

فهرست

صفحه

| | | |
|---|---|--|
| ۱ | ۱ | - پارادایم معادلات دیفرانسیل..... |
| ۲ | ۲ | - پارادایم STATE_CHARTS |
| ۴ | ۳ | - نتایج حاصل از یک مثال موجود در نرم افزار متلب..... |
| ۵ | ۴ | - ضمیمه..... |

فهرست اشکال

| | | |
|---|----|--|
| ۱ | ۱ | شکل ۱: دیاگرام سیستم Incubator در پارادایم معادلات دیفرانسیل..... |
| ۲ | ۲ | شکل ۲: نمودار پایداری سیستم Incubator در پارادایم معادلات دیفرانسیل..... |
| ۲ | ۳ | شکل ۳: سیستم State_Charts در پارادایم Incubator |
| ۳ | ۳ | شکل ۴: نمودار Stateflow سیستم Stateflow |
| ۳ | ۴ | شکل ۵: روند تغییرات کولر..... |
| ۳ | ۵ | شکل ۶: روند تغییرات بخاری..... |
| ۴ | ۶ | شکل ۷: ساختار سیستم househeat در پارادایم معادلات دیفرانسیل..... |
| ۴ | ۷ | شکل ۸: خروجی ناشی از سیستم تنظیم هوای خانه..... |
| ۵ | ۸ | شکل ۹: مدل تغییر یافته‌ی سیستم househeat |
| ۵ | ۹ | شکل ۱۰: ساختار Stateflow سیستم بخاری..... |
| ۵ | ۱۰ | شکل ۱۱: نمودار دمای خانه با استفاده از stateflow |

۱- پارادایم معادلات دیفرانسیل

در این تمرین می‌خواهیم یک سیستم تهویه مطبوع را با استفاده از پارادایم معادلات دیفرانسیل شبیه‌سازی نماییم. برای این منظور بایستی در ابتدا تابع تبدیل آن سیستم را بدست آورد. بنابراین پس از تحقیق و جستجو و استفاده از منبع موجود در ضمیمه تابع تبدیل زیر بدست آمد:

$$\theta(s) = \frac{Ap}{s(s + p)}$$

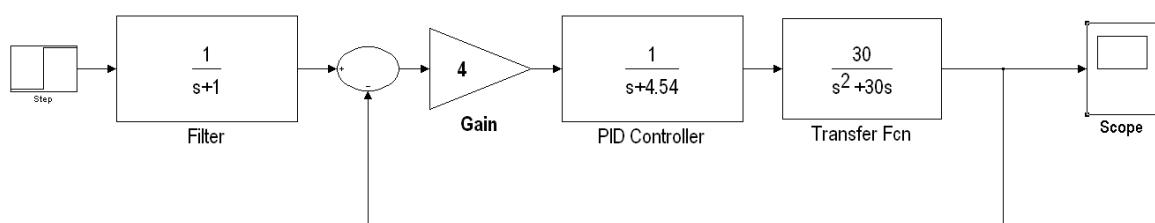
تابع تبدیل مذکور ساده شده‌ی تابع تبدیل درجه‌ی سه سیستم Incubator است که به دلیل محاسبات طولانی در ایجاد کنترلر PID و استفاده از پیش‌فیلتر و Gain می‌توان از آن استفاده نمود. برای این سیستم چون صفرهای سیستم اهمیت به سزاگی در کارکرد سیستم دارند از پیش‌فیلتر استفاده شده است. تابع کنترلر پیشنهادی برای این سیستم در نظر گرفته شده است که نشان از پایداری مطلق سیستم دارد. این کنترلر با استفاده از روش PID طراحی شده است.

$$PID(s) = \frac{1}{(s + 4.54)}$$

همچنین برای جلوگیری از نویز و اغتشاشات درون سیستم از یک پیش‌فیلتر استفاده شده است که به توصیف زیر است:

$$Filter(s) = \frac{1}{(s + 1)}$$

سپس برای شبیه‌سازی از نرم‌افزار مطلب استفاده شده است که پس از بازشدن صفحه‌ی آن و کلیک بر روی قسمت Simulink، دیاگرام آن همانند شکل ۱ رسم شد.

