Miniature Atomic Vapor Cell Temperature Controller

Doug Bopp[1]
[1]National Institute of Standards and Technology, CO 80305 USA

Atomic vapor cells, used as spectroscopic references, require thermal stability in order to reduce long-term systematic drifts. Chip scale atomic clocks require low power consumption and to this end a thermal control system is developed to meet three demands: 1) have zero steady state error, 2) minimize the settling time below 10 s, and 3) minimize the peak overshoot to below 10%. Straightforward modeling of a thermal system as an electrical circuit and application of control theory provides a control system with zero steady state error, 8 second settling time, and a peak overshoot of 20%. To improve the system behavior, a genetic algorithm is implemented in order to search the parameter space of the controller such that the transient response is improved. This optimzation method produces a set of digital PID parameters such that there is zero steady state error, the settling time is 1 second, and the peak overshoot is approximately 5%.

I. Introduction

THIS paper will outline the development of a thermal model for a miniature vapor cell, application of control analysis to develop a controller that meets the constraints, and optimization of the parameters using a genetic algorithm.

Atomic vapor cells typically rely on the high vapor pressure of alkali metals to generate a gaseous sample suitable for spectroscopic interrogation. This vapor pressure is proportional to temperature which makes it necessary to temperature stabilize the system and bring the steady state temperature error to zero. Additionally, chip scale atomic clocks (CSACs) typically use more power than is available in a portable system to run continuously so, in order to meet stringent power consumption demands, CSACs are opertaed periodically to stabilize the long term drift of a local oscillator. These two goals set three main specifications: 1) minimize the steady state error, 2) achieve a maximum settling time of 10 seconds, and 3) achieve a maximum overshoot of 10%.

The system under study is a microfabricated vapor cell in the form of a Si cube with inner dimension of 1 mm x 1 mm x 1mm and outer dimensions of 1.5 mm x 1.5 mm x 1 mm. The top and bottom of the cell is capped with pyrex windows with dimension of 0.5 mm x 1.5 mm x 1.5 mm. This cell then has small, planar 1 mm x 1 mm x 0.5 mm ITO heaters attached to both windows. The assembly is then suspended by polyimide wires with dimensions of 0.1 mm x 0.025 mm x 1 mm long with gold traces overlayed with dimensions of 20 μm x 2 μm x 1 mm long. This whole assembly is then mounted into a vacuum packaged assembly. A cartoon of this device is shown in Figure I.

II. Modeling

A. Thermal Circuits

Thermal systems can be modeled quantitatively using lumped system analysis when the thermal gradients inside of a thermal element is negligible [5]. This approach has the benefit of providing an equivalent circuit for the system to create a quantitative transfer function which is useful for designing a controller.

A thermal system is typically composed of heat sources and sinks, thermal masses, and coupling between thermal bodies.

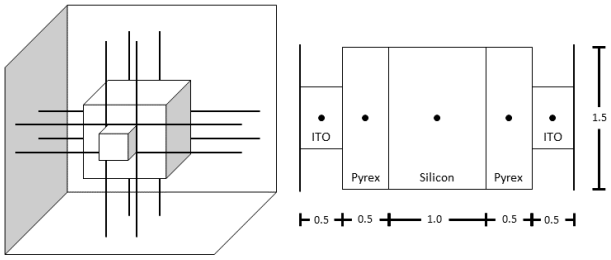

Fig. 1. Left) Depicts a cartoon of the atomic vapor cell Right) depicts a side view of the sandwiched items

The main parameter of interest is the temperature of discrete elements in the system. Heat flows between these points in one of three modes: conduction, convection, and radiation. The thermal system of equations for all nodes is essentially composed of a simple balance equation:

$$\text{Heat In} = \text{Heat Out} + \text{Heat Stored}. \tag{1}$$

Heat In is characterized as a heat source or sink and has units of power. Heat Out is typically proportional to the difference in temperature between two points. Heat Stored is proportional to the time rate of change of temperature and the thermal heat capacity of the system. When compared to a thermal mesh, equation 1 takes the form

$$\sum_{j=1} q_j = \sum_{i=j} \frac{(\theta_i - \theta_j)}{R_{ij}} + C_i \frac{d\theta_i}{dt} \tag{2}$$

