease parametrically depending on the changes made to the displace piston.

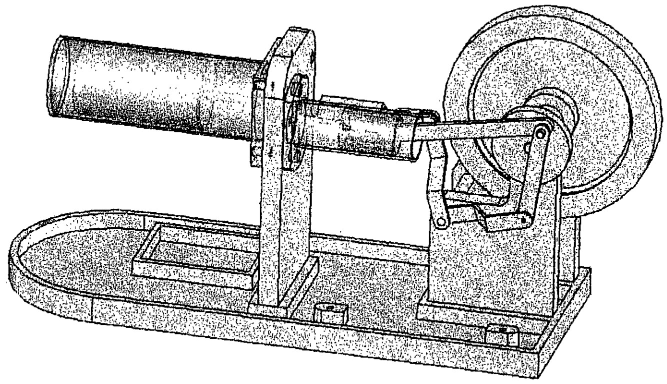

Figure 3: Solid Model of 2.670 Stirling Engine

5.3 DOME Wrapped Models

The two models discussed in section 5.2 were constructed as DOME wrapped models.

5.3.1 Matlab Wrapper Model

The Matlab code created for the analysis of the thermodynamic model takes the diameter and length of the displacer piston to calculate values of average volume of the engine hot side, V_H, and change in hot and cold engine volume from the displacer piston, v_d, in order to solve for the work per cycle, W. Therefore in creating the Matlab wrapper model, the diameter and length of the piston were defined as the independent parameters and the work per cycle was defined as the derived parameter as shown (**Fig 4**).

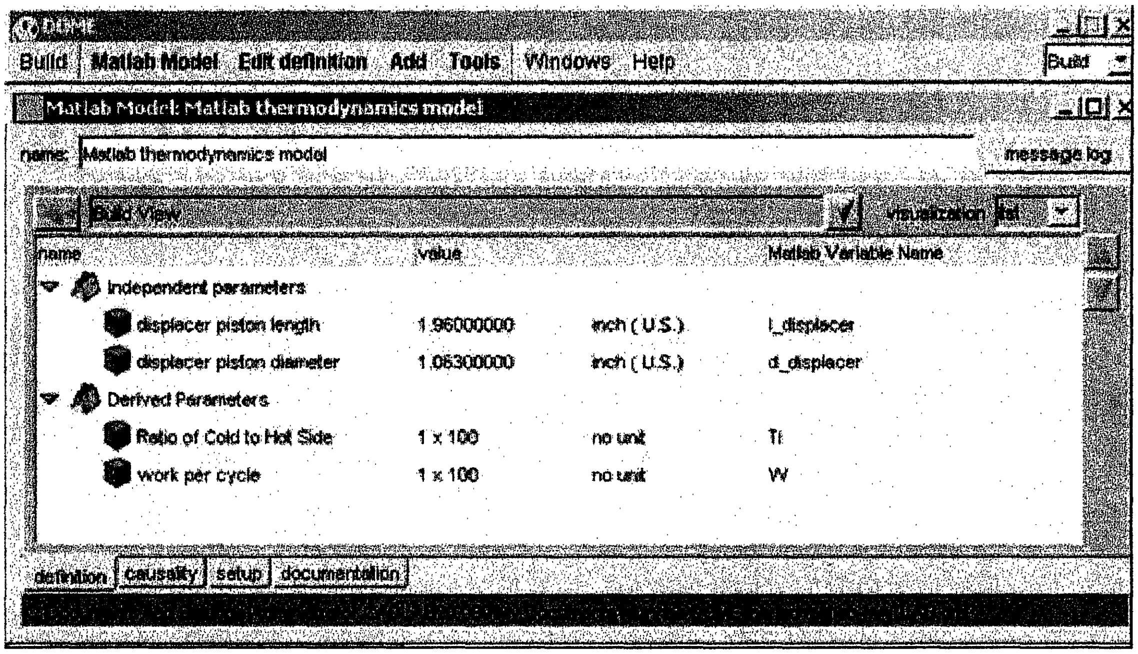

Figure 4: Parameters of Matlab Thermodynamics Model

Figure 4 shows the definition of the wrapper model. The current values shown are the default values set by the expert. Both the diameter and length of the displacer piston are real values whereas the work per cycle has been defined as a 1 x 100 matrix. The far right column shows the name of the Matlab variable to which each of these parameters are linked or mapped to. One may notice that the *ratio of cold to hot side*, Ti, has been included as a derived parameter although in reality it can stand alone. This was created as a "dummy" variable so that it could later be used as an output when creating the "*Work per Cycle* vs. *Ratio of Cold to Hot Side*" curve.

After the variables have been determined, the relationships are determined by the causality tab (**Fig 5**).

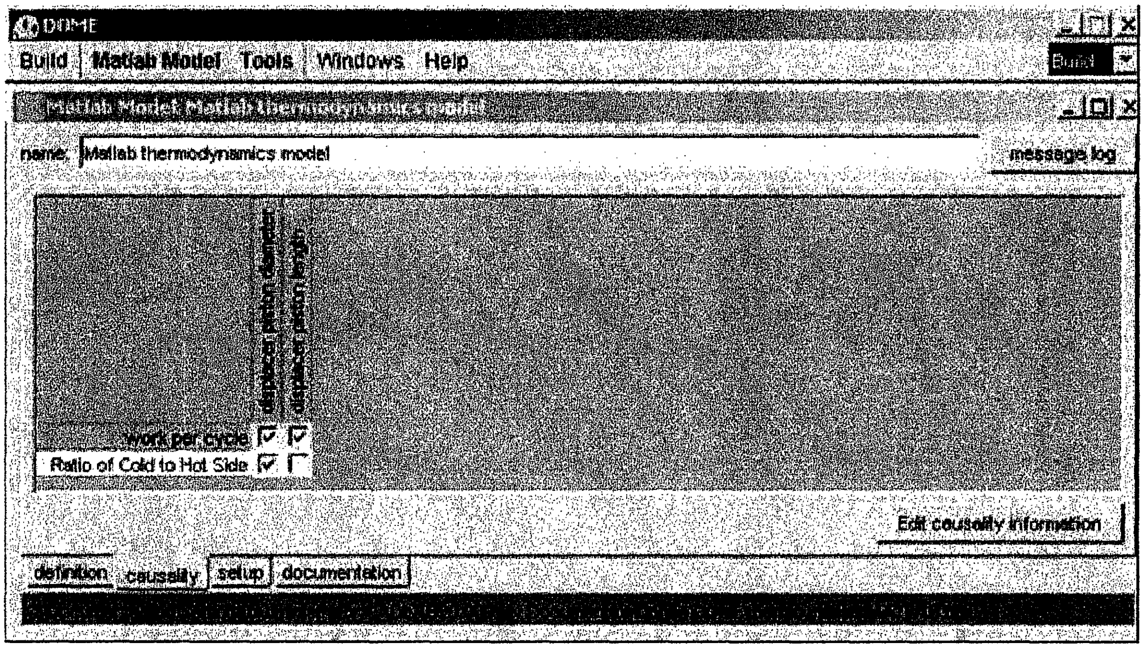

Figure 5: Causality of Matlab Thermodynamics Model

The checked boxes indicate which objects in the row depend on the objects in the column. Once again, the *ratio of cold to hot side* was used as a "dummy variable."