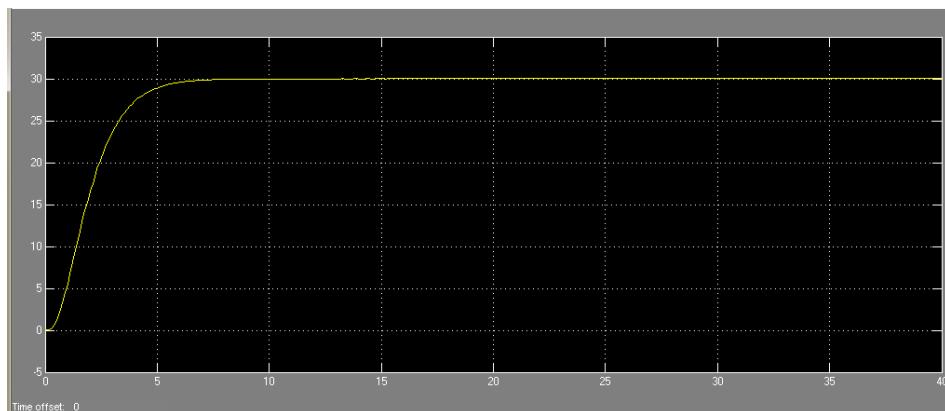


شکل ۱: دیاگرام سیستم Incubator در پارادایم معادلات دیفرانسیل

در نهایت پس از اجرای سیستم تعییه‌شده نتیجه‌ی شکل ۲ بدست آمد که نشان از پایداری سیستم مذکور دارد. برای تحلیل پاسخ گذرای سیستم‌ها که ناشی از دینامیک آن‌هاست یکی از مقولات مورد بحث در کنترل است. شکل پاسخ گذرا علاوه بر دینامیک سیستم به نوع سیگنال ورودی نیز بستگی پیدا می‌کند. از آن‌جایی که اغلب از ورودی پله استفاده می‌گردد، ضرورت می‌یابد رفتار پله‌ی سیستم‌ها مورد بحث قرار گیرد. در حالت کلی برای مشخص کردن رفتار و پاسخ به ورودی، خروجی سیستم محاسبه می‌گردد. این مستلزم صرف وقت مناسب با درجه‌ی سیستم می-

باشد. روشی که بوسیله‌ی آن‌ها بتوان رفتار تقریبی سیستم را بدست آورد، در طراحی کنترل‌ها نقش اساسی بازی می‌کند. رفتار سیستم درجه ۲ به لحاظ انعطاف‌پذیری به شدت مورد توجه قرار می‌گیرد.

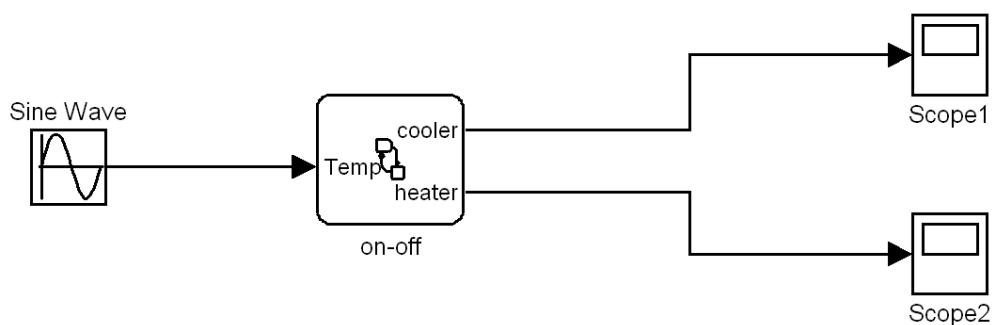
هدف سیستم کنترل رسیدن به پاسخ مطلوب است که توسط سیگنال مرجع تعریف می‌شود. در واقع کنترل واحدی است که عموماً الکترونیکی یا دیجیتالی است که به سیستم فیدبک می‌شود تا سیستم فیدبک بتواند به پاسخ مطلوب برسد. بنابراین در شکل ۲ پاسخ مطلوب سیستم به ورودی مورد نظر را نشان می‌دهد.



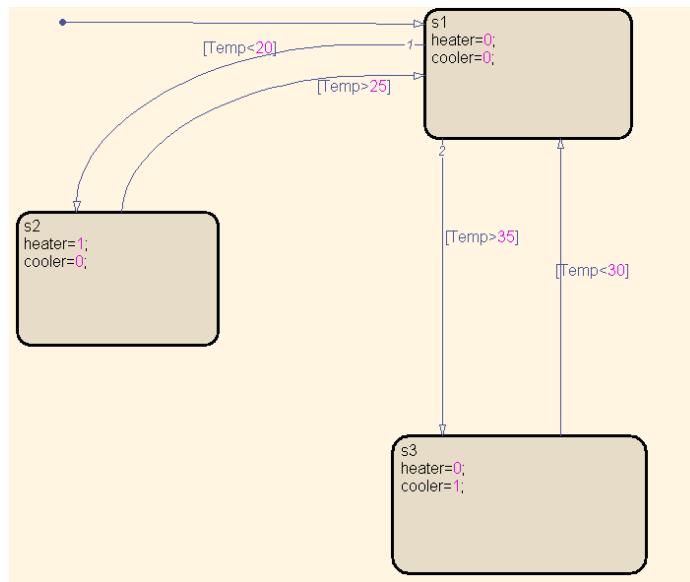
شکل ۲: نمودار پایداری سیستم Incubator در پارادایم معادلات دیفرانسیل

