GABSys: Using Genetic Algorithms to Breed a Combustion Engine

B. Danielson J. Foster D. Frincke

Abstract— **Most engineering systems are designed on the basis of a steady, or static, state. However, dynamic effects must also be considered. This paper presents GABSys (Genetic Algorithm-Bond Graph System), a tool for approximating design optimality for multi-dimensional dynamic systems. Although GABSys does require the assistance of an engineer, it can be used to "automatically" generate solutions to design problems. We illustrate GABSys' utility by using it to design a 2-stroke Combustion Engine. GABSys begins with behavioral requirements expressed in a bond graph and uses genetic algorithm techniques to produce a design that meets the specified behavioral requirements.**

Keywords— **Genetic Algorithms, Bond Graphs.**

I. Introduction

PRODUCING a successful design for a multi-dimensional dynamic system is a difficult task. One technique for describing dynamic systems is the bond graph. Bond graphs can model dynamic/static systems across engineering specialties from acoustic pressure to heat flow to electrical circuits[1]. *Bond graphs* are an abstract description of system behavior and provide a way to step state-space equations through time. Parameters of a bond graph include environment and structure. The interactions of effort and flow constrained by these parameters are described through the development of equations from the bond graph. In this work, we discuss GABSys (Genetic Algorithm-Bond Graph System), a tool for approximating design optimality for multi-dimensional dynamic systems described using bond graphs. By stepping through the bond graph, we can determine the acceptability of each potential system according to a pre-defined criteria. This acceptability may be used as a "fitness function" for guiding the selection of individuals for "breeding" using genetic algorithm techniques. Therefore, the genetic algorithm learns to evolve a mechanical system from a developed behavior described by the simulation. We have chosen the internal combustion engine as our target system.

GABSys can be used in the following manner. The preliminary design conception is created by an engineer through the development of a bond graph. System requirements are encoded as fitness functions and used to assess the acceptability of each design. The genetic algorithm develops the preliminary design through an automatically generated random population of chromosomes representing competing systems with random geometries (individuals). Chromosomes include the encoded parameters defined by the bond graph. Individuals are simulated by decoding the chromosomes to extract the value for each parameter, and fitness is assessed using acceptability calculations based on the resulting simulation.

This paper describes how GABSys may be used to to optimize 15 different parameters of a two-stroke internal combustion engine concurrently to produce an engine design.

II. Bond Graphs

A. Bond Graphs: The Behavioral Description

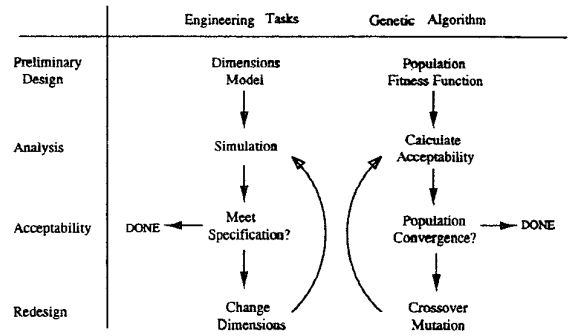


TABLE I
Tasks performed in designing dynamic systems

As shown in Table I, engineering design and genetic algorithms directly parallel one another. In engineering design, the preliminary design involves conception, developing the model, and an initial set of dimensions expected to meet a set of requirements. Simulation is used to determine if this model meets the requirements. If the model does not meet the specifications, adjustments are made and a return to simulation occurs. Once the requirements are met, the design is complete.

Bond graphs permit a mathematical simulation of components, machines, and systems. The equations used for the simulation are provided by systematically manipulating a bond graph to yield state-space differential equations for a final structural description. This manipulation is a straightforward process suitable for computer use [1]. Bond graphs represent systems more abstractly than physical prototypes. Therefore, computer-simulated experimentation can supplement or even replace hardware experimentation.

Bond graphs describe the interactions of effort and flow through a dynamic system using simple lines and symbols. They are a precise statement describing the use of inertia, friction, capacity, etc through various sections of the model and can potentially describe all possible effects of the initial conditions on a component or system. These effects are noted and the initial conditions are changed to create a

0-7803-4869-9/98 $10.00©1998 IEEE

new initial condition state. Through this method, bond graphs step state-space equations through time.

