Introduction to ctsm-r

(Based on slides created by Rune Juhl)

Peder Bacher

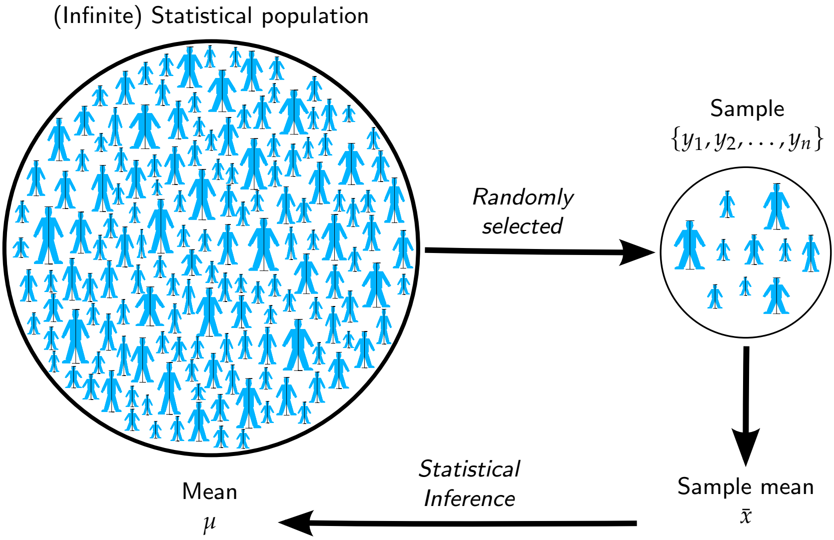
DTU Compute, Dynamical Systems
Building 303B, Room 010
DTU - Technical University of Denmark
2800 Lyngby – Denmark
e-mail: pbac@dtu.dk

Summer school 2021:

Time Series Analysis - with a focus on Modelling and Forecasting in Energy Systems

Overview

Population and sample



Parameter estimation with example

Simplest example: a constant model for the mean

- Model

$$Y_i = \mu + \varepsilon_i \quad \text{, where } \varepsilon_i \sim N(0, \sigma^2) \text{ and i.i.d.}$$

- i.i.d.: identically and independent distributed
- The parameters are: the mean μ and the standard error σ
- We take a sample of $n = 10$ observations

$$(y_1, y_2, \ldots, y_{10})$$

Likelihood

The likelihood is defined by the joint probability of the data

$$L(\mu,\sigma) \equiv p(y_1, y_2, \ldots, y_{10}|\mu,\sigma)$$

Hence, it's a function of the two parameters (the sample is observed, so it is not varying).

Due to independence

$$= \prod_{i=1}^{10} p(y_i|\mu,\sigma)$$

We assume in our model that the error $\varepsilon_i = Y_i - \mu$ is normal distributed (Gaussian), so

$$p(y_i|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i-\mu)^2}{2\sigma^2}\right) \qquad (1)$$

Maximum likelihood estimation

Parameter estimation

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \left(-\ln(L(\theta))\right)$$

where $\theta = (\mu, \sigma)$

Maximum likelihood estimation

Parameter estimation

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \left(-\ln(L(\theta)\right)$$

where $\theta = (\mu, \sigma)$

Run the example in R

Likelihood for time correlated data

Given a sequence of measurements $\mathcal{Y}_N$

$$L(\theta) = p(\mathcal{Y}_N|\theta) = p(y_N, y_{N-1}, \ldots, y_0|\theta)$$
$$= \left(\prod_{k=1}^{N} p(y_k|\mathcal{Y}_{k-1}, \theta)\right) p(y_0|\theta)$$

Parameter estimation

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \left(-\ln(L(\theta))\right)$$

Likelihood for time correlated data

If Gaussian

$$\hat{y}_{k|k-1} = E[y_k|\mathcal{Y}_{k-1}, \theta]$$
$$R_{k|k-1} = V[y_k|\mathcal{Y}_{k-1}, \theta]$$
$$\varepsilon_k = y_k - \hat{y}_{k|k-1}$$

then the likelihood is

$$L(\theta) = \left(\prod_{k=1}^{N} \frac{\exp(-\frac{1}{2}\varepsilon_k^T R_{k|k-1}^{-1} \varepsilon_k)}{\sqrt{|R_{k|k-1}|}\sqrt{2\pi}^l} \right)$$