۲- پارادایم State_Charts

در این قسمت می‌خواهیم با استفاده از پارادایم State_Charts سیستم Incubator را شبیه‌سازی نماییم. برای این منظور بایستی از قسمت Simulink پارادایم Stateflow انتخاب گردد. حال باید سیستم مورد نظر را طراحی کنیم. برای این منظور از سه استفاده کردیم. حالت اول که حالت ورود به سیستم است بیان می‌کند که سیستم در حالت تعادل است و نیازی به روشن کردن کولر یا بخاری وجود ندارد. در حالت دوم نیز چون دما کاهش یافته بایستی بخاری را روشن کند. از طرفی در حالت سوم نیز باید کولر روشن شود.

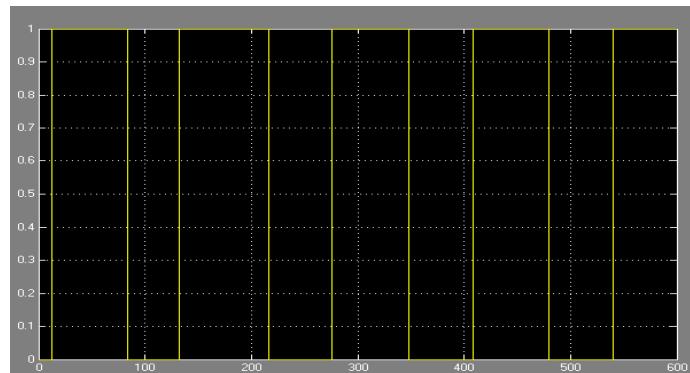


شکل ۳: سیستم State_Charts در پارادایم Incubator

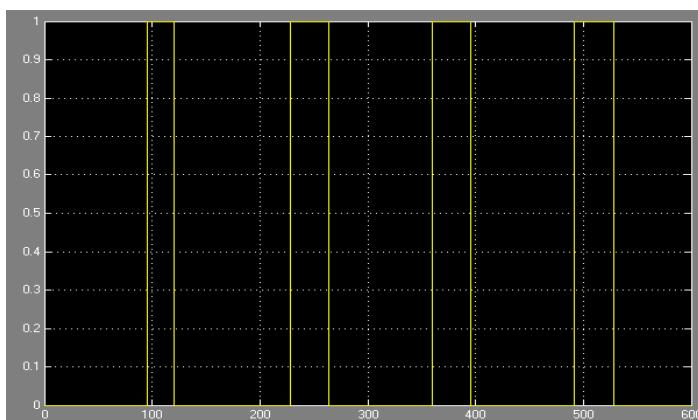


شکل ۴: نمودار Stateflow سیستم Incubator

پس از رسم سیستم مورد نظر در Stateflow در متلب نیاز به شبیه‌سازی و بدست آوردن خروجی است. در اینجا دو قسمت خروجی مربوط به سیستم بخاری و سیستم کولر است که به ترتیب در شکل ۵ و شکل ۶ نشان داده شده است. در نهایت نیز دو نمودار مذکور بر روی هم قرار داده شد تا نشان از پایداری سیستم باشد.