Each time step in a bond graph simulation is called an episode. During an episode, energy and power interactions are effectively linearized. Therefore, the smaller the episode, the more precise the solution. Dynamic systems inherently seek steady-state. The bond graph pushes current states toward a future steady-state one episode at a time. A balance is required between the length of an episode and the solution's required precision. Smaller episodes also require more calculations (and time) to simulate the dynamic system. The appropriate episode length varies from system to system. For example, determining acoustic pressure variations within a beam would require an episode length much smaller than determining the stresses in a beam due to atmospheric temperature changes. Hence, using the smallest required episode length to model the most complex component of a system would be satisfactory for the entire system.

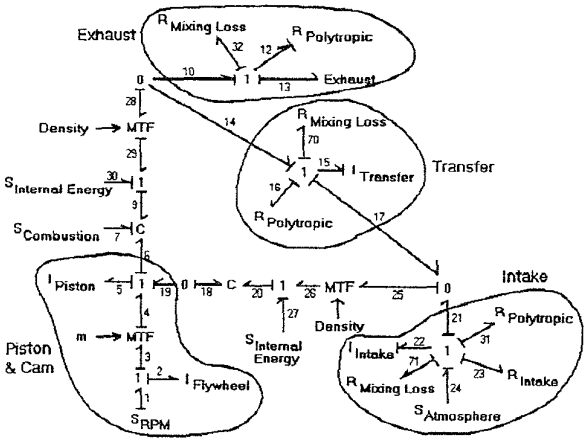


Fig. 1. Bond Graph of Two-Stroke Internal Combustion Engine taken from SAE Transactions 1975

A bond graph consists of junctions of constant effort (0), junctions of constant flow (1), causality (⊢), direction of flow (⟶), resistance (R), inertia (I), source (S), capacitance (C), and transfer functions (MTF) [1] [1]. The bond graph in Figure 1 describes the internal dynamic behavior of a two-stroke internal combustion engine. This example is essentially the model developed by Donald Margolis at the University of California at Davis [2]. Even for complex systems, this representation can be easily constructed by an engineer with prior training.

B. Parameters: The Structural Description

The structural description of the engine model is mathematically modelled through equations that constrain and direct the ability of the system. The bond graph is a particularly useful method for simulating the time history of important variables. Once defined by a bond graph, a dynamic system (or subsections of the system), can be reused in other systems with little effort, similar to "object-oriented" software development. For example, once a pipe has been modelled mathematically in a bond graph, the model can be considered sound for reuse in other systems using pipes. Other dynamic system control techniques, such as block diagrams[1], do not have this benefit, due to the feedback links complicating the procedure of connecting one system to another.

From the initial conditions of the system, a sequence of system states can be developed. Even a non-trivial task can require the representation and classification of a large amount of information. Consider a simple part with a limited potential: a pipe used to transfer fluids from Figure 3. The tracked variables could include stress, heat transfer, friction, mass flow (turbulent/laminar), and even phase change. What seemed to be a simple component has developed into a complex dynamic system with several variables and numerous states required. This can create a large number of degrees of freedom interacting in a non-linear manner.

Even with a more complex system, the number of parameters for optimization does not necessarily increase. The pipe has a radius, thickness, and material. The parameter concerning the fluid the pipe transfers is the media itself, since the fluid properties (viscosity, conductivity, etc.) contain all the information required to develop the structural behavior. Although the bond graph is very diverse and can model nearly any dynamic system, there are limits. Modeling a chemical reaction, for example, would be difficult with this procedure unless the model were changed from mass reaction to energy reaction.