Maximised using quasi Newton

Kalman filter

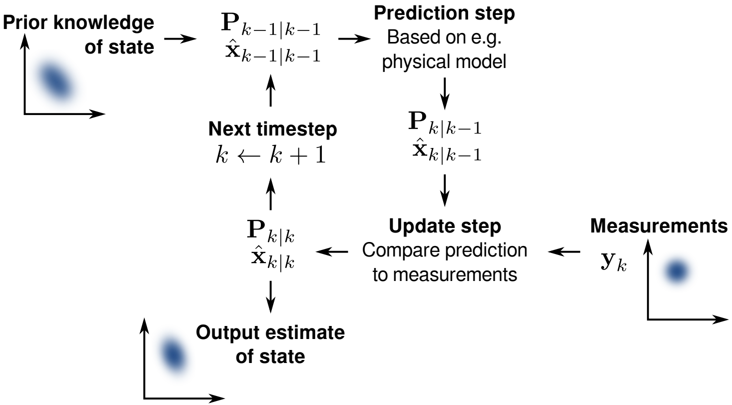


Figure: "Basic concept of Kalman filtering" by Petteri Aimonen. Wikipedia

Introduction to grey-box modelling and **ctsmr**

Grey-box modelling

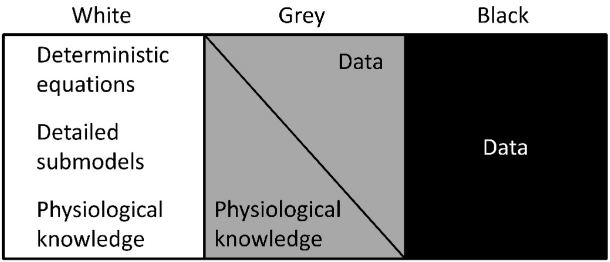


Figure: Ak et al. 2012

Grey-box modelling

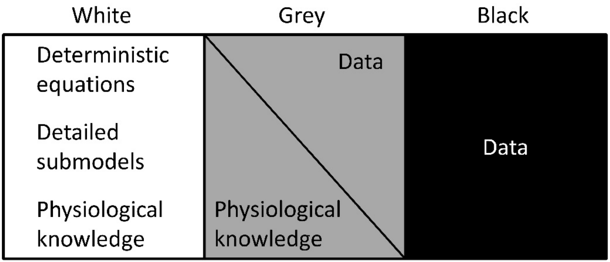


Figure: Ak et al. 2012

Bridges the gap between physical and statistical modelling.
THERE is a manual on `ctsm.info`

ctsmr

Continuous Time Stochastic Modelling in R

ctsmr

Continuous Time Stochastic Modelling in R

more correctly

Continuous-Discrete Time Stochastic Modelling in R

The model class

ctsmr implements a state space model with:

Continuous time stochastic differential system equations (SDE)

$$dX_t = f(X_t, U_t, t, \theta)dt + g(X_t, U_t, t, \theta)dB_t$$

Discrete time measurement equations

$$Y_{t_n} = h(X_{t_n}) + e_{t_n} \qquad e_{t_n} \in N(0, S(u_n, t_n, \theta))$$

- Underlying physics (system, states) modelled using continuous SDEs.
- Some (or all) states are observed in discrete time.

Features in CTSM-R

- Automatic classification (LTI or NL)
- Symbolic differentiation replaced AD (NL only)
  (Jacobians are computed faster.)
- Finite difference approximation of gradients are computed in parallel.
- Scriptable! Run multiple model during the night. Possible to use compute cluster.
- Direct access to plotting facilities from the R framework.

Loading the library

The R package is called **ctsmr**

R code

```
library(ctsmr)
```

Loading the library

The R package is called **ctsmr**

R code

```
library(ctsmr)
```

The model class is called **ctsm** - **C**ontinuous **T**ime **S**tochastic **M**odel.