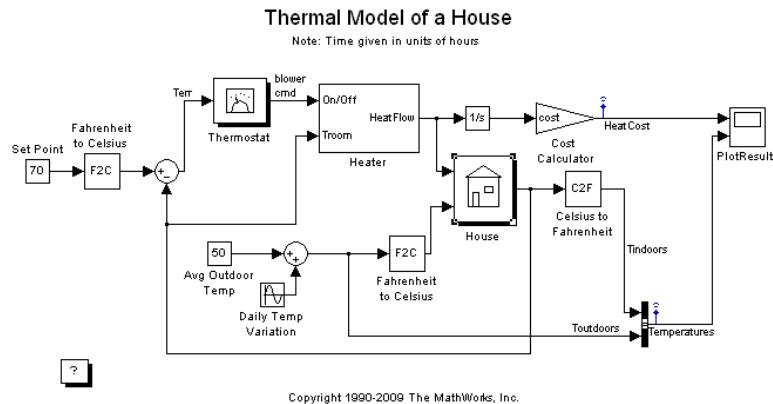
شکل ۵: روند تغییرات کولر



شکل ۶: روند تغییرات بخاری

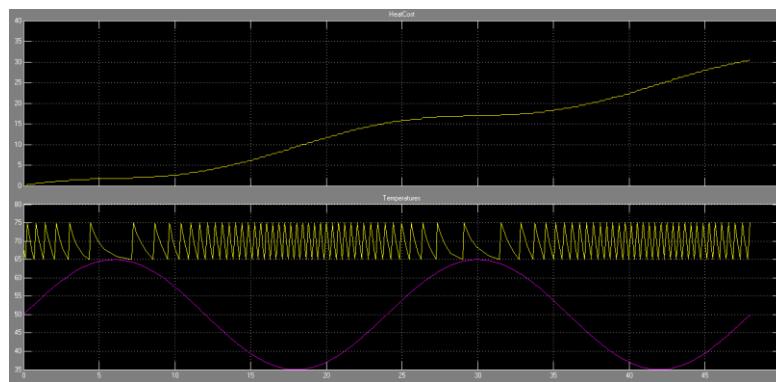
۳- نتایج حاصل از یک مثال موجود در نرم افزار متلب

در این قسمت از یکی از مثال‌های نرم افزار متلب به نام househeat استفاده شده است. در این سیستم دمای یک خانه توسط پارادایم معادلات دیفرانسیل تنظیم می‌شود. این سیستم دارای فقط یک بخاری است که در درجات مختلف عمل می‌کند و توسط یک ترموستات روشن و خاموش می‌شود. برای شبیه سازی سیستم، فرض شده است که دمای بیرون خانه به صورت سینوسی متناوب با دامنه‌ی ۲۵ درجه‌ی فارنهایت تغییر می‌کند. در شکل ۷ ساختار این سیستم به تصویر در آمده است.



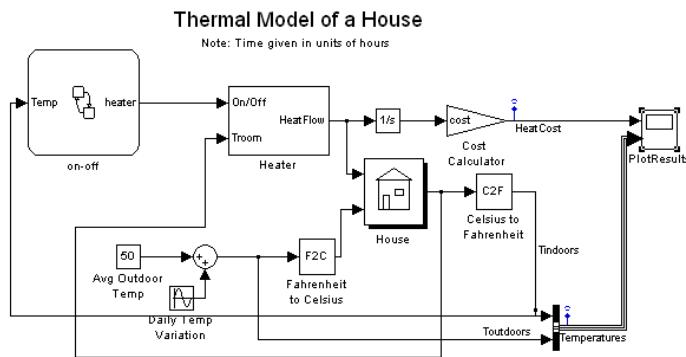
شکل ۷: ساختار سیستم househeat در پارادایم معادلات دیفرانسیل

حاصل شبیه سازی این سیستم نیز در شکل ۸ نشان داده شده است. نوسانات مربوط به سیستم در شکل پایین تر و با رنگ زرد مشخص است و رنگ بنفش نیز برای تغییرات دمای محیطی در نظر گرفته شده است. همچنین شکل بالا هزینه‌ی سیستم است.



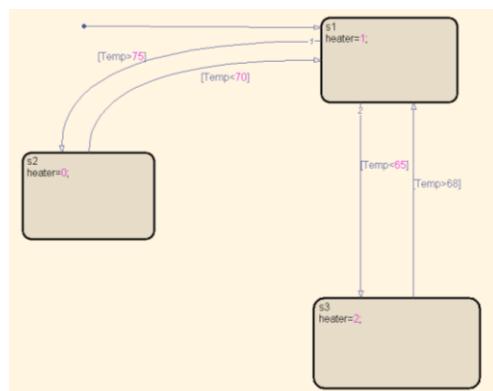
شکل ۸: خروجی ناشی از سیستم تنظیم هوای خانه

حال بایستی سیستم را با استفاده از State Flow پیاده سازی نماییم. برای پیاده سازی باید رخدادهای دمای خانه به State Charts تبدیل شود و حاصل نیز به کلید روشن/خاموشی بخاری ارسال شود. برای رسیدن به این منظور لازم است تا سیستم طراحی شده را مطابق شکل ۹ تغییر دهیم.



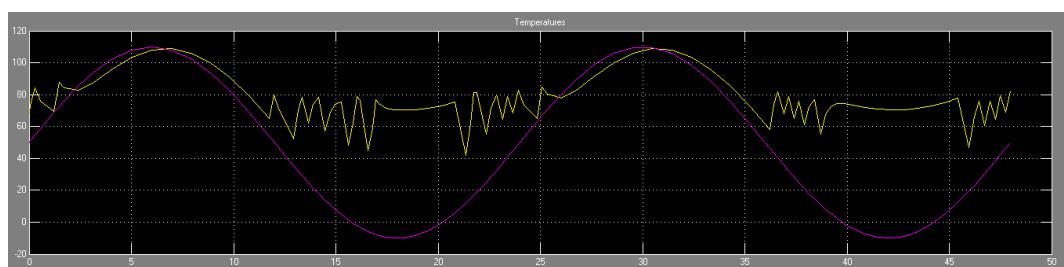
شکل ۹: مدل تغییر یافته‌ی سیستم househeat

در مدل تغییر یافته ساختار heater همانند شکل ۱۰ است. در این ساختار بخاری در دو فاز با ضرایب دمایی مختلفی کار می‌کند. همانطور که نشان داده شده است سیستم بخاری دارای ۲ سطح است که سطح یک با ضریب ۱ و سطح دو با ضریب ۲ تغییر می‌کند.