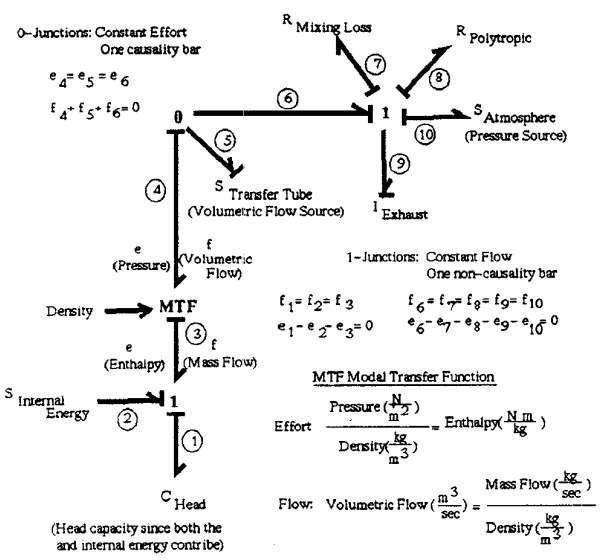
C. State-Space Equations: The Functional Description

State-space equations describe the function of the system by revealing the purpose of the system's behavior. As shown in Figure 2, reading a bond graph yields a set of coupled, first-order equations. These equations show how the structural behavior contribute to overall system function.

When accurate mathematical models are constructed, systems can be designed for optimum performance with a greatly reduced need for physical experimentation since the time history of important variables (e.g. temperatures, pressures, voltages, forces, and displacements) are calculated by the analytical solution of its state-space equations. Figure 2 illustrates how bond graphs develop the equations that model the two-stroke, internal combustion engine system with simple, yet powerful symbols.

In Figure 2, the state variables effort and flow are effectively coupled through the equations developed from the bond graph. Effort and flow describe the state of the system while the connections describe interactions of the states. The connections show both the direction of flow through a one-sided arrow and the direction of effort (causality) through a perpendicular bar. Causality explains how one component affects another or in other

[1] Not included are gyrators (MGY). They are similar to transfer functions, but work inversely. Gyrators are not required in this model.



$$C_{Head}\frac{d}{dt}e_1 = f_1 \qquad f_1 = f_2 = f_3$$

$$f_1 = f_3 = Density \cdot f_4$$

$$f_4 = -f_5 - f_6$$

$$f_5 = S_{Transfer\_Tube}$$

$$f_6 = f_7 = f_8 = f_9 = f_{10}$$

$$C_{Head}\frac{d}{dt}e_1 = Density \cdot (-S_{Transfer\_Tube} - f_9)$$

$$I_{Exhaust}\frac{d}{dt}f_9 = e_9$$

$$e_9 = e_6 - e_7 - e_8 - e_{10}$$

$$e_4 = e_5 = e_6$$

$$e_4 = e_3 \cdot Density$$

$$e_3 = e_2 - e_1$$

$$e_2 = S_{Internal\_Energy}$$

$$e_7 = f_7 \cdot R_{Mixing\_Loss}$$

$$e_8 = f_8 \cdot R_{Polytropic}$$

$$e_{10} = S_{Atmosphere}$$

$$I_{Exhaust}\frac{d}{dt}f_9 = Density \cdot (S_{Internal\_Energy} - e_1) \cdots + -f_9 \cdot (R_{Mixing\_Loss} + R_{Polytropic}) \cdots + -S_{Atmosphere}$$

Fig. 2. Coupled state-space equations. Following the causal links from a state variable (inertia, capacitance) to either a state variable or an input variable (effort or flow) results in two coupled equations with two unknowns.

words, describes the direction of effort and is the relationship/constraint that allows future prediction. There are precise rules for selecting causality within a bond graph; therefore, the causality can be automatically selected by the computer for any dynamic system that can be modelled with the bond graph technique.

III. The Two-Stroke Internal Engine

We successfully used GABSys to design a two-stroke internal combustion engine to a specified horsepower with minimal fuel-use. This engine is a dynamic system with a complex non-linear multi-dimensional domain; therefore, the techniques used in defining this system should apply to the design of other mechanical systems.