R code

```
MyModel <- ctsm$new()
```

Class.. huh?

- **ctsm** is a ReferenceClass.
- The functions are methods attached to the class.

Class.. huh?

- **ctsm** is a ReferenceClass.
- The functions are methods attached to the class.

ctsm methods

Specifying the model:

- $addSystem()
- $addObs()
- $setVariance()
- $addInput()

Estimate and prediction:

- $setParameter()
- $setOptions()
- $estimate()

ctsmr defined functions

- predict
- simulate
- filter.ctsmr
- smooth.ctsmr

How to add System Equations

Use the $addSystem method to add a stochastic differential equation as a system equation.

R code

```
MyModel$addSystem( dX ~ (mu*X-F*X/V)*dt + sig11*dw1)
MyModel$addSystem( dS ~ (-mu*X/Y+F*(SF-S)/V) * dt + sig22*dw2)
MyModel$addSystem( dV ~ F*dt + sig33*dw3 )
```

Pay attention to the $\sim$. Do not use $=$.
The diffusion processes must be named dw{n}

How to add Observation Equations

Use the $addObs method to add a measurement/observation equation.

$$\boldsymbol{Y} = \begin{bmatrix} Y1 \\ Y2 \\ Y3 \end{bmatrix} = \begin{bmatrix} X \\ S \\ V \end{bmatrix}$$

R code

```
MyModel$addObs(y1 ~ X)
MyModel$addObs(y2 ~ S)
MyModel$addObs(y3 ~ V)
```

Pay attention to the $\sim$. Do not use $=$.

How to set the Variance structure of the Measurement Equations

The Example

Use the `$setVariance` method.

Example

```
MyModel$setVariance(y1y1 ~ s11)
```

How to set the Variance structure of the Measurement Equations

The Example

Use the `$setVariance` method.

Example

```
MyModel$setVariance(y1y1 ~ s11)
```

For `y1,,y2,y3` the size of the variance-covariance matrix is $3x3$.

$$S = \begin{bmatrix} \text{s11} & & 0 \\ 0 & \text{s22} & \\ & & \text{s33} \end{bmatrix}$$

R code

```
MyModel$setVariance(y1y1 ~ s11)
MyModel$setVariance(y2 ~ s22)
MyModel$setVariance(y3^2 ~ s33)
```

Pay attention to the $\sim$. Do not use $=$.

Which variables are inputs?

Use the `$addInput` method to specify which variable is an input and not a parameter.

R code

```
MyModel$addInput(F)
```

How to specify initial values, boundaries and prior standard deviance (for MAP)?

Use the `$setParameter` method.

R code

```
MyModel$setParameter(X = c(init=1,lb=0,ub=2),
                     S0 = c(0.25,0,1))
MyModel$setParameter(V0 = c(1,lower=0,upperbound=2))
```

Pay attention to the $=$. Do not use $\sim$.

- Quite flexible.
- Named numbers (e.g. `init=3`) are processed first.
- Initial state values (e.g. X_0) can be named `X0` or `X`.
- `MyModel$ParameterValues` contains the parsed values.

How to change filtering and numerical optimisation options (advanced)?

Use the `$setOptions` method to change the options found in `MyModel$options`.

Specify the data

`ctsm` expects a data.frame containing time and all inputs and outputs.

Example

```
MyData <- data.frame(t = c(1,2,3), F = c(4,3,2), Y1 = c(7,6,5), Y2 =
...)
```

Multiple independent datasets can be given as a list of data.frames.

Example

```
AllMyData <- list(MyData1, MyData2, MyData2, ...)
```

Estimate the parameters

To estimate the parameters run:

```
fit <- MyModel$estimate(data = MyData)
```

Parameter inference

Like `lm()` use `summary()` on the fit for additional information.