شکل ۱۰: ساختار Stateflow سیستم بخاری

حاصل شبیه‌سازی مدل تغییر یافته در شکل ۱۱ نشان داده شده است.



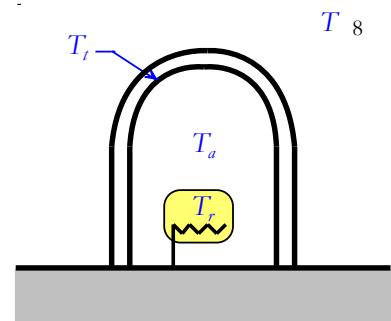
شکل ۱۱: نمودار دمای خانه با استفاده از stateflow

نتیجه‌گیری: زمانی که از پارادایم معادلات دیفرانسیل استفاده می‌شود پایداری سیستم نامناسب‌تر است زیرا که به صورت پیوسته دما را چک می‌نماید. همچنین در پارادایم state_charts با توجه به وجود حالات مختلف گستته و انتقالات بین این حالات، پایداری سیستم با گذشت زمان کمتری اتفاق می‌افتد.

Modeling the Incubator System

In the previous laboratory we implemented a simple on-off control system for the incubator, with a resistor for heating and a fan for cooling. The purpose of this laboratory is to develop a model of the incubator system for later use in creating a more sophisticated PID temperature control system. In this lab we will model only the resistive heating part of the system and ignore the fan. As we will see, a good model often requires a combination of analytical theory and experimental data.

An illustration of the incubator system is shown in the figure at right. In our model we will make one primary assumption, that the resistor, test tube, and air inside the test tube are each lumped thermal masses. In other words, the temperature is assumed constant within the resistor (T_r), within the test tube plastic (T_t) and in the air inside the test tube (T_a). We will perform energy balance calculations on each part of the system, and develop a transfer function for the entire system. The transfer function for the incubator is defined as:



$$G(s) = \frac{\text{Air temperature inside incubator}}{\text{Power input into resistor}}$$

The power input into the resistor is proportional to the PWM command from Arduino; thus, the *output* of the system is the air temperature inside the incubator and the *input* is the PWM command from Arduino. There are three temperatures of interest in this system: the temperature of the resistor, test tube and air inside the test tube (T_r , T_t and T_a , respectively). For the sake of simplicity, we will use temperature *differences* as our variables, defined as

$$\theta_r = T_r - T_\infty \quad \theta_a = T_a - T_\infty \quad \theta_t = T_t - T_\infty$$

Further, we assume that all temperatures start at T_∞ , so that all of the temperature difference variables start at zero. From the above definitions we see that

$$\frac{dT_r}{dt} = \frac{d}{dt}(\theta_r + T_\infty) = \frac{d\theta_r}{dt}$$

and

$$T_r - T_a = (\theta_r + T_\infty) - (\theta_a + T_\infty) = \theta_r - \theta_a$$

4-1-1- Resistor

Using the lumped temperature approach for the resistor, the energy balance can be written

$$\sum \dot{E}_{in} - \sum \dot{E}_{out} = m_r C_{pr} \frac{d\theta_r}{dt}$$

The power input into the resistor is

$$\sum \dot{E}_{in} = \dot{q} = \frac{V^2}{R}$$

and power leaves the resistor through convection

$$\sum \dot{E}_{out} = h_r A_r (\theta_r - \theta_a)$$

Thus, the energy balance is

$$\dot{q} - h_r A_r (\theta_r - \theta_a) = m_r C_{pr} \frac{d\theta_r}{dt}$$

define:

$$m_r C_{pr} = C_r \quad h_r A_r = \frac{1}{R_r}$$

then

$$\dot{q} - \frac{1}{R_r} (\theta_r - \theta_a) = C_r \frac{d\theta_r}{dt}$$

Taking the Laplace transform of this equation and rearranging gives

$$\left(C_r s + \frac{1}{R_r} \right) \Theta_r - \frac{1}{R_r} \Theta_a = \dot{Q}$$

Remember that \dot{Q} is the power input into the system; i.e. the denominator of the transfer function.