The setup is used to link the matlab files to the model wrapper, indicate which file is the main one, and determine the software version of the program, in this case matlab (**Fig 6**).

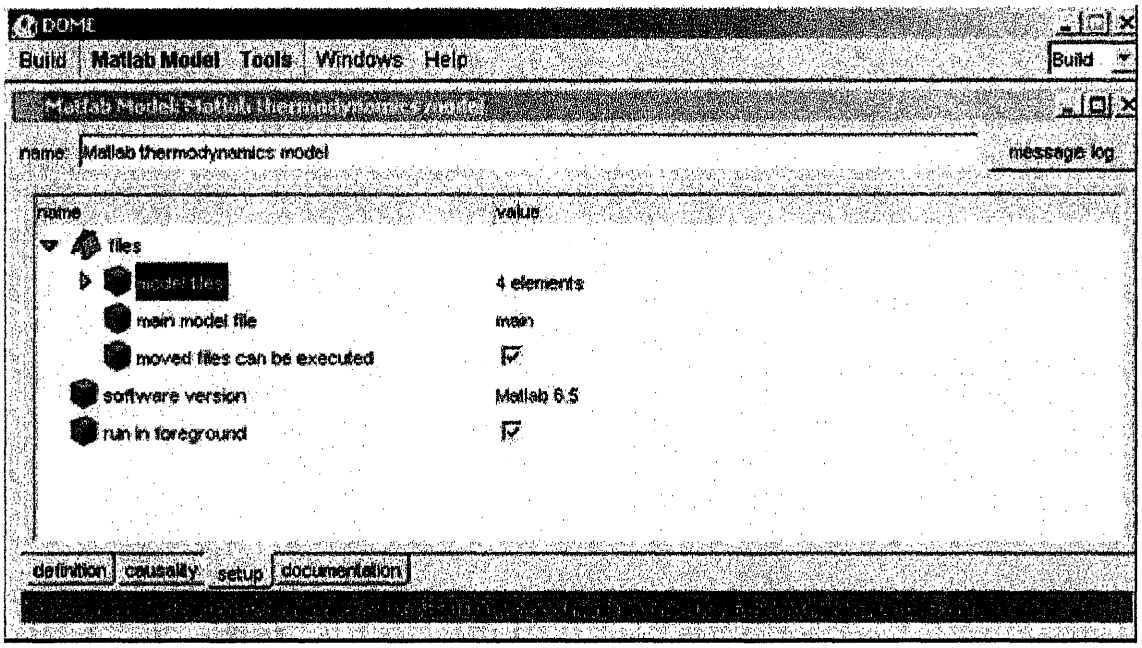

Figure 6: Setup of Matlab Thermodynamics Model

The model files icon can be opened to add the files (**Fig 7**). As seen below, the four Matlab files previously discussed in 5.2.1 are associated with this model.

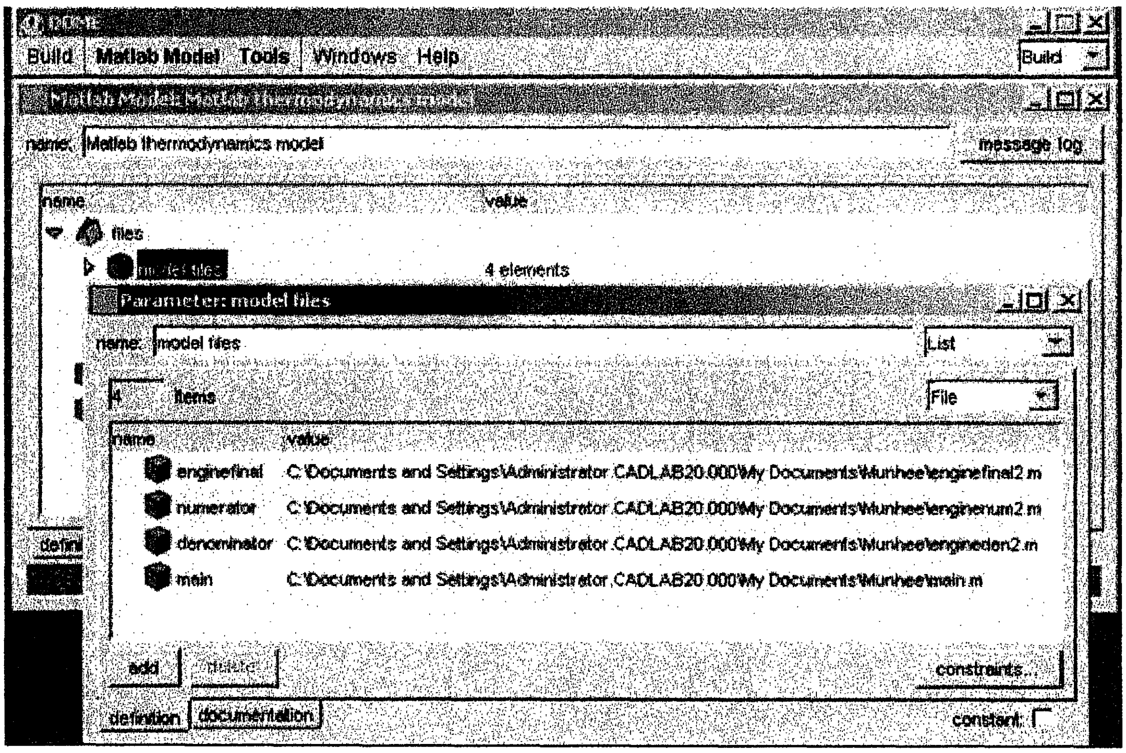

Figure 7: Files of Matlab Thermodynamics Model

Next, an interface must be created. The interface is defined by the expert to define the user accessibility in a model. In this case, the independent parameters have been defined as the user inputs and the outputs are the *work per cycle* matrix and the *ratio of cold to hot side* matrix (**Fig 8**).