- Parameter estimates alone:

  `fit`

- + standard deviance, t-statistics and p-values:

  `summary(fit)`

- + correlation of parameter estimates:

  `summary(fit, correlation=TRUE)`

- + additional information $\left(\frac{dF}{d\theta}, \frac{dPen}{d\theta}\right)$:

  `summary(fit, extended=TRUE)`

How to get k-step predictions

Use the `predict` function.

Usage

```
one.step.prediction <- predict(fit)
```

Available options:

- `n.ahead` number of steps ahead to predict.
- `newdata` to predict using a new dataset.

Diagnostics

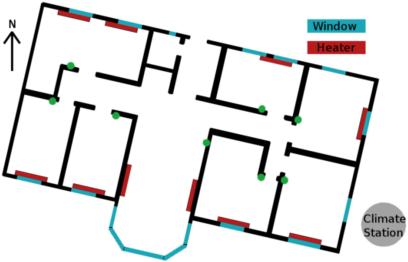
- k-step predictions `predict`
- filtered states `filter.ctsmr`
- smoothed states `smooth.ctsmr`
- simulations `simulate`
- likelihood ratio tests

Example: Selecting **a suitable grey-box model** for the heat dynamics of a building

Test case: One floored 120 m^2 building

Objective

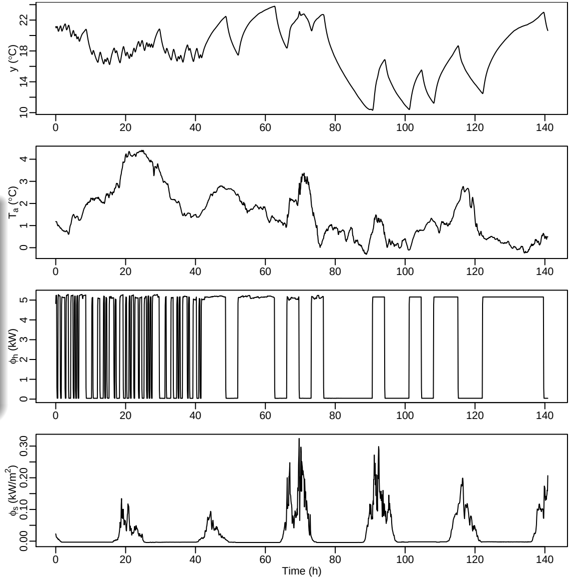
Find the best model describing the heat dynamics of this building



Data

Measurements of:

- y_t Indoor air temperature
- T_a Ambient temperature
- Φ_h Heat input
- Φ_s Global irradiance



Two big challenges when modelling with data

- **Model selection:** How to decide which model is most appropriate to use?
  We are looking for a model which gives us un-biased estimates of physical parameters of the system. This requires that the applied model is neither too simple nor too complex
- **Model validation:** How to validate the performance of a dynamical model?
  We need to asses if the applied model fulfill assumptions of white-noise errors, i.e. that the errors show no lag-dependence

Model selection

Likelihood ratio test: Test for model expansion

Say we have a model and like to find out if an expanded version will give a significantly better description of data

i.e. give an answer to: Should we use the expanded model instead of the one we have?

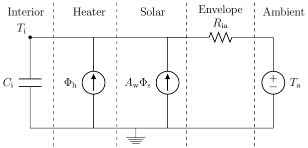
The likelihood ratio test

$$\lambda(\boldsymbol{y}) = \frac{L_{\text{sub}}(\hat{\boldsymbol{\theta}}_{\mathbf{mle,sub}})}{L(\hat{\boldsymbol{\theta}}_{\mathbf{mle}})}$$

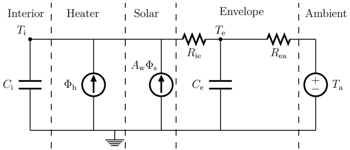
can be applied to test for significant improvement of the expanded model (with maximum likelihood $L_{\text{sub}}(\hat{\boldsymbol{\theta}}_{\mathbf{mle,sub}})$) over the sub-model (with maximum likelihood $L(\hat{\boldsymbol{\theta}}_{\mathbf{mle}})$)