where the i subscript indicates the ith node, R_{ij} is the nearest neighbor coupling, q_j are the heat sources and sinks coupled to the ith node, and C_i is the thermal capacitance of the ith node. Equation 2 clearly maps to electrical systems by making the following substitutions: heat sources are constant current sources; temperature differences are voltages; thermal resistances are electrical resistances; thermal capacitances are electrical capacitances. The lumped element approach necessarily trades some accuracy for manageability and becomes more valid in the limit that the thermal gradients within each element tend to zero. This is a fair assumption when the

lumped bodies have good thermal transport internally relative to the thermal transport between lumped bodies; further justification can be drawn from electrical circuit theory.

In the system under study, there is negligible convection due to vacuum packaging. Conduction is given by material and geometrical factors:

$$R_{ij} = \frac{L_{ij}}{Ak_i} \tag{3}$$

where L_{ij} is the distance between two points, A is the area of the surface in contact, and k_i is the thermal conductivity of the material. If nodes connect two dissimilar materials, then the couplings must be broken down into equivalent series networks where the total coupling is equal to the sum of couplings arising from moving through a single material. The electrical analogy to thermal systems extends to concepts such as series and parallel resistances and generalizes well.

The third mode of thermal transport, radiation, is inherently nonlinear as seen from the Stefan-Boltzmann Law for transfer between two grey bodies:

$$q_{rad} = \frac{A\sigma(T_1^4 - T_2^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} \tag{4}$$

where A is the area of the emitting surface, ϵ_i is the material emissivity (a value between 0 and 1), σ is the Stefan-Boltzmann constant ($\sigma = 5.67 * 10^{-8} \frac{[W]}{[m]^2[K]^4}$), T_i is the temperature of a given material surface. Attempts to linearize this equation through a Taylor expansion about the operating point (in this case 353 K) yields a constant power loss and a linear coupling to a bath at the operating temperature. Additionally, there is a shape factor that must be taken into account to properly account for radiation from a non planar object which introduces further uncertainty into these equations. Thus, for simplicity, thermal radiation will be modeled as a constant heat sink at the operating point.

The system under study is comprised of several materials sandwiched together. Each of these materials have separate thicknesses and thermal capacities and thus the entire coupled system is equivalent to a seris of cascaded systems. Due to the time dependence of the temperature being proportional to nodal couplings and heat capacities, the overall system order increases linearly with the number of cascaded elements. Large system orders can be reduced into a series of terms through partial fraction reduction and then inverse laplace transforms yield decaying-exponential terms. The system will have a dominant pole which will describe the vast majority of the behavior of the system and it is this dominant pole which this analysis attempts to deduce. Thus, simplifications in the system model into a single dominant pole (equivalently, a first order, linear system) will yield the gross behavior of the system and provide a useful system transfer function to design a controller.

B. Building the Thermal Circuit

The system will be modeled as four coupled elements: a heat source, a thermal capacitance, a heat sink, and a thermal resistance. The vacuum packaging casing will be assumed to be at ambient temperature, the thermal mass of all bodies will be added together, and the thermal resistance will be modeled in greatest detail.

$$C_{eff} = \sum c_i m_i = \sum c_i \rho_i V_i \tag{5}$$

The effective capacitance of the system is dominated by the vapor cell itself; the volume of the polyimide and gold suspsension is approximately 1000 times less than that of the central volume. This makes it reasonable to neglect the thermal capacitance of the suspension wires.

$$q_{rad} = \frac{A\sigma(353^4 - 293^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} = A\frac{367}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} \tag{6}$$

Because the areas of interest are on the order of 6 mm^2, the radiated power is on the order of 2 mW for perfect black body radiators. If the interior of the vacuum packaging is coated in low emissivity material, such as a metal ($\epsilon \sim 0.02$), the radiated power is reduced by a factor of 50 or more. Thus, it is reasonable to neglect the power lost to radiation.

The final step is to determine the thermal resistances using equation 2 when accompanied by the correct dimensions. Each length of polyimide wire has resistance

$$R_{leg} = \frac{1}{\frac{1}{R_{gold}} + \frac{1}{R_{kapton}}} \tag{7}$$

with eight wires in parallel, the effective resistance to the ambient temperature is $R_{leg}/8$.