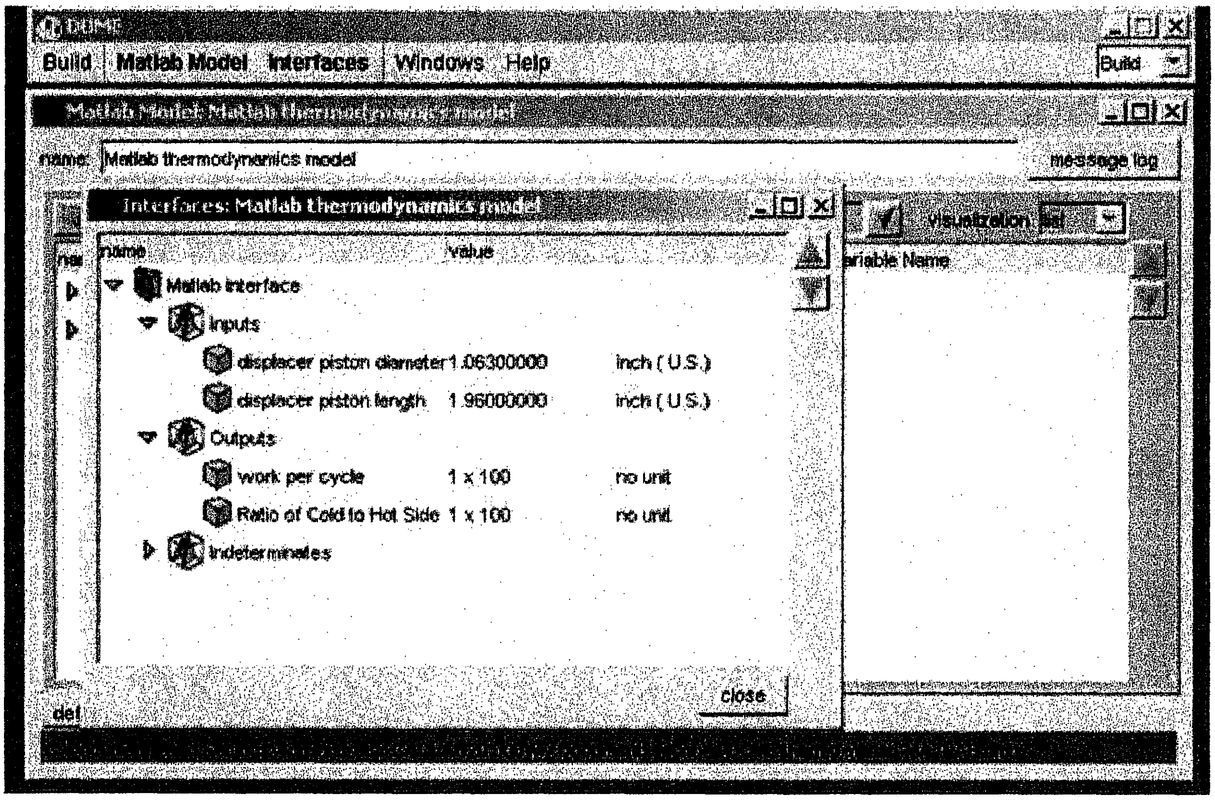

Figure 8: Interface of Matlab Thermodynamics Model

Finally, the model was deployed to the server.

5.3.2 SolidWorks Wrapper model

The 2.670 Stirling Engine was created as a parametric Solid works model in which the dimensions of the diameter and length of the displacer piston are changeable. Therefore, the Solid works Wrapper model was created in which the dimensions of the displacer piston could be modified by an increase or decrease of 0.5 inches in the diameter and a decrease of 1.5 inch in the length. Similarly to the Matlab wrapper model, the independent parameters are the displacer piston diameter and length (**Fig 9**). The current values shown are the default values set by the expert. Both the diameter and length of the displacer piston are real values and the far right column shows the name of the dimension or property of the part in SolidWorks to which each of these parameters are mapped to.

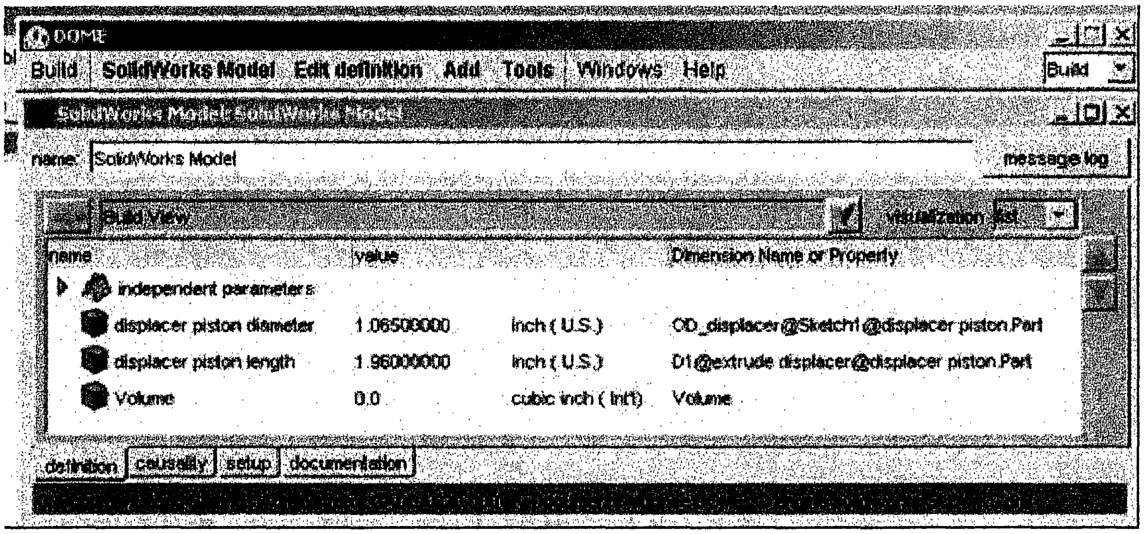

Figure 9: Parameters of SolidWorks 3D Model

Another dummy variable, "Volume," was introduced this time as well. Although this time it was in order to be able to set up causality (**Fig 10**).

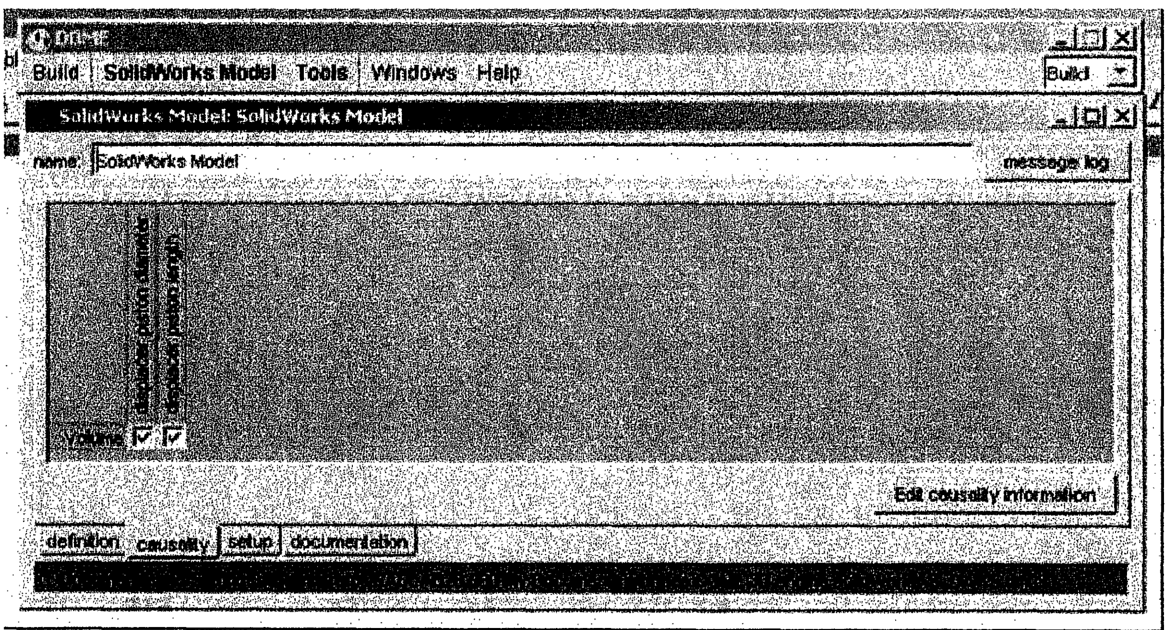

Figure10: Causality of SolidWorks 3D Model

In this case, the part file of the displacer piston and assembly file of the Stirling engine have been added to the model (**Fig 11**). The assembly file has been chosen as the main file and SolidWorks has been chosen to run in the foreground so that the changes propagated in the engine assembly could be seen.