Test for expansion

Simplest model

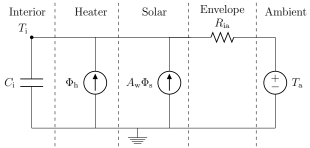


First extension: building envelope part (*TiTe*)

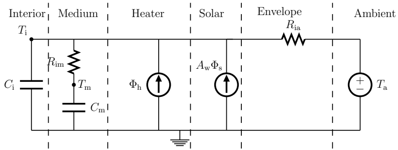


Test for expansion

Simplest model

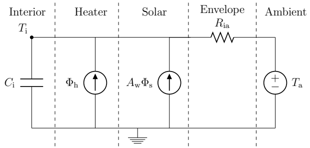


First extension: indoor medium part (*TiTm*)

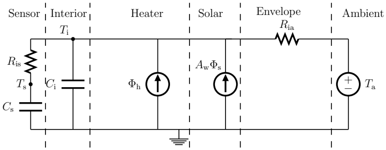


Test for expansion

Simplest model

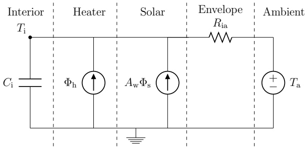


First extension: sensor part (*TiTs*)

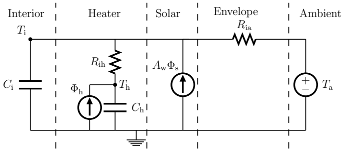


Test for expansion

Simplest model

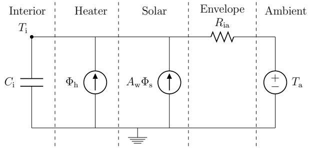


First extension: heater part (*TiTh*)



Test for expansion

Simplest model



First extension: Which one??

TiTe, *TiTm*, *TiTs*, or *TiTh* ?

Log-likelihoods

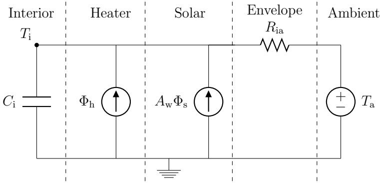
Simplest	*Ti*			
$l(\theta; \mathcal{Y}_N)$	**2482.6**			
m	6			
Expanded	*TiTe*	*TiTm*	*TiTs*	*TiTh*
$l(\theta; \mathcal{Y}_N)$	3628.0	3639.4	3884.4	**3911.1**
m	10	10	10	10

Likelihood-ratio test

Sub-model	**Model**	$m - r$	p-value
Ti	*TiTh*	4	$< 10^{-16}$

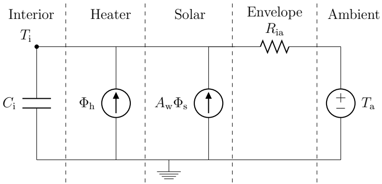
Identify the best physical model for the data

Simplest model

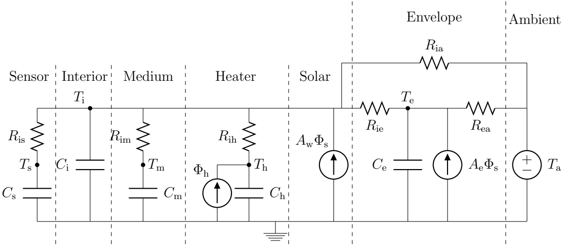


Identify the best physical model for the data

Simplest model

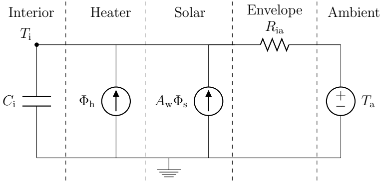


Most complex model applied



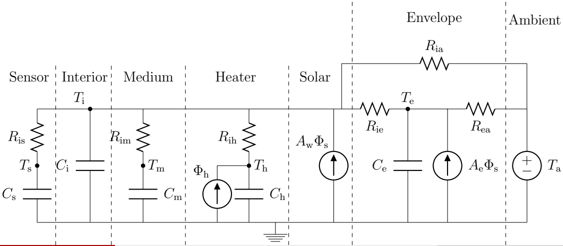
Identify the best physical model for the data