4-1-2- Test Tube

The energy balance for the test tube plastic is

$$\sum \dot{E}_{in} - \sum \dot{E}_{out} = m_g C_{pg} \frac{d\theta_g}{dt}$$

No energy is developed within the test tube, and it gains (and loses) heat through convection only.

$$h_g A_g (\theta_a - \theta_g) - h_\infty A_g \theta_g = m_g C_{pg} \frac{d\theta_g}{dt}$$

using

$$m_g C_{pg} = C_g \quad h_g A_g = \frac{1}{R_i} \quad h_\infty A_g = \frac{1}{R_o}$$

we have

$$\frac{1}{R_i}(\theta_a - \theta_g) - \frac{1}{R_o}\theta_g = C_g \frac{d\theta_g}{dt}$$

Taking the Laplace transform and rearranging

$$\left(C_g s + \frac{1}{R_i} + \frac{1}{R_o}\right)\Theta_g - \frac{1}{R_i}\Theta_a = 0$$

4-1-3- Air Inside Test Tube

Energy is not generated in the air inside the test tube, and again it gains and loses heat through convection

$$h_r A_r (\theta_r - \theta_a) + h_g A_g (\theta_g - \theta_a) = m_a C_{pa} \frac{d\theta_a}{dt}$$

using

$$m_a C_{pa} = C_a$$

we have

$$\left(C_a s + \frac{1}{R_r} + \frac{1}{R_i}\right)\Theta_a - \frac{1}{R_i}\Theta_g - \frac{1}{R_r}\Theta_r = 0$$

We can collect the three equations into a single matrix expression

$$\begin{bmatrix} C_r s + \frac{1}{R_r} & -\frac{1}{R_r} & 0 \\ -\frac{1}{R_r} & C_a s + \frac{1}{R_r} + \frac{1}{R_i} & -\frac{1}{R_i} \\ 0 & -\frac{1}{R_i} & C_g s + \frac{1}{R_i} + \frac{1}{R_o} \end{bmatrix} \begin{Bmatrix} \Theta_r \\ \Theta_a \\ \Theta_g \end{Bmatrix} = \begin{Bmatrix} \dot{Q} \\ 0 \\ 0 \end{Bmatrix}$$

Careful observation will show that the same rules for constructing the matrix apply for thermal systems as held for mass/spring/damper systems.

5- Parameter Extraction for the Incubator

One approach we might take to modeling the system would be to estimate values for the thermal resistances and capacitances, and use Matlab/Simulink to calculate the time response of the system to a given power input. This approach generally works well for mechanical and electrical systems, but parameter estimation for thermal systems can be more problematic. There are, in general, no closed form solutions for convective heat transfer coefficients from arbitrary shapes, so our estimates for thermal resistances may have large errors.

An alternative approach is to take measurements in the actual system, and use the results to estimate system parameters. From the above matrix equation, we can see that the transfer function is third order, and can be written

$$\frac{\Theta_a}{\dot{Q}} = \frac{b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0}$$

where

| | |
|---|--|
| $b_1 = C_g R_i R_o$ | $b_0 = R_i + R_o$ |
| $a_3 = C_a C_r C_g R_r R_i R_o$ | $a_1 = (C_a + C_r)(R_i + R_o) + C_r R_r + C_g R_o$ |
| $a_2 = C_a C_r (R_r R_i + R_o R_i) + C_r C_g (R_r R_o + R_i R_o) + C_a C_g R_i R_o$ | $a_0 = 1$ |

Instead of computing each coefficient analytically, we will see if we can estimate them using experiment. If the power input is a step function, then the Laplace transform of the response is

$$\Theta_a = \frac{b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0} \cdot \frac{1}{s}$$

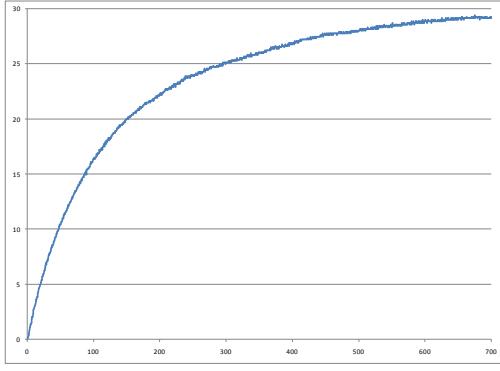
which can be transformed using partial fractions into the form

$$\Theta_a = \frac{r_0}{s} + \frac{r_1}{s + p_1} + \frac{r_2}{s + p_2} + \frac{r_3}{s + p_3}$$

The inverse Laplace transform of this expression is

$$\theta_a(t) = A(1 - c_1 e^{-p_1 t} - c_2 e^{-p_2 t} - c_3 e^{-p_3 t})$$

where we assume that all poles have negative real parts; i.e. that the system is stable. It may happen that some of the exponentials decay much slower than the others – that is, their poles have a smaller negative real part (a small negative exponent translates into slow decay). These are known as *dominant poles*, because they tend to overshadow the other poles.