C. Plant Transfer Function

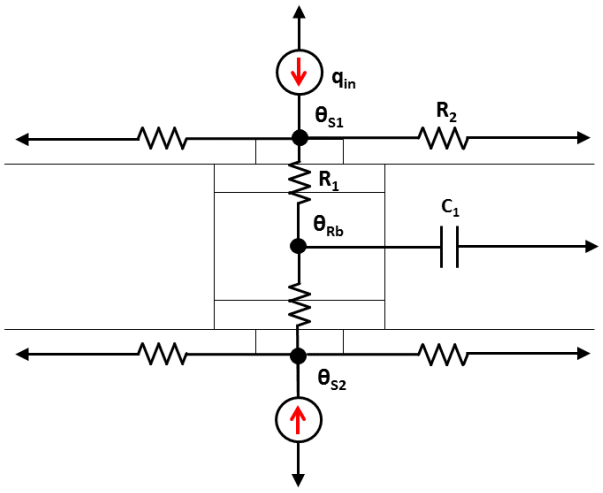

Fig. 2. Thermal System Equivalent Circuit

Figure 2 depicts the equivalent thermal circuit which overlays a simple system cartoon to show the correspondence. Simple circuit analysis solves the system in terms of $\frac{\theta_i}{q}$. In order to minimize the energy used per cycle, the amount of time the system is delivering power to the vapor cell must be minimized. This means that if the maximum power q_0 is very large, it will take very little time to bring the system up to temperature which minimizes the amount of energy used

per cycle do to a decreased amount of power lost in the same amount of time. Thus, q_0 should be made as large as feasible. Additionally, q_0 represents a saturation value of the maximum power deliverable to the system and is one of the system rails.

From the appendix, the values of the circuit diagram are listed below:

$$\begin{aligned} R_1 &= R_{Cell} \\ R_2 &= R_{Leg}/4 \\ C_eff &= \sum \rho_i V_i c_i \end{aligned} \tag{8}$$

The plant transfer function can be found with simple circuit analysis focusing on three nodes: θ_{Rb} representing the rubidium temperature and the center of the vapor cell structure and θ_{S1} and θ_{S2} representing the symmetrical temperatures of the surfaces of the vapor cell. By symmetry, $\theta_{S1} = \theta_{S2}$.

$$q - 8\frac{\theta_{S1} - \theta_a}{R_{leg}} - \frac{\theta_{S1} - \theta_{Rb}}{R_{Cell}} = 0 \tag{9}$$

Taking θ_a, the ambient temperature, equal to zero due to gauge invariance of thermal systems (neglecting radiation)

$$q - 8\frac{\theta_{S1}}{R_{leg}} - \frac{\theta_{S1} - \theta_{Rb}}{R_{Cell}} = 0 \tag{10}$$

$$C_{eff}\frac{d\theta_{Rb}}{dt} + \frac{\theta_{Rb} - \theta_{S1}}{R_{Cell}} + \frac{\theta_{Rb} - \theta_{S1}}{R_{Cell}} = 0 \tag{11}$$

$$q - \theta_{S1}\left[\frac{8}{R_{Leg}} + \frac{1}{R_{cell}}\right] + \frac{\theta_{Rb}}{R_{Cell}} = 0 \tag{12}$$

$$\theta_{S1} = \theta_{S2} = \frac{qR_{Cell} + \theta_{Rb}}{\frac{8R_{Cell}}{R_{Leg}} + 1} \tag{13}$$

$$\frac{\theta_{Rb}}{q} = \frac{\frac{R_{Leg}}{4}}{1 + sC_{eff}\frac{R_{Leg}}{8}(\frac{4R_{Cell}}{R_Leg} + 1)} \tag{14}$$

This yields a steady state gain of $q\frac{R_{Leg}}{4}$ and a characteristic time eaual to the pole $C_eff\frac{R_{Leg}}{8}(\frac{4R_{Cell}}{R_leg} + 1)$. From the dimensional values attached in the appendix this means:

$$\begin{aligned} G(s) &= \frac{K_0}{1 + Ts} = \frac{9.1}{1 + 75s} \\ T_{rise} &= 74.9s \\ G(0) &= 18.2\frac{\deg C}{mW} \end{aligned} \tag{15}$$