Simplest model



The best model for the given data is probably in between

Most complex model applied



Iteration	Models			
Start	*Ti*			
$l(\theta; \mathcal{Y}_N)$	**2482.6**			
m	6			
1	*TiTe*	*TiTm*	*TiTs*	*TiTh*
	3628.0	3639.4	3884.4	**3911.1**
	10	10	10	10
2	*TiThTs*	*TiTmTh*	*TiTeTh*	
	4017.0	5513.1	**5517.1**	
	14	14	14	
3	*TiTeThRia*	*TiTeThAe*	*TiTmTeTh*	*TiTeThTs*
	5517.3	5520.5	5534.5	**5612.4**
	15	15	18	18
4	*TiTeThTsRia*	*TiTmTeThTs*	*TiTeThTsAe*	
	5612.5	5612.9	**5614.6**	
	19	22	19	
5	*TiTmTeThTsAe*	*TiTeThTsAeRia*		
	5614.6	**5614.7**		
	23	20		

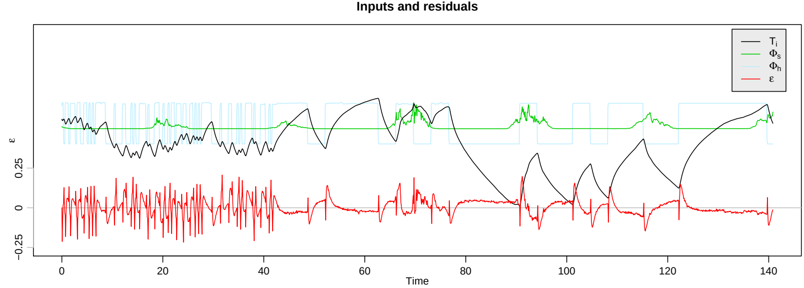
Iteration	Sub-model	Model	$m-r$	$-2log(\lambda(y))$	p-value
1	*Ti*	*TiTh*	4	4121	$< 10^{-16}$
2	*TiTh*	*TiTeTh*	4	4634	$< 10^{-16}$
3	*TiTeTh*	*TiTeThTs*	4	274	$< 10^{-16}$
4	*TiTeThTs*	*TiTeThTsAe*	1	6.4	0.011
5	*TiTeThTsAe*	*TiTeThTsAeRia*	1	0.17	0.68

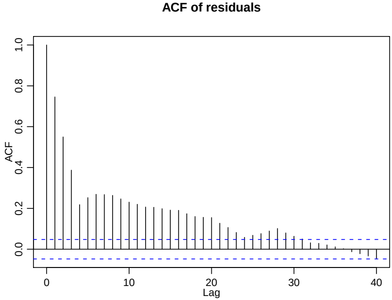
Model validation

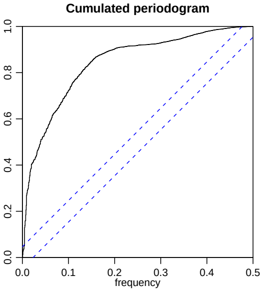
How can the performance of a dynamical model be evaluated?

- We assume that the residuals are i.i.d and normal
- Auto-Correlation Function (ACF) and Cumulated Periodogram (CP) of the errors are the basic tools
- Time series plots of the inputs, outputs, and the errors are valuable for pointing out model deficiencies