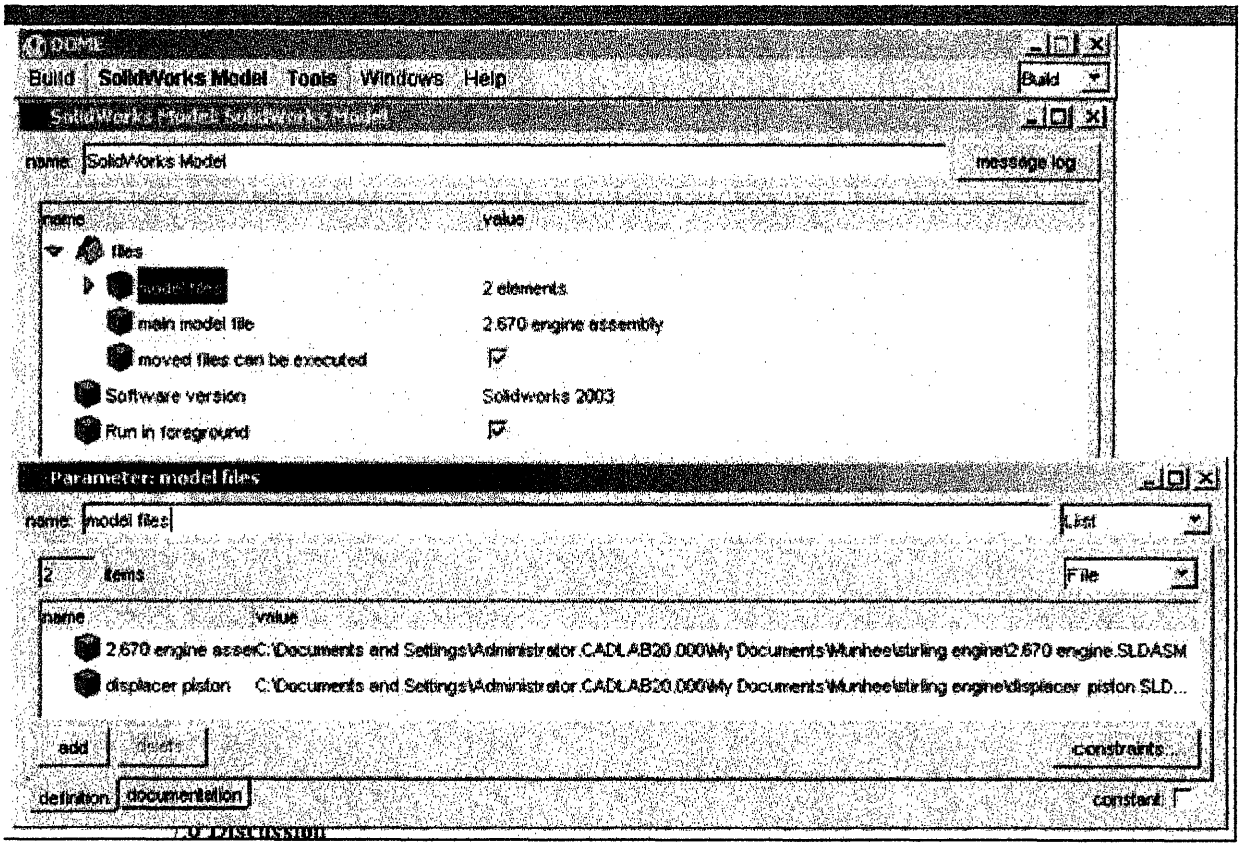

Figure11: Setup and files of SolidWorks 3D Model

Finally, an interface was created and the model was deployed to the server. In this case, the independent parameters have been defined as the user inputs and the output is the dummy variable "Volume" (**Fig 12**).

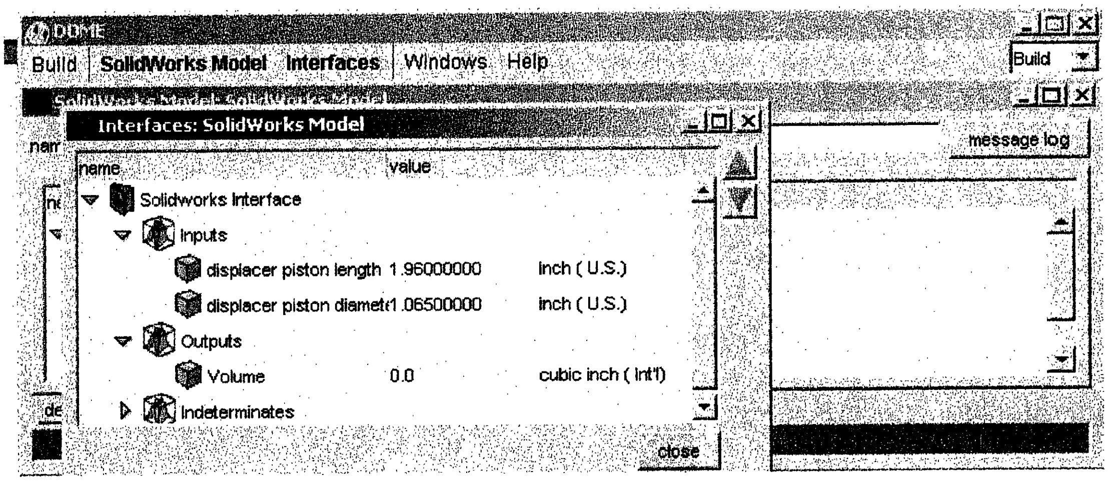

Figure12: Interface of SolidWorks 3D Model

5.3.3 Integrated Model

After the construction of the Matlab wrapper model and the SolidWorks wrapper model, an integrated model was formed in an integrated project. The integrated model is called iModel. In order to do this, an integrated project called Engine project was created in which both the Matlab and SolidWorks wrapper model were added as resources to the project (**Fig 13**).