This rise time gives the characteristic time required for an open loop system subjected to a constant input power. The above analysis indicates that there is a thermal resistance of 9.1 degrees C per mW. This is quite large for these physics packages and inherently beats the calculations made in references [1, 2] where values of 6.28 degrees C per mW enable low powered operation of CSACs. The discrepancy comes from neglecting the effects of thermal radiation. Thermal radiation included with good black body radiators yields about 8 mW of power needed to bring the temperature up from room temperature to 80 degrees C. This is consistent with references [1] and shows that this thermal modeling is consistent with more advanced FEA techniques.

The above equation clearly indicates that the system has a single dominant pole and is first order under the assumptions made earlier in this section. This is to be expected from a thermal system and a generalized equaiton for a thermal system transfer function can be expected to be:

$$\frac{\theta_i}{U} = \frac{e^{-s*T_{dead}} * k_0}{1 + T_{rise}s} \tag{16}$$

Equation 16 incorporates a dead time where the system is not responding to the actuator input which could come from turn-on times for circuit elements and initial transient turn-on behavior. The above equation has three free parameters: a dead time T_{dead}, a rise time T_{rise}, and a gain factor k_0. These parameters can be determined experimentally by fitting a line to the transient response of the system as depicted in Figure 3.

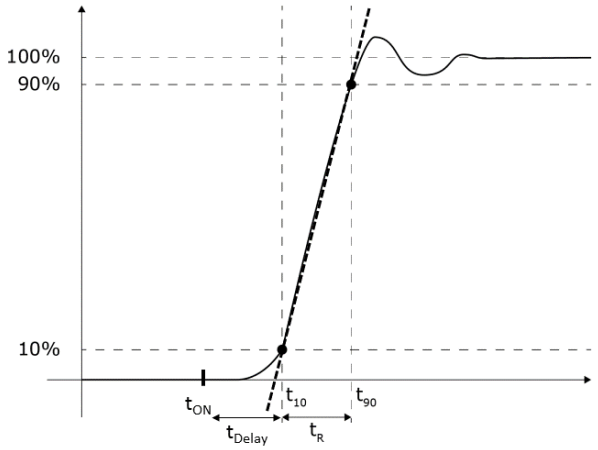

Fig. 3. Method For Determining The System Transfer Function For A First Order System

III. Controller Design

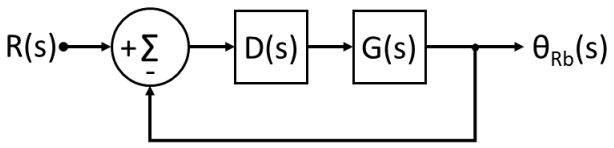

Fig. 4. System Block Diagram

Now that the plant model has been determined, it is necessary to design the controller in order to meet the specifications. The controller will be enacted through the use of a thermistor to sense the temperature of the vapor cell and a digital controller to modulate the power delivered to the vapor cell. Because the plant has a single dominant pole, this indicates that it has a specific time constant given by 75 s. This time constant represents the 1/e free decay time. Taking 10% of this characteristic time is a sufficient cycle time for the controller to update its output value but because of the speed of microcontrollers a controller update rate of 1Hz is chosen.

To approximate the continuous nature of an analog system, the digital controller will use pulse width modulation (PWM) to linearly control the power delivered to the system.

Proportional, integral, derivative (PID) control will be used as the controller model. This simple scheme is robust and capable of moving the closed loop poles to suitable locations in the s plane. The PID controller can be made either with analog electronic circuits involving passive and active components or through the use of a digital microcontroller; the latter approach is taken here. The best method for the digital control analysis is to proceed with the z tranform. MATLAB makes this simple because it is capable of transforming a function in the s plane into a function in the z plane using the straightforward command c2d(TransferFunction,SampleTime). This enables transient response analysis to be accurately modeled due to the finite update rate of the controller.

