Grey-box models (continued)
Models for the heat dynamics of a building

Summer school 2018 DTU - CITIES and NTNU - ZEN:

Time series analysis - with a focus on modelling and forecasting in energy systems

August 29, 2018

Installation

If you did not do this in advance, then you need to install the package `ctsmr`. See first the web page `ctsm.info` for OS specific instructions, some dependencies might be needed for the installation.

Introduction

The exercise is focused on grey-box modelling of the heat dynamics of a (small) building using stochastic differential equations (SDEs). In addition to the first exercise on greybox modelling, we will in this exercise test different techniques to:

1. Alter the noise level or system uncertainty to account for e.g. non-linear phenomena
2. Build a semi-parametric model to take into account that the solar penetration (i.e. relation between measured solar radiation and radiation entering into the building) as function of the position of the sun
3. Balance heat gains to the air temperature and the temperature of the thermal mass.

The data consists of several measurement from a small test box with a single window. In this exercise the following signals are used:

- T_{i} (`yTi` in data) the observed indoor temperatures. (°C)
- Q_{i} (`Qi` in data) the heat emitted by the electrical heaters in the test box (W)
- T_{e} (`Te` in data) the ambient temperature (°C)
- G_{v} (`Gv` in the data) the vertical south total solar radiation (W/m^2)
- G_{vn} (`Gvn` in data) the vertical north total solar radiation (W/m^2)

A full description of the data and the test setup can be found in the document `ST3 CE4 Instruction document.pdf`.

Questions

Question 1

Open the script `q1_system_noise_levels.R` . Remember to change the path with `setwd()` (in line 5) to set the working directory to the where the script file is located [1].

First we will look at the following two-state model from the the exercise *Grey-box models and model selection*. The RC-diagram for the deterministic part of the model is shown in Figure 1. The system equations are

$$dT_\mathrm{i} = \left(\frac{1}{R_\mathrm{ia}C_\mathrm{i}}(T_\mathrm{e} - T_\mathrm{i}) + \frac{1}{C_\mathrm{i}}A_\mathrm{w}\Phi_\mathrm{s} + \frac{1}{C_\mathrm{i}}\Phi_\mathrm{h}\right)dt + \sigma_\mathrm{i} d\omega_\mathrm{i} \tag{1}$$

$$dT_\mathrm{e} = \left(\frac{1}{R_\mathrm{ie}C_\mathrm{e}}(T_\mathrm{i} - T_\mathrm{e}) + \frac{1}{R_\mathrm{ea}C_\mathrm{e}}(T_\mathrm{a} - T_\mathrm{e})\right)dt + \sigma_\mathrm{e} d\omega_\mathrm{e} \tag{2}$$

and the measurement equation is

$$Y_k = T_{\mathrm{i},k} + \epsilon_k \tag{3}$$

where k counts the measurements from 1 to N and where the measurement error is assumed to be i.i.d. and follow a normal distribution with $\epsilon_k \sim N(0, \sigma_\epsilon^2)$.

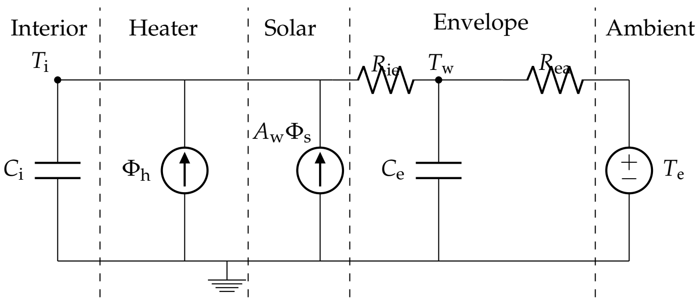

Figure 1 *RC-network of the most simple model extended with a state in the building envelope TiTe.*

Run the script line by line, stop right after plotting the data.

- The lower time series plot is of `stepQi`, which goes from 0 to 1. Try to change the argument `samples_after_Qi_step` above in the function preparing the data. How does it change `stepQi`?
- Now compare the two models implemented in `functions/sdeTiTw.R` and `functions/sdeTiTw_sigmalevels.R`. What is the difference?

Now go to the script and fit the two models. Compare the results:

[1] In RStudio menu "Session->Set Working Directory->To Source File Location" can be used

- What is plotted in the upper two plots? (You maybe have to look into the `analyzeFit()` function).
- What is indicated by the blue lines in the upper plot?
- Step back in the plots and compare the results, and look at the summary output. Which of the two models will you prefer and why?

So it becomes clear that we have some (possible non-linear) dynamics when the heating turns on and off, which our models doesn't predict so well. But instead of adding a more detailed description to the deterministic part of the model, we simply vary the system noise, or in other words, change the uncertainty level of our states under under different conditions. This is a very useful thing, since there will be many phenomena in buildings, especially occupied buildings, which will lead different to levels of noise, e.g., solar radiation and occupants doing funny things.