The two-stroke internal combustion engine (Figure 3) is a complex dynamic system with four separate operational states: compression, combustion, exhaust, and intake. Starting the piston at bottom dead center (as low as

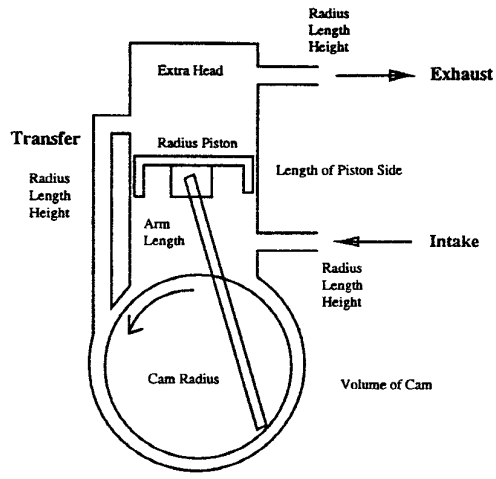


Fig. 3. Two-Stroke Internal Combustion Engine

it can go), the piston rises, building pressure in the head and reducing pressure in the cam area. The piston also closes the exhaust and transfer ports, and opens the intake port to allow fuel intake due to the decreasing pressure in the crank area. The power cycle begins with combustion. It increases the temperature and pressure in the head which pushes the piston downward. As the piston falls, the exhaust port opens, releasing the burnt fuel while increasing the pressure in the crank area. Next, the transfer port opens and the intake port closes. As the piston continues to fall, fuel is pushed from the crank area through the transfer port then into the head. With each turn of the crank, various pressures, flows, temperatures, densities, etc. move almost independently in various sections of the system. This is a difficult system to simulate, much less optimize.

We begin by developing a bond graph to provide a behavioral description of the engine, using Dr. Margolis' work as our starting point. We then describe individual combustion engines by encoding bond graph parameters in a chromosome. 15 parameters (Table II) must be optimized concurrently for the two-stroke internal combustion engine. For each of these parameters, the range and precision represented within the chromosome must be specified.

The chromosome defines the encoded parameters of the individual. To ease chromosome development, the range in this example was selected as a power of two, so each parameter can assume one of 256 values, using eight binary bits. The arm offset thus has a range from .001 m to .256 m. We chose to represent these values at even intervals throughout the range. Using a precision for the arm offset of 1 mm should include the optimal arm offset position. While the precision of each parameter can also be selected by the engineer perhaps by using a range of actual parts being produced, all the parameters in the list of this example have 256 potential values.

For the combustion engine, combining 15 parameters yields a chromosome with a length of 120 bits (15 * 8 = 120). The search space is thus 2^{120} ($1.329 * 10^{36}$) en-

No.	Parameters	Range (m)
1	Arm Offset	.001 – .256 m
2	Extra Height	.01 – 1.285 m
3	Radius of the Piston	.001 – .256 m
4	Piston Thickness	.001 – .256 m
5	Intake Height	.01 – .256 m
6	Intake Radius	.001 – .052 m
7	Intake Length	.01 – 1.285 m
8	Transfer Height	.01 – 1.285 m
9	Transfer Radius	.001 – .052 m
10	Transfer Length	.01 – 1.285 m
11	Exhaust Height	.01 – 1.285 m
12	Exhaust Radius	.001 – .052 m
13	Exhaust Length	.01 – 1.285 m
14	Volume of the Crank	.01 – 2.56 m^3
15	Radius of the Crank	.001 – .511 m

TABLE II

DEFINING THE CHROMOSOME

Arm Offset	Extra Height	Piston Radius	Piston Thickness	Intake Height	•••
01001011	01101001	01001101	10010011	01101110	

••	Intake Radius	Intake Length	Transfer Height	Transfer Radius	Transfer Length	••
	10101110	10111001	10010000	00110110	11011011	

•••	Exhaust Height	Exhaust Radius	Exhaust Length	Volume Crank	Radius Crank
	10101100	11010010	01001011	10011111	00100100

TABLE III

A CHROMOSOME - BINARY REPRESENTATION OF AN INTERNAL COMBUSTION ENGINE

gines, only some being viable.[2] Similar individuals may behave very differently. There are up to 15 separate dimensions described through the chromosome each having a non-linear effect on both the power and fuel consumption. The parameter range selected more than exceeds existing internal combustion engines and therefore, guarantees an engine supplying the required horsepower exists within the domain. This domain is so large that it may contain engines creating over 50 horsepower although the search is for engines creating only 8 horsepower.