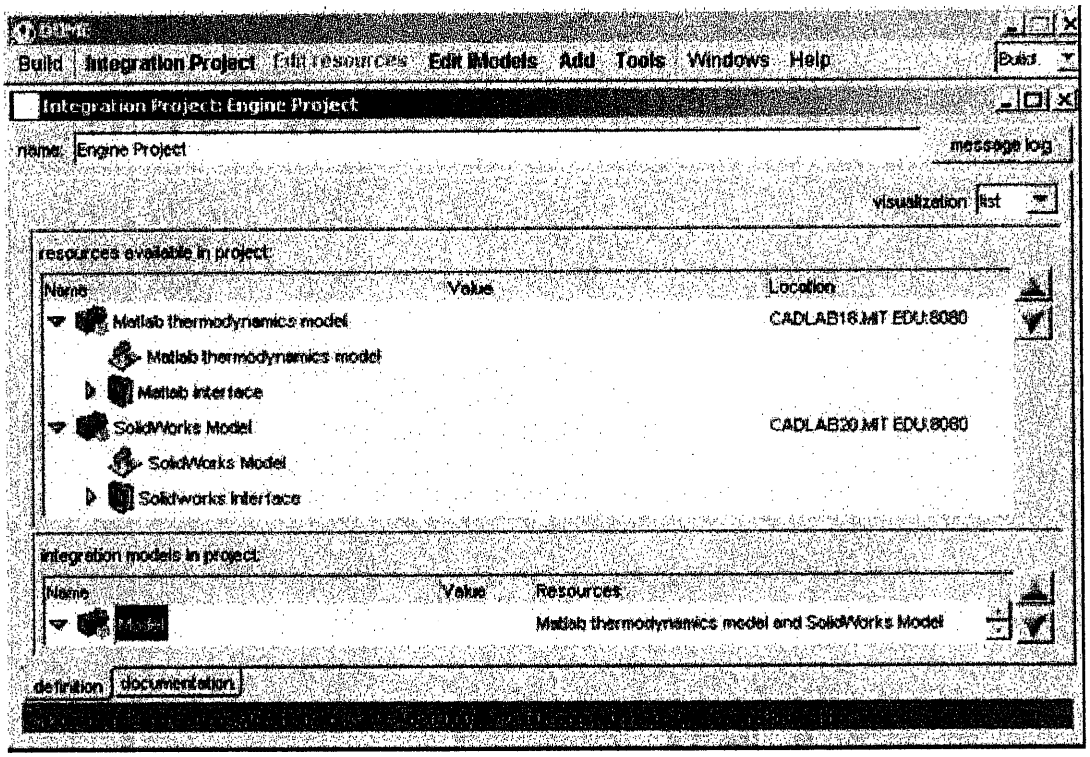

Figure 13: Integration Project and Resources

In the iModel, three subscriptions were initially added: the Matlab interface, the SolidWorks interface, and visualization, which will be used to output the work per cycle vs. Tc/Th curve derived from the Matlab thermodynamics model (**Fig 14**).

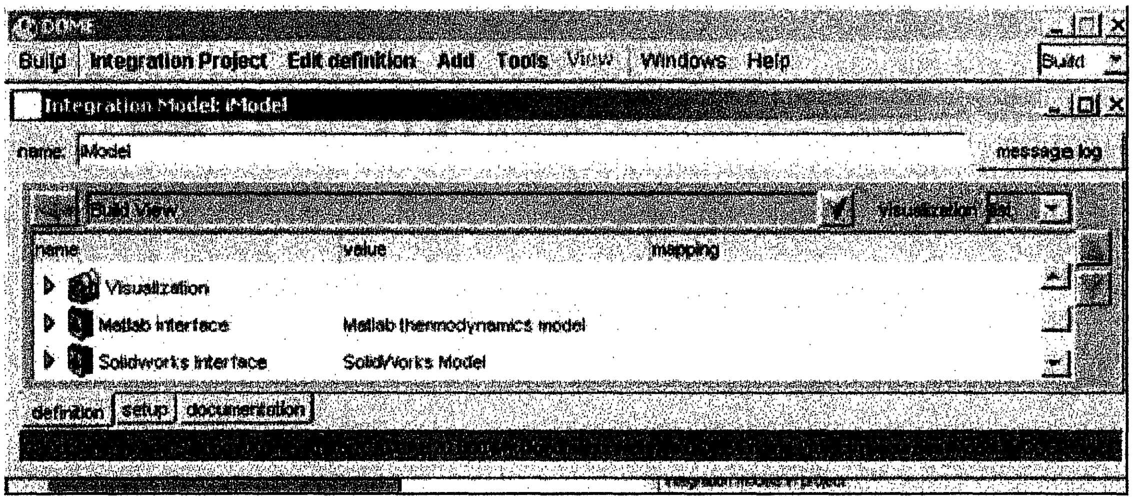

Figure 14: iModel and Subscriptions

First, the Matlab model and visualization were linked. Since the visualization would be produced by the *work per cycle* and *Ratio of cold to hot side*, a copy of these two matrices were added into the iModel and these corresponding variables from the Matlab interface and visualization were mapped to the copies (**Fig 15**).

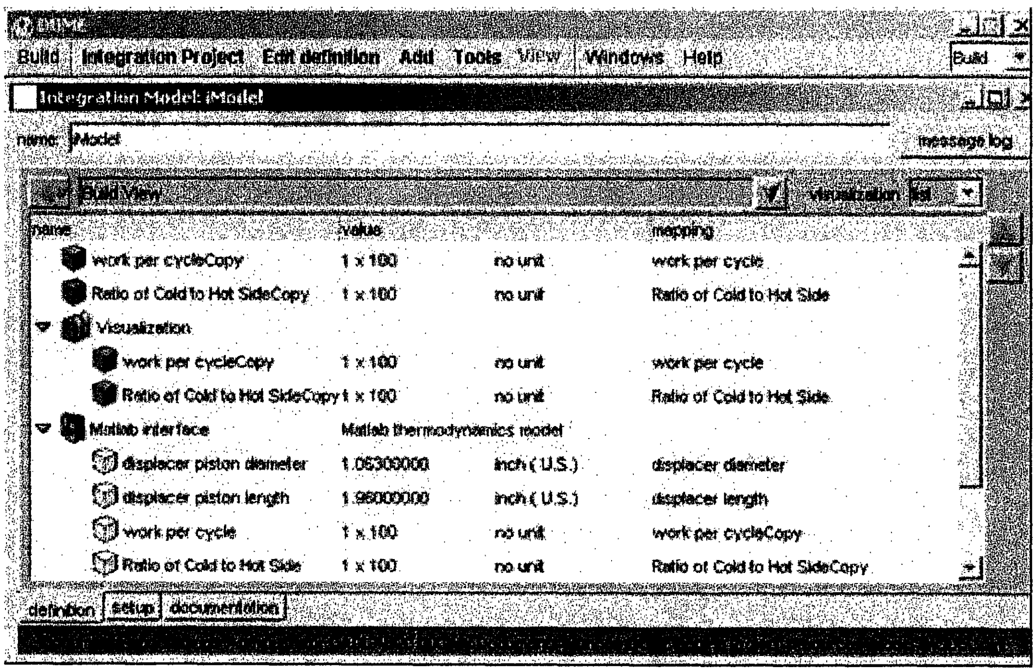

Figure 15: Matlab and Visualization Mapped

Similarly, the Matlab model and SolidWorks model were linked. A copy of the dimensions for the diameter and length of the displacer piston were created and mapped to the respective dimensions in the respective models (**Fig 16**).

