HW 2: Model Learning for Smart Home Thermal Management

Due: Friday February 19, 2016 at 5:00pm PT

This assignment will provide hands-on practice for parameter identification with application to smart home thermal management. The assignment is organized in a tutorial fashion, thereby allowing you to practice parameter identification on a relevant real-world energy system example.

Problem 1: Reading
Read the article on bCourses

- K. Bradbury, "Energy Storage Technology Review," Tutorial, pp. 1-34, 2010.

Clearly, there is not one "silver bullet" technology for energy storage. Each has its respective advantages and disadvantages. In your submission, please populate the following table with two or three bullet points for the pros and cons of each energy storage technology

Table 1: Energy Storage Technolog Pros and Cons

Energy Storage Tech	**Pros**	**Cons**
Pumped Hydro Storage (PHS)		
Compressed Air Energy Storage (CAES)		
Flywheels		
Electrochemical Capacitors		
Superconducting Magnetic Energy Storage (SMES)		
Lead Acid Batteries		
Lithium-ion Batteries		
Flow Batteries		

Background
Suppose you work at a smart home energy management company that seeks to make home HVAC systems intelligent. To develop this technology, you must "learn" the dynamics of each unique home. More specifically, we consider a generic format shown in Fig. 1. The home transfers heat between the interior (modeled as a lumped thermal mass) and the environment. The home also transfers heat between the interior and boiler pipes distributed throughout the house. The home interior, environment, and boiler all have time-varying temperatures given by $T(t), T_A(t), T_B(t)$, respectively. Consequently, it's easy to verify the interior

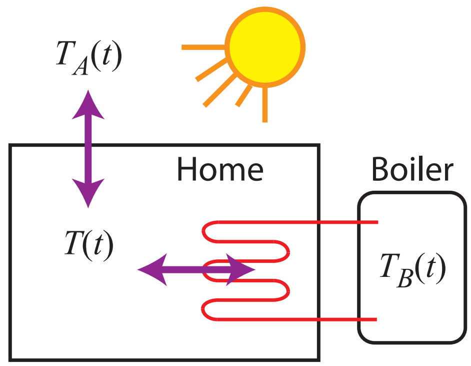

Figure 1: Diagram of heat transfer in a smart home. Heat is transferred between the home and outside environment, and between the home and boiler pipes.

temperature dynamics evolve according to

$$C \cdot \frac{d}{dt} T(t) = \frac{1}{R_1} \left[T_A - T(t) \right] + \frac{1}{R_2} \left[T_B - T(t) \right] \tag{1}$$

where C, R_1, R_2 are the interior thermal capacitance, interior/environment thermal resistance, and interior/boiler pipe thermal resistance parameters. This basic model captures the essence of home thermal dynamics, yet the parameters C, R_1, R_2 are unknown and can vary widely across homes. *Our goal is to identify these parameters from data. It is straight-forward to place sensors everywhere and collect data. You will learn to do something really innovative with this data!*

Problem 2: Parametric Modeling

(a). Reformulate (1) into the linear-in-the-parameters form $z(t) = \theta^T \phi(t)$. What signals play the role of $z(t)$ and $\phi(t)$? In this formulation, what are the elements of the parameter vector θ? Hints:

- $z(t) = \frac{d}{dt} T(t)$
- Parameter vector θ can be two-dimensional or three-dimensional. Think about why! Provide θ, ϕ for both the 2D and 3D cases.

If $\theta \in \mathbb{R}^3$, then is it possible to recover the individual three original parameters C, R_1, R_2? What if $\theta \in \mathbb{R}^2$?

(b). Next we use persistence of excitation (PE) to determine if the 2D or 3D version of θ is easier to identify. Recall that regressor signal is PE if

$$\lambda_{\min} \left\{ \int_0^t \phi(\tau) \phi^T(\tau) d\tau \right\} > 0 \tag{2}$$

where $\lambda_{\min}(\cdot)$ signifies the minimum eigenvalue of $(\cdot)$. The larger the left-hand side of (2) is, then the faster the estimate $\hat{\theta}(t)$ converges to the true value θ. Download the skeleton code `HW2_Skeleton.m` or HW2_Skeleton.ipynb, and data file HW2_Data.csv from bCourses, and consider the section entitled "Problem 2(b) - Persistence of Excitation". Enter the appropriate signals into the 2D and 3D versions of the phi signal. Run the remaining code in this section. Which parametric model is easier to identify, the 2D or 3D version?

Problem 3: Gradient Algorithm
In the following problems you will implement the *normalized* recursive gradient update law, given by

$$\frac{d}{dt}\hat{\theta}(t) = \Gamma\epsilon(t)\phi(t), \qquad \hat{\theta}(0) = \hat{\theta}_0, \tag{3}$$

$$\epsilon(t) = \frac{z(t) - \hat{\theta}^T\phi(t)}{m^2(t)}, \tag{4}$$

$$m^2(t) = 1 + \phi^T(t)\phi(t) \tag{5}$$

where $\Gamma = \Gamma^T > 0$ is a symmetric positive definite matrix, $\epsilon(t)$ is the *normalized* prediction error, and $m^2(t)$ is known as the normalization signal. This signal has the effect of better conditioning the algorithm. Note that everything is automatic in this algorithm, besides the selection of Γ and $\hat{\theta}_0$.