Taking 15 as the plant transfer function

The closed loop transfer function from Figure 4 is:

$$\frac{\theta_{Rb}}{q} = \frac{D(s)G(s)}{1 + D(s)G(s)} \tag{17}$$

The error between the system state and the setpoint is given by

$$E(s) = R(s) - Y(s) = R(s)\frac{1}{1 + G(s)D(s)} \tag{18}$$

In order to drive the error to zero when subjected to a step reference (e.g. upon turn-on), the controller must contain an integrator. To decrease the rise time, a proportional stage can be added to the controller. To reduce the overshoot, a differentiator stage can be added. Thus the obvious controller choice is a PID controller.

$$D(s) = K_P + \frac{K_I}{s} + K_D s \tag{19}$$

Plugging this into equation 17,

$$\frac{\theta_{Rb}}{q} = \frac{K_0\left[K_P + \frac{K_I}{s} + K_D s\right]}{1 + Ts + K_0\left[K_P + \frac{K_I}{s} + K_D s\right]} \tag{20}$$

$$\frac{\theta_{Rb}}{q} = \frac{K_0\left[K_P + \frac{K_I}{s} + K_D s\right]}{s^2(T + K_D K_0) + s(1 + K_P K_0) + K_0 K_I} \tag{21}$$

$$\frac{\theta_{Rb}}{q} = \frac{1}{T + K_D K_0}\frac{K_0\left[K_P + \frac{K_I}{s} + K_D s\right]}{s^2 + s\frac{1 + K_P K_0}{T + K_D K_0} + \frac{K_0 K_I}{T + K_D K_0}} \tag{22}$$

By manipulating the values in the denominator, it is possible to move the poles of the closed loop transfer function to desired locations. In this case, the goal criteria to beat are 1) achieve a settling time below 10 s and 2) to reduce the peak overshoot to below 10%. These are parameterized by ζ and ω_n which come from a simplified second order transfer function:

$$A(s) = \frac{B(s)}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \tag{23}$$

The modification of ω_n and/or ζ leads to predictable transient behavior [4] and is the method used here to meet the criteria.

$$\zeta \sim \sqrt{\frac{\frac{-ln(0.1)^2}{\pi}}{1 + \frac{-ln(0.1)^2}{\pi}}} \geq 0.59 \tag{24}$$

$$t_{settle} \sim \frac{3.9}{\zeta\omega_n}$$
$$\omega_n \geq 0.66 \tag{25}$$

Equating like coefficients between equation 22 and 23 using the values found in equations 24 and 25 yields

$$\begin{aligned} K_0 &= 18.2 \\ K_P &= 10 \\ K_D &= 8.77 \\ K_I &= 84. \end{aligned} \tag{26}$$

This yields a continuous closed loop transfer function which should meet the desired specifications.

IV. System Analysis

The system model and the control system have been identified and it is now important to verify the system response using a suitable numerical analysis package. MATLAB was chosen because of its wide breadth of control system analysis features but the process could be completed in any modern programming language. The s plane transfer function in equation 22 with the appropriate values is tranformed to the z plane using MATLAB. This z plane transfer function is then used to calculate the transient response, root locus, and bode plots of the system.

Simulation of the system shows that the peak overshoot is equal to 19.5 % and a settling time of 8 seconds. This fails to satisfy the first requirement and modification to the parameters given above is necessary to yield better behavior. First, however, several specifics of a digital controller must be determined.

Special care must be taken to minimize spurious feedback noise. The digital controller will incorporate a processing step to clean the signal through the use of a median filter. A median filter is a nonlinear filter which better eliminates outlying values than a traditional averaging or low pass filter. This filtering technique will reduce the impact of electrical nosie coming from the sensor.

Because the output of the controller can saturate when there are large changes in the setpoint, the controller needs a method of avoiding windup from the integrator. This is achieved through the use of both clamping and disabling the integral loop until the system is in the controllable region. This is advantageous because the thermal system should only be in one of two states: off and at ambient temperature or on and at operating temperature. No change in the reference value is currently planned but it is easy to imagine that if several modes of operation were desired then it would be advantageous to change the temperature setpoint e.g. to change the SNR of the system.