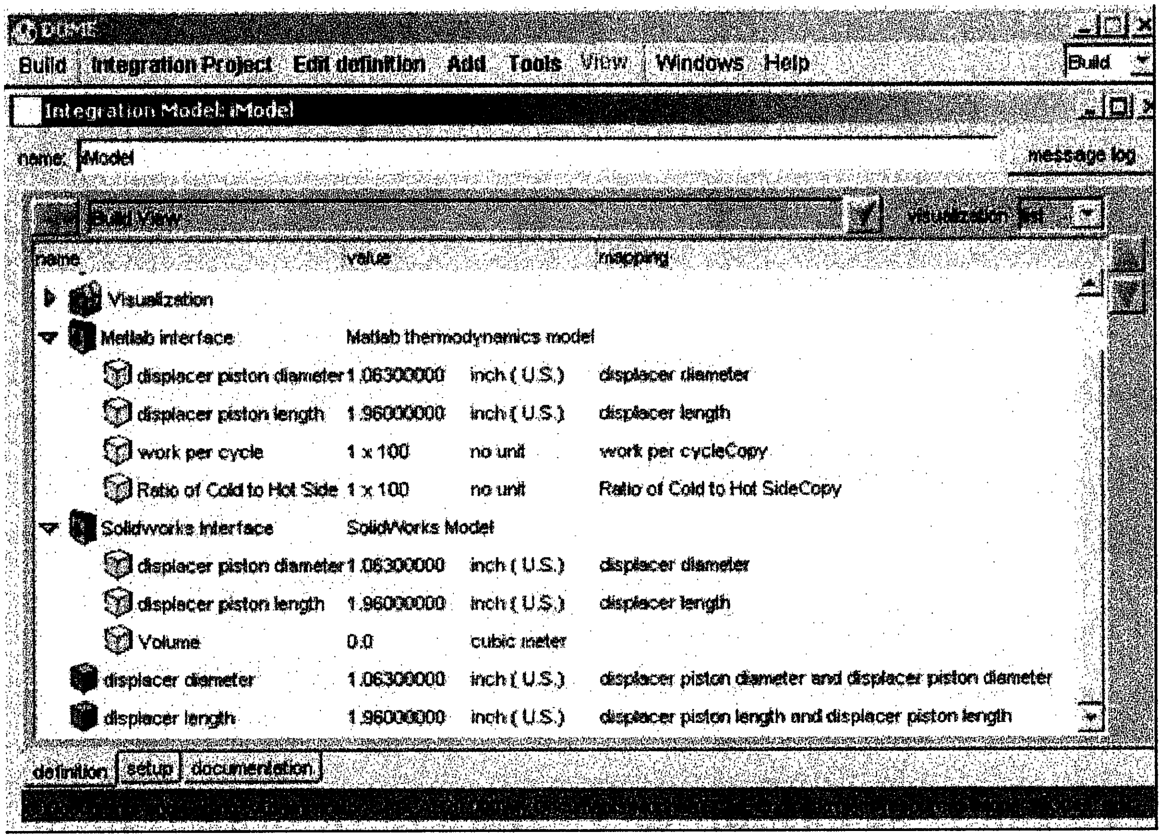

Figure 16: Matlab and SolidWorks Mapped

Finally, an iModel interface called "Dome Model Inteface" was created in which the input was the piston diameter and length and the output was the visualization. The following section will show a demonstration of this interface and the output of the iModel.

6.0 Demonstration

Once the project has been deployed, it is ready to run. Once the user has logged in and the project is opened, the iModel is opened to reveal the interface (**Fig 17**).

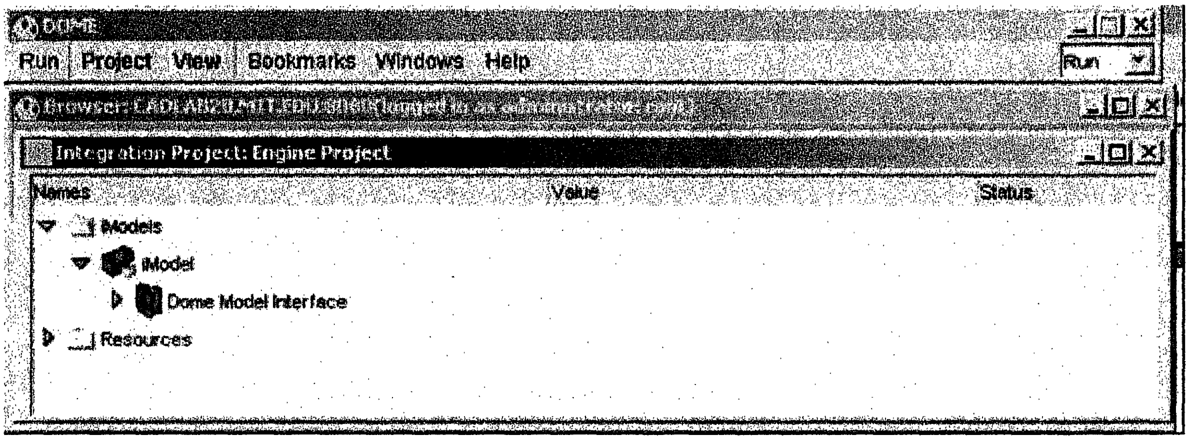

Figure 17: Opening iModel

Double-clicking on the Dome Model Interface icon will open up the user interface (**Fig 18**).

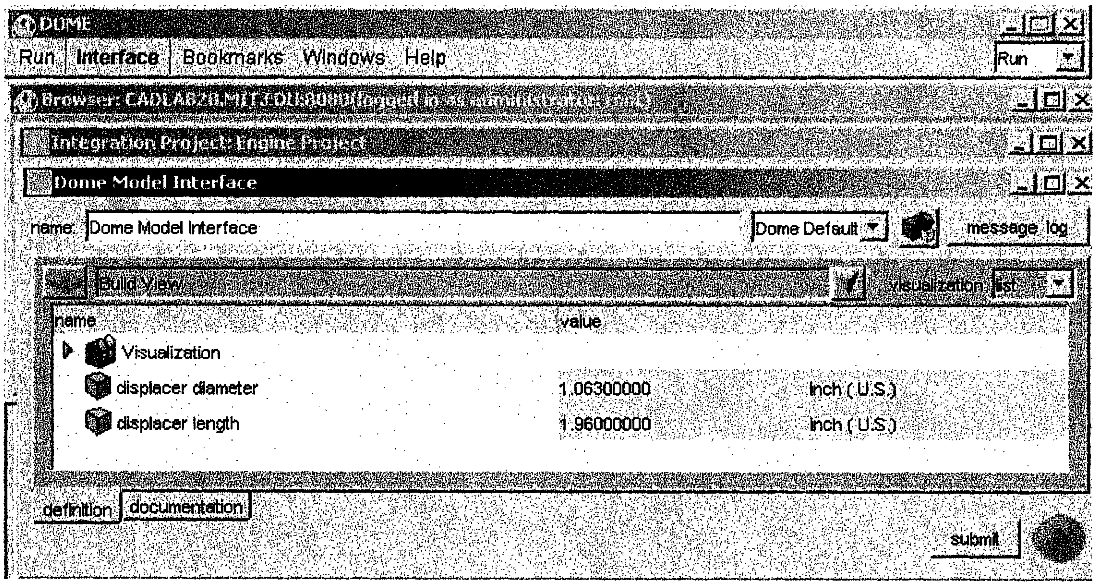

Figure 18: User Interface

The dimensions shown are the default values. The green indicates that they have not been changed. In this demonstration, the diameter of the displacer will be increased by 0.5 inches and the length of the displacer length will be decreased by 1.5 inches. The yellow indicates that the dimensions have been changed (**Fig 19**).

