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

Summer school 2019 DTU - CITIES and NTNU - ZEN:

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

August 27, 2019

Installation

If you did not install the package `ctsmr` in advance, then you need to do it now. See first the web page `ctsm.info` for OS specific instructions.

Introduction

The exercise is about 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 use different techniques to:

1. Alter the noise level (or system uncertainty) over time to take into account changing uncertainties.
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 external (or 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`.

Q1: Noise level

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 (in RStudio menu "Session->Set Working Directory->To Source File Location" can be used).

First we will work with the 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

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

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

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 $\epsilon_k \sim N(0, \sigma_\epsilon^2)$.

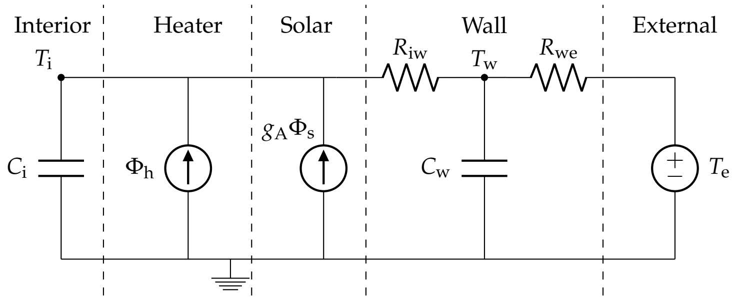

Figure 1 *RC-network of the most simple model extended with a state in the building envelope TiTw.*

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:

- 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 last part of the script under "Nice features of R and Rstudio" to learn some nice tricks for modelling in R and RStudio.

Q2: gA-curve modelled with spline

The "solar gain" is defined as the ratio of: heat absorbed inside the building to the measured solar power. So far we have assumed that the solar gain is simple always proportional to the radiation measured outside the test box. In reality, the solar gain depends highly on building geometry, surroundings, window properties, etc. In this part of the exercise, we will apply base splines to estimate the solar gain as a non-parametric function of the sun position.

The solar gain is more specifically referred to as the gA value (g is the percentage of solar heat that enters through the window, multiplied with the window area A).

We will start out using a trick to learn how the solar gain behaves. We make a hidden state for `gA`, such that we can investigate if it changes over time and further if it changes as the function of other variables - in particular the sun position.

The state equation which is added to the model is

$$dg_{\mathrm{A}} = \sigma_{\mathrm{a}} d\omega_{\mathrm{a}} \tag{4}$$

hence g_{A} is now not a parameter, but a state – which is modelled as a random walk and will be estimated when fitting the model.

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

It is clear that the state of `gA` 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 the script and apply base splines to model the gA curve:

- The spline function we want to estimate is the gA value 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 be used as the boundary knots. 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 curve from sunrise to sunset.

- Generate the base splines and stop after you have merged the base splines to the dataframe with the command `X <- cbind(X,Xbs)`. Now play around with the four parameters in in the vector `gA` and plot the resulting spline function. Try different values to get an understanding of how the base splines and the resulting spline function behave. What happens if `gA` only consists of 1's? (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. Are the parameters `gA1`, `gA2`, `gA3` and `gA4` significant different from zero? and is the magnitude reasonable when the actual glazed area is 52 x 52 cm?

- Plot the estimated 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. If the estimated gA curve have a shape which is asymmetrical around south (180 degrees), what could be an explanaitionS?

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

You can have a 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 model).

(Optional) Q3: Balancing heat gains

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 in the 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 control the proportions of the solar radiation entering the states `Ti` and `Tw`. Open the script for the function `sdeTiTw` and `sdeTiTw_GinTw` and compare them. How is the solar gain modelled, and for which value of `p` does the model `sdeTiTw_GinTw` correspond to in 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 parameter, `p`, to divide 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 enter into different states. Until now, we have assigned it directly to the indoor air temperature, `Ti`. To which state the heat should be assigned depends on the response time of the heating system. E.g. an electrical heat blower has a much faster response time than a built-in floor heating system, and should most likely not be assigned a state with very slow heat dynamics.

- Open the script of the function `sdeTiTw2_QinTw` and check how the model is defined. 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 log-likelihood – 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`