Finally, to incorporate the use of PWM as the digital controller output, a change of variables is necessary where $q(s) = M(s)q_0$ where M(s) is a function whose value is between 0 and 1. M(s) now acts as a linear controller which avoids the nonlinear nature of typical power controllers.

V. Optimization

The system so far described can be improved using a heurstic parameter search to numerically optimize the controller in order to produce better system behavior. The choice made in this text is to develop a simple genetic algorithm (GA) to provide iterative local improvements in a parameter search by minimizing a cost function to sort the individual parameter sets. Excellent resources abound on GAs [3] and only a coarse description of the technique will be given here.

1) Genetic Algorithm

A genetic algorithm is a heuristic, local parameter search whose optimization is determined through the declaration of a suitable "fitness", or cost, function. A good cost function should punish unwanted behavior as well as reward good behavior. The purpose of the GA is to minimize the cost function (maximization can lead to runaway behavior and is discouraged) by combining and modifying parameters from indiviual parameter sets whose cost is lowest. A simple recipe follows:

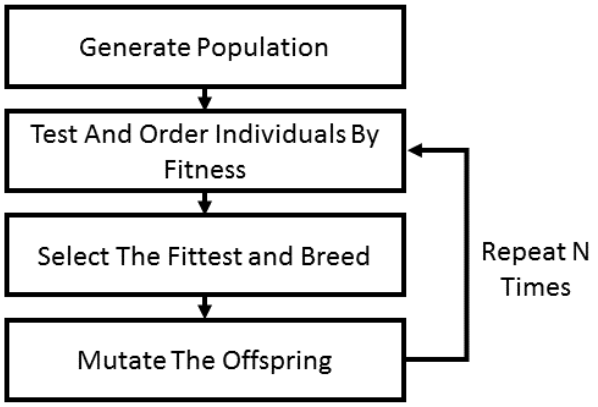

Fig. 5. Gentic Algorithm Flow Chart

Subtleties abound with respect to how many individuals to specify, what the cost function should be, method of breeding and especially the mutation rate and number of elite children. These are typically empirically tested against a well understood problem to test for convergence and multiple runs with different initial parameters are necessary.

For the system of interest, a simple GA was applied to the problem at hand, namely minimizing both the settling time and the peak overshoot. The results of this are plotted in Figure 6

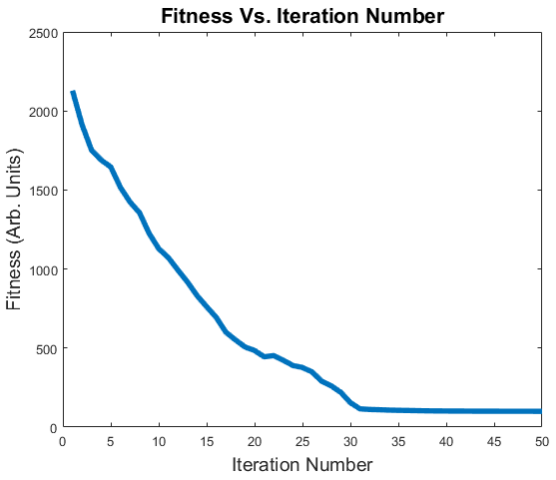

Fig. 6. Gentic Algorithm Flow Chart

where a fitness function of the form Fit $= 100T_{settle}+10M_p$ was used.

As can be easily seen in Figure 6, the GA minimizes the fitness function. The results of this optimization were then directly compared to the initially calculated values:

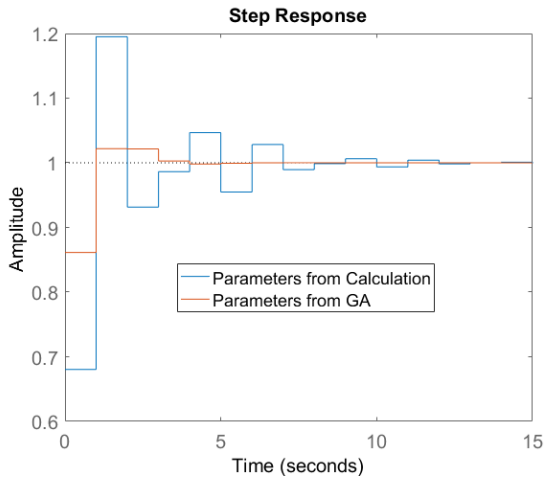

Fig. 7. Comparison of the step responses of the system using the calculated PID parameters (orange) and the GA optimized parameters (blue)

After optimization, the GA PID parameters achieved zero steady state error, a 1 s settling time, and a 5% overshoot which satisfies the specifications.