IV. THE OPTIMIZED ENGINE

Figures 5 and 6 show the evolved 8 horsepower and 20 horsepower engines. They were developed using the same bond graph and same genetic algorithm method, merely changing the fitness functions to target different horsepowers. Both evolutions used the same fitness functions, one for fuel-use and one for horsepower selection, to develop acceptability. The engines were evolved with power fitness only making up acceptability until 80% of the desired horsepower was located on an individual basis. After the

[2] Some engines will not create power or are structurally impossible, for example.

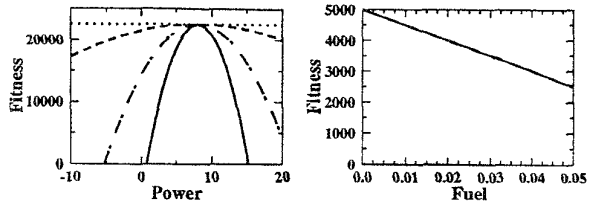


Fig. 4. Acceptability Functions. Fuel is not included until 80% of the power seeked is found. Power Fitness = $15 * (1500 - \frac{((PowSeek - PowDev) * 15)^2}{(\frac{NumGen}{CurGen+1})^3})$. Fuel Fitness = $5000 - (Fuel * 50000)$

80% threshold is reached, two-point crossover was incorporated on all crossover events since it has a greater ability to minimize fuel and the evolution power could no longer be halted. Fuel minimization fitness was phased in as a bonus on an individual basis to those internal combustion engines achieving 80% of the desired horsepower. The 8 HP engine consumes a minimal amount of fuel based upon the model and acceptability given.

The fuel flow is very low, perhaps because parameters such as friction and inertia were omitted. While the exhaust, intake, and transfer ports may seem either short or long, methods described in Section V can be used to prevent these occurrences. Additionally, the transfer tube's length could be derived from the size of the crank and the height of the port. Another problem is the fact that my model does not allow unused fuel to combine then escape with the combusted fuel in this model and that a constant enthalpy of combustion is used (the internal energy added to the head at the time of combustion should be directly based on pressure). Any fuel that enters the head is used for combustion and no fuel is lost out the exhaust, up to the total mass of fuel available in the head. This can be corrected through a more complex fuel count.

Rotation 1	Power - -1.003 HP, Fuel - 0.000571 kg/sec
Rotation 2	Power - 8.991 HP, Fuel - 0.00493 kg/sec
Rotation 3	Power - 9.061 HP, Fuel - 0.00201 kg/sec
Rotation 4	Head Imploded: Power - NaN, Fuel - NaN
Rotation 5	Backflow – No Fuel in Head:

TABLE IV

A POSSIBLE SIMULATION RESULTING FROM ENGINES CREATED WITH ONLY THREE ROTATIONS OF CRANK.

The first rotation omitted combustion to allow the model to determine initial fuel in the head. Eight rotations of the crank were used to find steady-state within the engine. Without this allowance, GABSys is able to find engines that do not work but are acceptable based on fewer crank rotations as shown in the simulation listed in Table IV. While this engine does not in fact reach steady-state within the eight rotations of the crank, a similar evolved engine shown in Figure 7 does. Solving this problem may be as simple as the use of pattern matching the simulation to extract steady-state conditions.

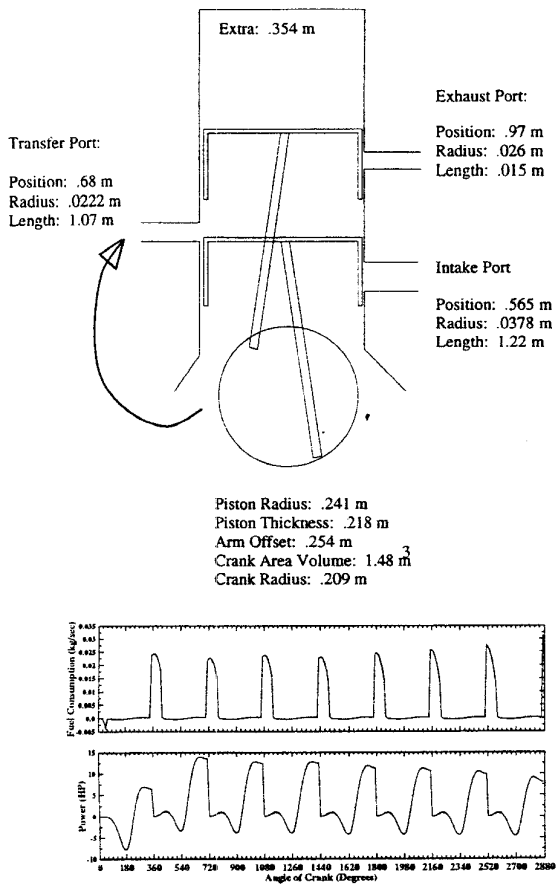


Fig. 5. Scaled representation of a 8 HP evolved engine with Fuel and Power development.