One way the dominant pole phenomenon may manifest itself can be seen in the chart above, which is actual data taken from an incubator system. At first glance, the data appears to closely follow an exponential curve of the form

$$\hat{\theta} = A(1 - e^{-pt})$$

where $\hat{\theta}$ is the estimating curve, A is the asymptotic limit approached by the curve and p is the rate of decay for the exponential. Let us now see if we can determine the constants A and p using the data in the plot above.

5-1-1- Least-Squares Method for Curve-Fitting

The first constant, A , is simple to determine – it is the value that the curve approaches as t grows large. In this case

$$A = 29.43^{\circ}\text{F}$$

To find p is slightly more involved. First, it will prove convenient for us to transform the above equation into a linear relation involving p and t . Rearrange the equation to form

$$e^{-pt_i} = \frac{A - \hat{\theta}(t_i)}{A}$$

and take the natural logarithm of both sides

$$pt_i = -\ln \frac{A - \hat{\theta}(t_i)}{A}$$

Define

$$\hat{y}_i = -\ln \frac{A - \hat{\theta}(t_i)}{A}$$

Then we have the linear relation

$$\hat{y}_i = pt_i$$

It is very important to distinguish between the estimator function, $\hat{\theta}(t_i)$, and the actual datapoints, θ_i . We may use the same procedure to transform the datapoints into a linear space

$$y_i = -\ln \frac{A - \theta_i}{A}$$

The *error* between the estimated curve and the actual datapoints at each time t_i is

$$\varepsilon_i = y_i - \hat{y}_i$$

$$\varepsilon_i = y_i - pt_i$$

We are actually more interested in the magnitude of the error, rather than its sign, so we commonly calculate the *square* of the error

$$\varepsilon_i^2 = [y_i - pt_i]^2$$

To obtain the best estimate for p we should minimize the total error. In other words, we want to find p such that

$$\sum_{i=1}^N [y_i - pt_i]^2$$

is minimized. To do this, we can differentiate the above expression with respect to p .

$$\frac{d}{dp} \sum_{i=1}^N [y_i - pt_i]^2 = \sum_{i=1}^N -2t_i [y_i - pt_i] = 0$$

Simplifying, we have

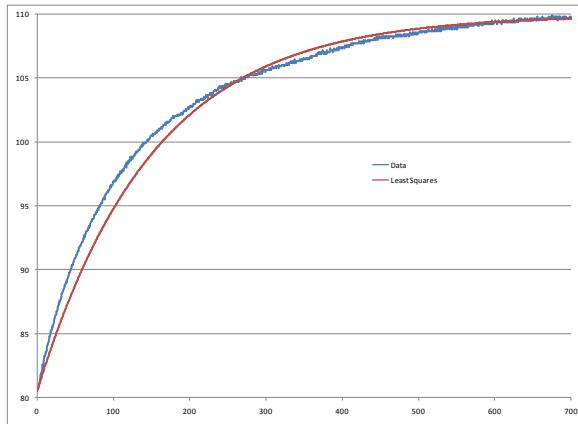
$$\sum_{i=1}^N t_i y_i - \sum_{i=1}^N t_i^2 p = 0$$

But the factor p may be brought outside the summation so that

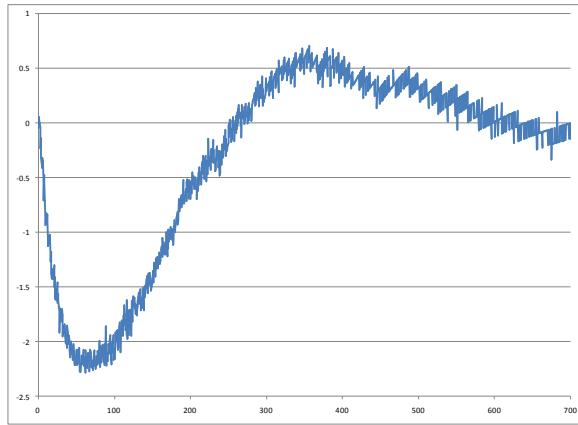
$$p \sum_{i=1}^N t_i^2 = \sum_{i=1}^N t_i y_i$$

or

$$p = \frac{\sum_{i=1}^N t_i y_i}{\sum_{i=1}^N t_i^2}$$



The results of our calculations are shown in the figure above. As you can see, the least squares curve follows the data reasonably well, but has a noticeable error between 100 and 200 seconds. This is a result of trying to fit a fourth order system to a first order model.