Attached in the appendix is the MATLAB code used to improve the transient response. What is perhaps more interesting is the possibility of incorporating a GA in a situation where it is intractable to calculate the transfer function of a system which is expected to be first order. In that case, a system can be developed to meausure key characteristics, such as settling time and overshoot, from an oscilloscope trace and feed that into a GA to optimize its performance in an experimental setting. Small modifications of the attached code can incorporate this added functionality and the key benefit is the technique is plant agnostic.

VI. Conclusion

This paper has outlined the basic modeling process for thermal systems using nodal analysis. Simplifications to the plant system were made in order to generate a linear model to simply understand the gross behavior of the structure under analysis. This simple linear model was used to build a control system which is composed of a PID controller implemented using a PWM digital microcontroller which linearly controls the power delivered to the system by modifying the duty cycle of a constant heat source. The PID parameters were then optimized through the use of a genetic algorithm to improve the transient response characteristics. This produced a system response with a settling time of approximately 1 s, zero steady state error and a peak overshoot of approximately 5%. These values satisfy the specifications.

Further analysis will include experimental determination of the dominant pole time constant and the characteristic delay time of the system. This will provide an empirical model which can be compared to the theoretical model to expose discrepancies between the theoretical and experimental frameworks. This improved model can then be used to design future vapor cells with optimized mechanical properties. This process aids in modifying the plant transfer function. The model used in this analysis can be made more accurate by reducing the number of simplifying assumptions and increasing the number of thermal nodes. The natural extension of this idea is to use FEA to calculate the temperature at different points and times in the structure when subjected to a heat source.

A second set of analysis will include connecting the controller to a computer loaded with a genetic algorithm where the GA will load the controlelr with PID parameters and then test the transient system response. This procedure is useful for optimizing the controller transfer function for a physical system.

Appendix A
MATLAB GA Code

Below are the commented MATLAB scripts which form the GA. MATLAB offers a global optimizaiton toolbox which incorporates a built in genetic algorithm solver (along with many other useful tools) but the cost can be prohibitive. The code attached here is meant to be easy to read and readily transferrable to other languages such as C or Python. This appendix is not meant to be comprehensive; it is merely meant to provide a functional understanding of how to build a genetic algorithm and the key attributes that change its behavior.

The first step is to generate an initial population of individuals which independently have a fixed number of attributes and descriptor parameters. For this example, there are three PID tuning parameters and there is one descriptor, namely the fitness of that set of tuning parameters. This is stored in an Nx4 matrix where N is the number of individuals and the first element of each individual is the fitness.