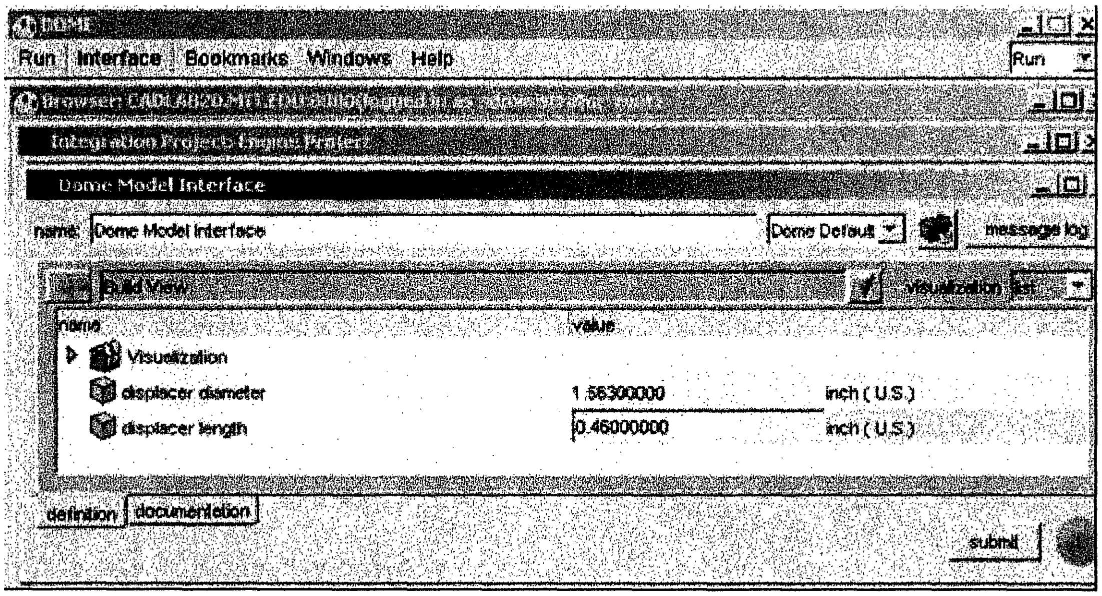

Figure 19: Changing Inputs

Next the submit button is pressed and SolidWorks will pop up with the dimension changes. A comparison of the original and new dimensioned engine is shown below (**Fig 20**).

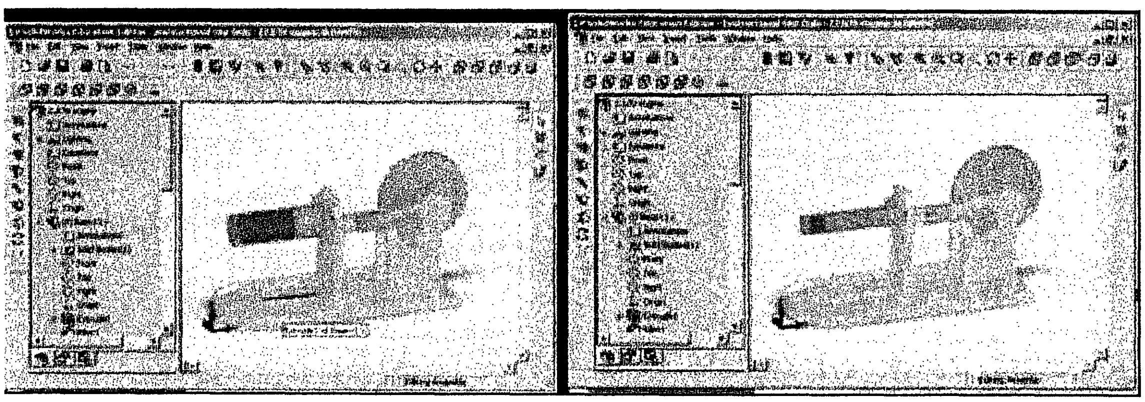

Figure 20: Original vs. New SolidWorks Model

Double clicking on the visualization icon will produce the following curves of Work per cycle vs. Ratio of Cold to Hot Side (Tc/Th). Figures 21 and 22 show the graphs for the original and new dimensions, respectively.

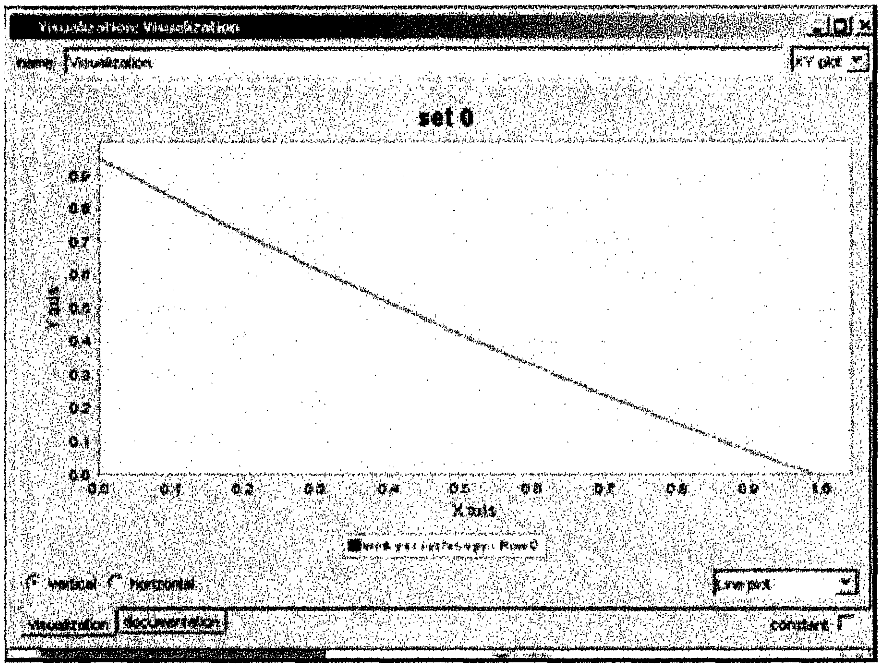

Figure 21: Work per cycle vs. Ratio of Cold to Hot Side (Tc/Th), Displacer diameter = 1.063in, length = 1.96in