5-1-2- Using Excel Solver to Fit Data to a Curve

If we plot the *error* between our least-squares estimate and the measured temperature, we obtain the plot shown above. At first glance, it resembles a decaying sinusoid of the type

$$y(t) = B e^{-\zeta \omega_n t} \sin \omega_d t$$

where ζ is the damping ratio, ω_n is the natural frequency and ω_d is the damped natural frequency. This fits in well with our prediction of an exponential solution, since (using Euler's identity) we can easily switch between sinusoids and complex exponentials. In fact, if two of the poles have the form

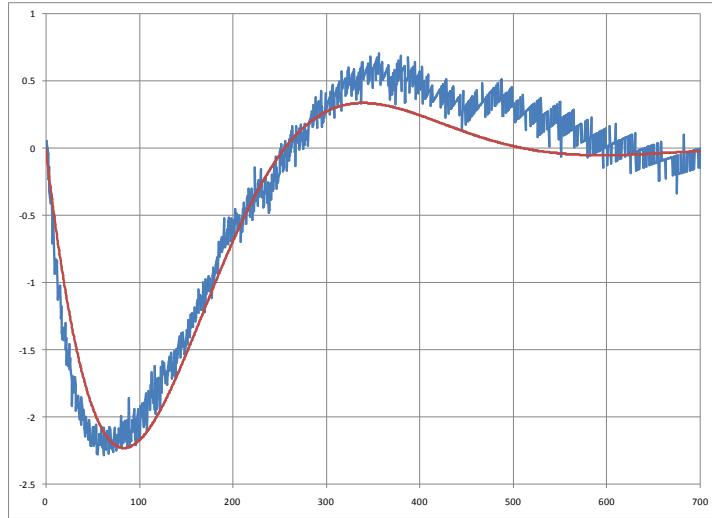
$$\begin{aligned} p_3 &= -\zeta \omega_n + j\omega_d \\ p_4 &= -\zeta \omega_n - j\omega_d \end{aligned}$$

then we can write

$$\begin{aligned}
 e^{(-\zeta\omega_n+j\omega_d)t} - e^{(-\zeta\omega_n-j\omega_d)t} &= e^{-\zeta\omega_n t} e^{j\omega_d t} - e^{-\zeta\omega_n t} e^{-j\omega_d t} \\
 &= e^{-\zeta\omega_n t} (e^{j\omega_d t} - e^{-j\omega_d t}) \\
 &= 2e^{-\zeta\omega_n t} \sin \omega_d t
 \end{aligned}$$

Thus, part of the system response may include a decaying sinusoidal term.

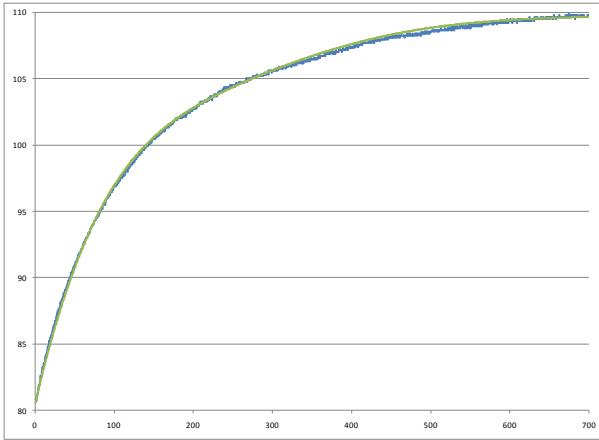
We are now faced with determining constants B , ζ and ω_n that minimizes the difference between the error plotted above and our decaying sinusoid expression. There is no easy way to linearize this expression so as to use the least-squares method, so we choose to use the “Solver” feature in Excel to find the best values.



The error, along with the best-fit curve is shown above. The total expression for the step response of the incubator is

$$\theta(t) = A(1 - e^{-pt}) + Be^{-\zeta\omega_n t} \sin \omega_d t$$

Plotting this expression against the measured data results in the plot shown below. As you can see, the approximation is quite close to the actual data, and we can have some confidence in the accuracy of our model.



6- Transfer Function of the Incubator System

Up until now we have modeled the incubator as a third order system. It is clear, however, that the first order model does a pretty good job of approximating the incubator's behavior, and will probably be good enough for designing a control system. In the first order model, the step response of the incubator is

$$\theta(t) = A(1 - e^{-pt})$$

The Laplace transform of this function is

$$\Theta(s) = \frac{Ap}{s(s + p)}$$

This is the response of the system to a step input of magnitude A . Thus, the input to the system (in the Laplace domain) is

$$\dot{Q} = \frac{A}{s}$$

and the estimated transfer function for the system itself is

$$\frac{\Theta}{\dot{Q}} = \frac{p}{s + p}$$