The next step is to arrange the individuals in the population by fitness in ascending order. The fitness function should be minimized and have a fixed lower bound e.g. in this example both the settling time and the overshoot have fixed lower bounds of 0. Once sorted, M elite indivduals are chosen from the top of the sorted list. These provide the source DNA for the next generation. From this list of elite individuals, two parents are chosen at random to breed and mix their dna. The breeding program randomly selects one of the parameters from each parent and fills it into a daughter individual. This daughter individual is then mutated by some small amount to increase the diversity of the children which effectively broadens the parameter search. Finally, this individual is added to a new population and the cycle repeats until the new population is formed. At this point, it is useful to record the average fitness of this new population in order to see how the average fitness varies with the number of cycles. This whole process is then repeated a fixed number of times or until some target fitness value is achieved.

```
%This function is meant to wrap the entire GA into a single
%callable function

function [fitvscycle,mostfit]=runGA(size,cycles,seed)
fitvscycle=[];
mostfit=[];
%pop=10.*randn(size,4)+10;
weight=10;
A=zeros(size,1);
B=seed(1)+seed(1).*randn(size,1)/weight;
C=seed(2)+seed(2).*randn(size,1)/weight;
D=seed(3)+seed(3).*randn(size,1)/weight;
pop=[A,B,C,D];
last=10000000000;
for i=1:cycles
    pop=cycle(pop)
    a=length(pop(:,1));
    avg=0;
    for i=1:a;
        avg=avg+fitness(pop,i);
    end
    avg=avg/a;
    fitvscycle=[fitvscycle, avg];
    if fitness(pop,1)<last
        mostfit=[mostfit,pullparent(pop,1)];
        last=fitness(pop,1);
    end
end
plot(fitvscycle)
end
```

```
%This function is meant to cycle through one generation
%of the genetic algorithm

function newpop=cycle(pop)
pop=sortrows(pop);
newpop=[];
elites=[];
a=length(pop(:,1));
b=a/5;
```

```

for i=1:b
    elites=[elites ; pop(i,:)];
end

newpop=elites;

for i=1:a
    pop(i,1)=fitness(pop,i);
end

%for i=1:a
%    newpop=[newpop ; 0,breed(mutate(pop(randi(a),:)),mutate(pop(randi(a),:)))];
%end

for i=1:(a-b)
    newpop=[newpop; 0,breed(mutate(elites(randi(b),:)),mutate(elites(randi(b),:)))]
end

for i=1:a
    newpop(i,1)=fitness(newpop,i);
end
newpop=sortrows(newpop);
end
```

```
%This function is meant to pull the values of the parameters,
%excluding the fitness, of one individual from the population

function parent=pullparent(pop,number)
parent=zeros(1,3);
for i=1:3
    parent(i)=pop(number,i+1);
end
end
```

```
%This function is meant to breed two individuals from the population

function daughter=breed(parent1,parent2)
daughter=zeros(1,length(parent1)-1);
dna=[parent1;parent2];
for i=1:3
    daughter(i)=dna(randi(2),i+1);
end
```

```
%This function is meant to randomly mutate the individuals
%by modifying the parameters by a scaled random number

function vector = mutate(vals)
a=10;
for i=1:3
    vals(i)=vals(i)+vals(i)*randn()/a;
vector=vals;
end
```

```
%This function is meant to test the fitness of an individual in the
%population.

function fit = fitness(population,num)
parent=pullparent(population,num);
%parent(1) is the P
%parent(2) is the D
%parent(3) is the I
a1=18.2*parent(2);
a2=18.2*parent(1);
a3=18.2*parent(3);
b1=75+18.2*parent(2);
b2=1+18.2*parent(1);
b3=18.2*parent(3);
numerator=[a1,a2,a3];
denominator=[b1,b2,b3];
sys=c2d(tf(numerator,denominator),1);
parent(1);
parent(2);
parent(3);
S=stepinfo(sys);
ts=S(1).SettlingTime;
os=S(1).Overshoot;
fit=100*ts+50*os;
population(num,1)=fit;
end
```

Appendix B
System Parameters

References

[1] R. K. Chutani, S. Galliou, N. Passilly, C. Gorecki, A. Sitomaniemi, M. Heikkinen, K. Kautio, A. Keränen, and A. Jornod. Thermal management of fully LTCC-packaged Cs vapour cell for MEMS atomic clock. *Sensors and Actuators, A: Physical*, 174(1):58–68, 2012.

[2] J. Kitching, S. Knappe, P.D.D. Schwindt, V. Shah, L. Hollberg, L.-a. Liew, and J. Moreland. Power dissipation in a vertically integrated chip-scale atomic clock. *Proceedings of the 2004 IEEE International Frequency Control Symposium and Exposition, 2004.*, 00(c):781–784, 2004.

[3] A. D. Laws, III R. Borwick, P. Stupar, J.F. DeNatale, and Y. C. Lee. Thermal and Structural Analysis of a Suspended Physics Package for a Chip-Scale Atomic Clock. *Journal of Electronic Packaging*, 131(4):041005–041005, 2009.

[4] Katsuhiko Ogata. *Modern Control Engineering*. Prentice Hall, 3rd edition, 1997.

[5] Stephen D. Senturia. *Microsystem Design*. Kluwer Academic Publishers, Boston, 3rd edition, 2001.