Finally, go through the "Nice features of R and Rstudio" to learn some nice tricks for modelling in R and RStudio.

Question 2

So far we have assumed that the solar gain is proportional to the radiation outside the test box. In reality, the heat gain from the sun depends highly on building geometry, surroundings, window properties, etc. In this part of the exercise, we will apply splines to estimate the solar heat gain as a function of solar position.

First we make a hidden state for `Aw` (also called the gA-value) to investigate if it changes over time and as the function of the sun position.

Open the script `q2_splined_ga_value.R` and run it line by line. Stop after you have plotted the state of `Aw` as a function of time, and the state of `Aw` as a function of the sun azimuth.

It is clear that the state of `Aw` not is constant, but change. Furthermore, it seems like there could be a relation to the sun azimuth.

Now, answer the questions below as you progress in modelling the solar radiation with use of splines.

- The spline function we want to estimate is the gA value (e.g. the percentage of solar heat that enters through the window, multiplied with the window area) as a function of the sun azimuth. First, plot the sun elevation as a function of the sun azimuth, as well as a horizontal line through 0 (notice that the angles is in radians). Find the azimuth angles (in radians) that corresponds to the sunrise and sunset, and assign them to `azumith_bound <- c(...   , ...)` below. These two angles will in a moment be our boundary azimuth angles. Outside the boundaries the gA value is 0, as the sun is below the horizon and the radiation is zero. Thus, we are only interested in the gA values from sunrise to sunset.

- Define the base splines in the following lines of the script and stop after you have assigned the base splines to the data frame with the command `X <- cbind(X,Xbs)`. Now play around with the four parameters in in the vector `Aw`, and plot the resulting spline function to get an understanding of how the base splines and the resulting spline function work. What happens if `Aw` only consists of 1's?
  (Pro tip: the package `lubridate` is very useful when working with dates and time. Which often is the case for when dealing with time series!)

- Fit the model and investigate the estimated parameters. Is the parameters `Aw1`, `Aw2`, `Aw3` and `Aw4` significant, and is the magnitude reasonable when the actual glazed area is 52 x 52 cm?

- Plot the gA curve and the 95 % confidence interval. The window in the test box is facing south towards an open area, and should therefore be rather unobstructed. Why do the gA curve then have a shape which is asymmetrical around the south (180 degrees)?

- (Optional) If you think you know why the gA curve is asymmetrical, try to expand or modify the model to take that into account.

You can look into the report `CE4_Denmark_DTU_v2.pdf` Chapter 6, where the same things with the gA-value is carried out for a discrete model (an ARX).

(Optional) Question 3

Now, there are other ways to improve the TiTw model that we started with. Open `q3_balanced_heat_gains`:

- Estimate the parameters in the model `fitTiTw_X` and the following model, `fitTiTw_GinTw_X`. Which of the models has the highest (log) likelihood?

- The second model has included a parameter, `p`, which is used to balance how much of the solar radiation that should be assigned to each of the two states, namely `Ti` and `Tw`. Open the script for the function `sdeTiTw` and `sdeTiTw_GinTw` and compare them. How is the balancing actually done, and for which value of `p` does the model `sdeTiTw_GinTw` correspond to the first model `sdeTiTw`?

- What is the estimate of the parameter, `p`, and is it significantly different from 0?

- Explain in words what the meaning of a small and a large value of `p` means. Based on the estimated parameter, `p`, is it reasonable to assume that the solar radiation entering the building solely should be assigned directly to the air temperature?

- With the model `sdeTiTw_sigmalevels` as starting point, setup a model that includes two layers in the wall and a balancing parameter, `p`, to balance the solar radiation between `Ti` and `Tw`.

As it is the case for the solar radiation, the heat input from the heating system can also be assigned to different thermal capacities. Until now, we have assigned it directly to the indoor air temperature, `Ti`. To which state the heat should be assigned depends on the reaction time of the heating system. E.g. an electrical heat blower has a much faster responds time than a built-in floor heating system, and should most likely not be assigned state with very slow heat dynamics.

- Open the script of the function `sdeTiTw2_QinTw` and see how the model is made. Compared to the previous model, we have introduced an additional layer in the wall and assigned the heat input from the heating system to the inner wall. Fit the model `fitTiTw2_QinTw_X` and assess—only from the loglikelihood—if it has improved compared to the previous model.

- Eyeball the residual plots (ACF, cummulated periodogram, and the residuals as function of time). Why does it not seem reasonable to conclude that the model has improved?

- What does it mean in physical terms when we include an additional state for the heating system, `Th`, as done in the model `sdeTiThTw2`?

- Look closely at the plots for the fit `fitTiThTw2_X`. What seems to drive the large fluctuations in the residuals, and what can the reason be?

Questions and comments: `pbac@dtu.dk` or `chrras@dtu.dk`