Download the skeleton code, ode_gradient.m from bCourses (for Matlab users only). This function encodes the algorithm (3)-(5). Complete the missing code in the sections entitled "Parametric model notation" and "Gradient update law". Include this code in your report.

Problem 4: Implementation
Consider the skeleton code HW2_Skeleton.m or HW2_Skeleton.ipynb, and data file HW2_Data.csv.

(a). Plot $T(t), T_A(t), T_B(t)$ versus 24 hours of time. Use axis labels with units, legends, linestyles, colors, and appropriate font sizes. Make the plots look professional! Provide this plot in your report.

(b). Complete the missing code in the section "Problem 4(b)". Initialize the ODE with values of $\hat{\theta}_0 = [0.1, 0.1]^T$. For Γ, it is recommended that you start small (e.g. $\Gamma = 10^{-3} \times I$, where I is the identity matrix), and then iteratively increase on a logarithmic scale. That is, try $\Gamma = 10^{-2} \times I$ next. Then perhaps $\Gamma = 10^{-1} \times I$. In your report, provide your completed code.

(c). Plot $\hat{\theta}_1(t), \hat{\theta}_2(t)$ versus time. What is the final value? Suppose the true values of parameters C, R_1, R_2 are $C = 10$, $R_1 = 2$, $R_2 = 0.75$. How do your estimates compare with the true values?

Problem 5: Model Validation
This final and important step validates your parameter estimates by testing them on data NOT used for identification.

(a). In the section entitled "Problem 5(a)," complete the code that creates a state-space object for your identified model. Use the "exit" estimates of the parameters from the last problem. That is, use the estimated parameter values at the final time-step. What are the A and B matrices for the LTI system? What is $u(t)$? Provide this section of completed code in your report.

(b). Simulate your identified model using this state-space model, with an initial condition of $\hat{T}(0) = 22°\text{C}$. Plot the indoor temperature $\hat{T}(t)$ predicted by your identified model, and the true indoor temperature $T(t)$ from the validation data set `HW2_ValData.csv`, available on bCourses. Use dashed and solid line styles to differentiate between the true data and model predictions. How do they compare? Be quantitative!

Problem 6: Least Squares Algorithm with Forgetting Factor
Now we consider the least squares algorithm, given by

$$\frac{d}{dt}\hat{\theta}(t) = P(t)\epsilon(t)\phi(t), \qquad \hat{\theta}(0) = \hat{\theta}_0, \tag{6}$$

$$\frac{d}{dt}P(t) = \beta P(t) - P(t)\frac{\phi(t)\phi^T(t)}{m^2(t)}P(t), \qquad P(0) = P_0 = Q_0^{-1}, \tag{7}$$

$$\epsilon(t) = \frac{z(t) - \hat{\theta}^T\phi(t)}{m^2(t)}, \qquad m^2(t) = 1 + \phi^T(t)\phi(t) \tag{8}$$

where β is called the "forgetting factor". This algorithm minimizes the *time-integrated* square error between the model and measurements, with one added twist. It exponentially discounts past data. That is, it applies an exponentially decaying factor on data measured backwards from the present. This inherently discounts old data relative to recent data. See Table 2 in the Appendix of CH2. Mathematically, it minimizes cost function

$$J(\hat{\theta}) = \frac{1}{2}\int_0^t e^{-\beta(t-\tau)}\frac{[z(\tau) - \hat{\theta}^T(t)\phi(\tau)]^2}{m^2(\tau)}d\tau + \frac{1}{2}e^{-\beta t}\left[\hat{\theta}(t) - \hat{\theta}_0\right]^T Q_0\left[\hat{\theta}(t) - \hat{\theta}_0\right] \tag{9}$$

Extend your gradient algorithm code to implement the least squares with forgetting factor equations (6)-(8). In your report, provide:

- Your least squares code, in a code box.
- Plot $\hat{\theta}_1(t), \hat{\theta}_2(t)$ versus time. What forgetting factor value β did you use?
- Plot of $\hat{T}(t)$ predicted from your model (using least squares) and the true $T(t)$, just like Problem 5(b). How do they compare? Be quantitative!

Congratulations! You have developed software to *learn* the dynamics of a home's thermal dynamics. Add a few more bells and whistles, some slick hardware, and maybe Google will buy your company for 3.2B USD!

Deliverables

Submit the following on bCourses. Be sure that the function files are named exactly as specified (including spelling and case), and make sure to zip all Matlab files.

`LASTNAME_FIRSTNAME_HW2.PDF`
`LASTNAME_FIRSTNAME_HW2.zip` which contains your respective Matlab or Python files.