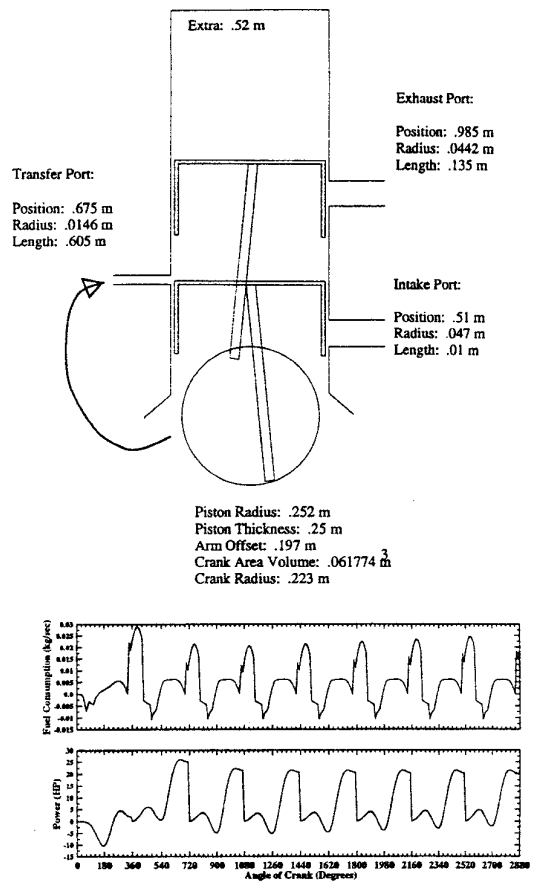


Fig. 6. Scaled representation of a 20 HP evolved engine with Fuel and Power development.

A. *Concerns about the Engine*

The insight gained about the engine from this optimization technique proved to be very useful in achieving a final design. There were some issues worth noting, since they reflect general concerns with designing from bond graphs.

1. Overall engine size
2. New Engine Form

While the form of the engines being created is correct, the overall size of this engine is much larger than found in constructed engines. This occurred because the bond graph did not include all possible behaviors. Also, the equation used to define the power developed within the engine, $Force = Pressure * Area$, could lead to both greater piston radius and constant enthalpy of combustion, which in turn could lead to larger head volume. The greater piston radius increases the ability of the combustion pressure to produce power due to the greater area the pressure is placed upon and the greater head volume reduces the power losses while the piston is rising from bottom dead center to top dead center. By adding friction, stress, and inertia of engine pieces, more realistic sizes would be expected. Alternately, engine size could be included in the chromosome.

In early tests of the system, it was found that the position of the intake port was placed above the transfer port. To avoid this new engine form, we added a heavy penalty, essentially forcing the intake port to be below the transfer port. The new engine discovery occurred because GABSys exploited the inherent bond graph error caused by the linearization of power over an episode. By having the intake port over the transfer port, the pressure in the head is increased while the pressure in the crank area is decreased, both to the extremes available within the engine created. This extreme pressure difference allows the simulation to slightly diverge in the head and crank which results in increased error over a crank rotation. To correct for this problem, a check for a compression ratio of 6:1 in the head prior to combustion was used to limit the divergence. Under closer examination, a variable enthalpy of combustion may correct this problem as well. It is also possible to eliminate such engines by increasing the simulation run to determine whether they would explode (a consequence of this engine form).

V. Sizing a Dynamic System

This section discusses ways to optimize systems around specified parameters to the same set of requirements. This is important because it is usually unnecessary optimize all the parameters over their full ranges.

Suppose some dimensions, such as the length of the exhaust and/or radius of the crank, have a minimum or forced value. There are at least four ways to handle such constraints.