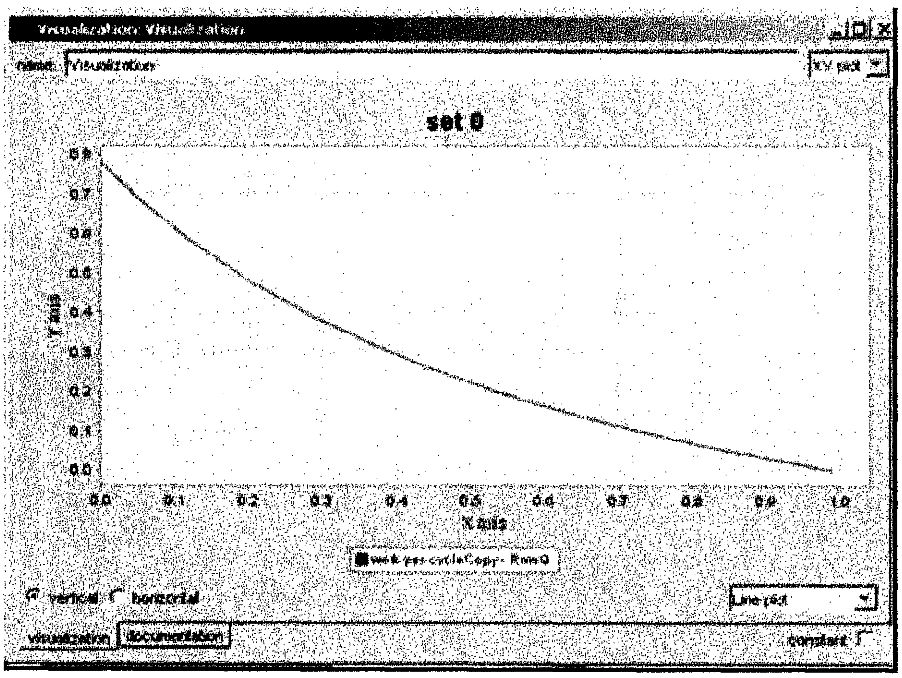

Figure 22: Work per cycle vs. Ratio of Cold to Hot Side (Tc/Th), Displacer diameter = 0.563in, length = 0.46in

7.0 Conclusion

Through this thesis project I learned many new things and got to apply the things that I learned. First of all, I was able to learn how to better use SolidWorks for 3-D parametric modeling. I was also able to apply my computer programming skills to write the code for the Matlab analysis. Most important of all, I learned what current integrated design looked like and how DOME played a role in the future improvement of integrated systems. With the help of many graduate students in the lab and through DOME tutorials, I was able to learn how to use DOME in order to carry out my thesis project.

In general, I am pleased with the results of this project. DOME was easy to learn how to use and I was impressed at the capabilities of web accessibility and the speed of design analysis and optimization. However, if I could go back, there are several things I would change about this project and the methods in which I pursued it. First of all, I created the SolidWorks model without really understanding how parametric modeling worked. This caused problems because there were a few cases in which parametric modeling could have been much more simplified had I created my parts in a particular way that is friendly to parametric modeling, for example, thinking about the location of the reference point of the part. Secondly, I started off with creating the solid model of the engine in SolidWorks and writing Matlab code for the thermodynamic analysis before I learned how to use DOME. However, this turned out to be a problem because there were times when I had to go back to redo or rename SolidWorks dimensions or Matlab file names in order for it to be better compatible to DOME code. Lastly, because DOME is still in its prototype stage, tutorials and help sections were not entirely complete and thus whenever I ran into a problem or question, at times, it took a long time to find someone who knew how to answer my question and had the time to answer it.

Future modifications to this project could include a 2-D animation wrapper that links to my thermodynamics and solid model wrappers. It would also be nice to include a custom GUI so that it would be more user-friendly to future 2.670 students. Once, DOME is officially released, and if it will be available at MIT, the future hope is that 2.670 students could use my project to better analyze the 2.670 Stirling Engine.

Appendix A: MATlab code for Thermodynamics Model

main.m

```
global d_displacer;
global l_displacer;

Ti = 0:.01:.99;

W = enginefinal2(d_displacer, l_displacer, Ti);
```

enginefinal2.m

```
function W = enginefinal2(d_displacer, l_displacer, Ti)

global T;
global d_power;
global l_power;

d_power = 0.625;
l_power = 0.875;


for m= 1: 1: 100

   T = m/100;
   Q1(m) = quad(@enginenum2,0,2*pi);
   Q2(m) = quad(@engineden2,0,2*pi);
   W(m)=2*pi*Q1(m)./Q2(m);

end

plot (Ti, W);
title('Normalized Engine Work per Cycle vs. Ratio of Cold to Hot side Temperature');
xlabel('Tc/Th');
ylabel('W');
```

enginenum2.m

```
function num = enginenum2(x)

global T
global d_displacer
global l_displacer
global d_power
global l_power

d_heatcyl = d_displacer + 0.125;
d_powercyl = d_power + 0.035;

Vh_max = pi*d_heatcyl^2/4*(3.510-1.960);
Vh_min = pi*d_heatcyl^2/4*(3.510-1.960-.5);

Vc_max = pi*d_powercyl^2/4*(.25-(l_power-.875)) + pi*d_heatcyl^2/4*(.5+(1.960-
l_displacer));
Vc_min = pi*d_powercyl^2/4*(.25-(l_power-.875)) + pi*d_heatcyl^2/4*(1.960-
l_displacer);

Vc = (Vc_max+Vc_min)/2; %Average volume of engine cold side
Vh = (Vh_max+Vh_min)/2; %Average volueme of engine hot side
vp = pi*d_power^2/4*.5; %Change in engine volume from the piston
vd = pi*d_displacer^2/4*.5; %Change in hot and cold engine volume from the displacer
piston
p = pi/2; %phi= phase difference between the displacer and power piston

num = (cos(x).*(Vh + vd*sin(x+p) + (Vc + vp*sin(x) - vd*sin(x+p))*T))./((Vc + Vh +
vp*sin(x)).^2);
```

engineden2.m

```
function den = engineden2(x)

global T
global d_displacer
global l_displacer
global d_power
global l_power

d_heatcyl = d_displacer + 0.125;
d_powercyl = d_power + 0.035;

Vh_max = pi*d_heatcyl^2/4*(3.510-1.960);
Vh_min = pi*d_heatcyl^2/4*(3.510-1.960-.5);

Vc_max = pi*d_powercyl^2/4*(.25-(l_power-.875)) + pi*d_heatcyl^2/4*(.5+(1.960-
l_displacer));
Vc_min = pi*d_powercyl^2/4*(.25-(l_power-.875)) + pi*d_heatcyl^2/4*(1.960-
l_displacer);

Vc = (Vc_max+Vc_min)/2; %Average volume of engine cold side
Vh = (Vh_max+Vh_min)/2; %Average volueme of engine hot side
vp = pi*d_power^2/4*.5; %Change in engine volume from the piston
vd = pi*d_displacer^2/4*.5; %Change in hot and cold engine volume from the displacer
piston
p = pi/2; %phi= phase difference between the displacer and power piston

den = ((Vh + vd*sin(x+p) + (Vc + vp*sin(x) - vd*sin(x+p))*T))./((Vc + Vh +
vp*sin(x)).^2);
```

References

[1] *Dome Fact Sheet*
http://cadlab.mit.edu/research-dome/fact_sheet.shtml

[2] David R. Wallace, Shaun Abrahamson, Nicola Senin, Peter Sferro, Integrated Design in a Service Marketplace. *Computer-aided Design*, volume 32, number 2, pp. 97-107, 2000.

[3] Nicola Senin, David R. Wallace, Nicholas Borland, Distributed Object-Based Modeling in Design Simulation Marketplace, *ASME Journal of Mechanical Design*, volume 125, March, 2003.

[4] Hart, Douglas. *2.005 Stirling Redesign*, Fall 2002

[5] Hart, Douglas. *Stirling Engine Analysis*. March 2002.

[6] 2.670 MIT Mechanical Engineering Tools handbook