1. Remove the variable from the chromosome, giving the parameter a constant value. This effectively sizes the engine to a specific part or even several parts.
2. Place a range in the chromosome that would only allow a part comparison check between a minimum or maximum value.
3. Place a bonus (or penalty) on the fitness of systems complying (or not complying) with the requirements. This is less effective since it does not provide the genetic algorithm an opportunity to learn how to improve – either it is improved or it is not. Bonuses and penalties are best used only to help direct the search.
4. Adjust the parameters artificially with the fitness function. This adds another level of complexity to the acceptability calculations and is not recommended. Balancing fitness functions within the acceptability calculations is difficult, since one fitness function could potentially halt the progress of the other.

Figure 7 describes the engine sized for a 10 cm crank radius by incorporating the same method used in the previous total optimization example and the first method of sizing, a constant valued parameter. By fixing the cam radius, the parameter is removed from the chromosome. Other parameters can also be incorporated into this model, as well. If desired, these alternatives could also be directly represented in the chromosome. For example, a search for the optimal cam speed for a specific fuel can be created. With a more complete mathematical model of the engine, the developed engine geometry could be very different. If friction were accounted for, it would most likely affect the radius of the piston.

VI. Conclusion

In this paper, we have discussed development of an internal combustion engine using genetic algorithms. We found that it was possible to identify well-behaved engines meeting complex criteria. Although there is no guarantee that the result is optimal, the resulting engine is still very impressive considering that after simulating fewer than 3,000,000 designs from a domain of $1.329 * 10^{36}$, the genetic algorithm selects a constructable, fuel-efficient engine design.

The underlying model for our system, based on Margolis' work at UC Davis, was not intended to simulate all aspects of a combustion engine. For example, the model omits heat transfer and stress and assumes the use of a constant cam speed, 3000 RPM. Therefore, neither the mass of the cam and piston nor the friction within the motor can be calculated. Hence, the inertia of the engine sections, a very important part of the engine dynamics, is excluded. These behaviors could, however, be inserted into the graph to obtain a more complex behavior description. However, the model is sufficiently complex to illustrate the utility of our system.

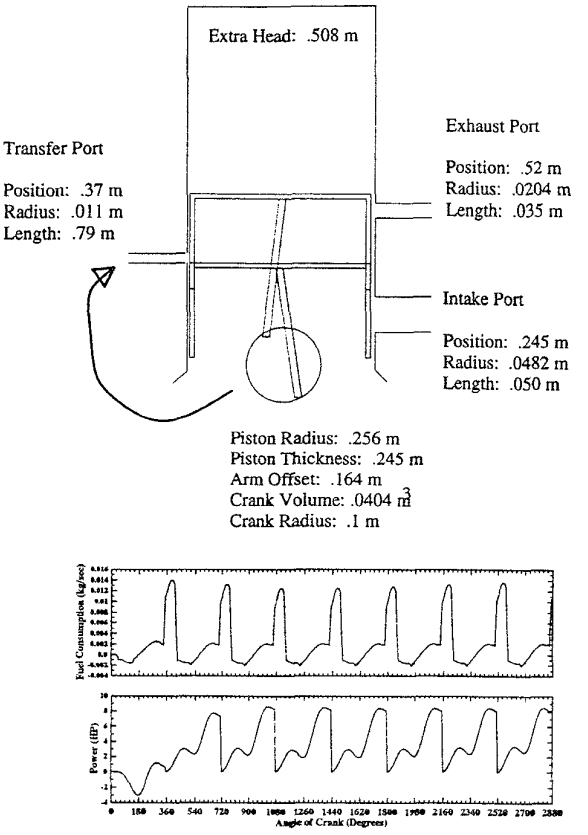


Fig. 7. 8 HP Engine with Minimized Fuel Flow

References

[1] Dean C. Karnopp R. C. Rosenberg, *McGraw-Hill Series in Mechanical Engineering: Introduction to Physical Systems Dynamics*, McGraw-Hill, 1983.

[2] D. Margolis, "Modeling of two-stroke internal combustion engine dynamics using the bond graph technique," in *AE (Society of Automotive Engineers) Transactions*, 1975.