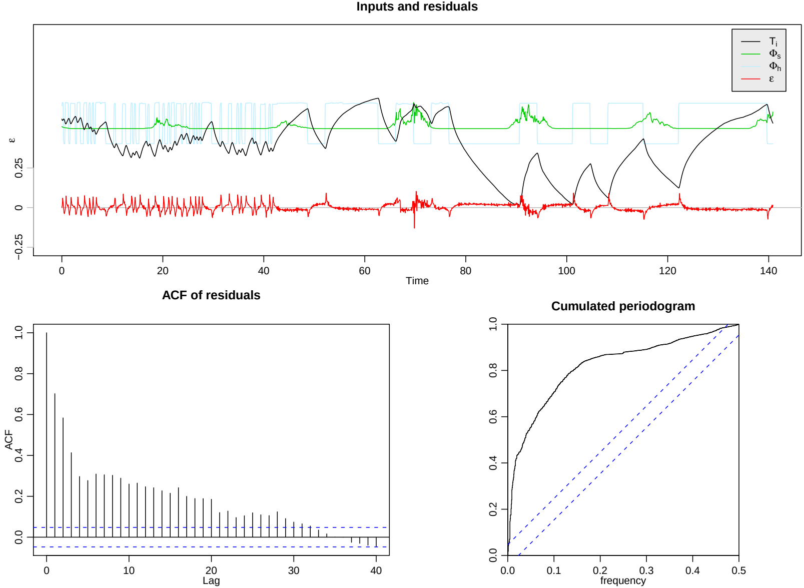
Evaluate the simplest model



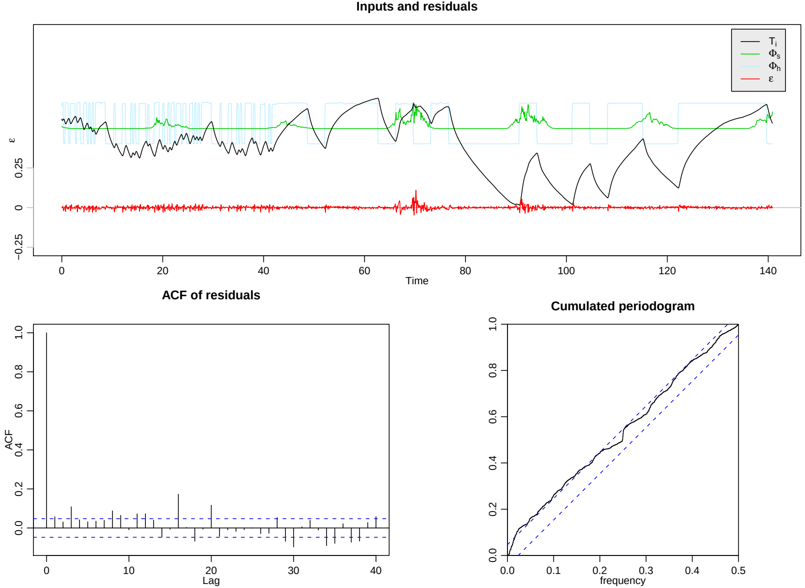




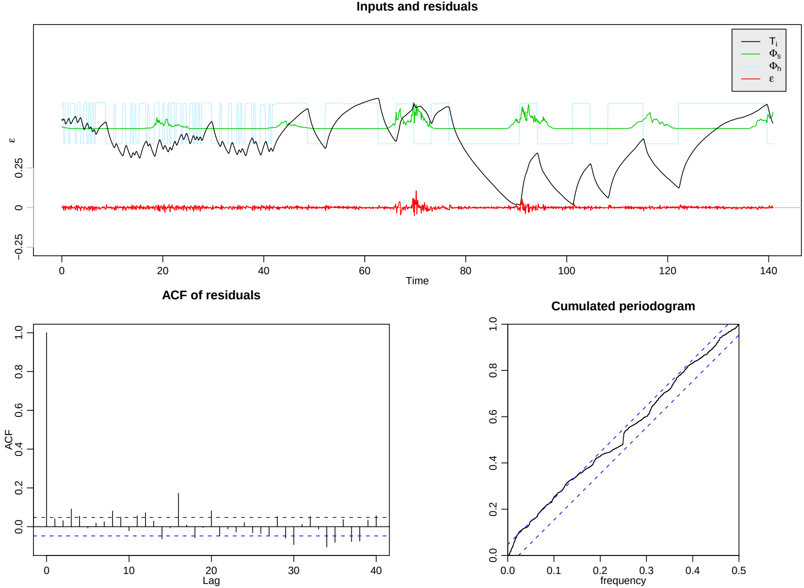
Evaluate the model selected in step one



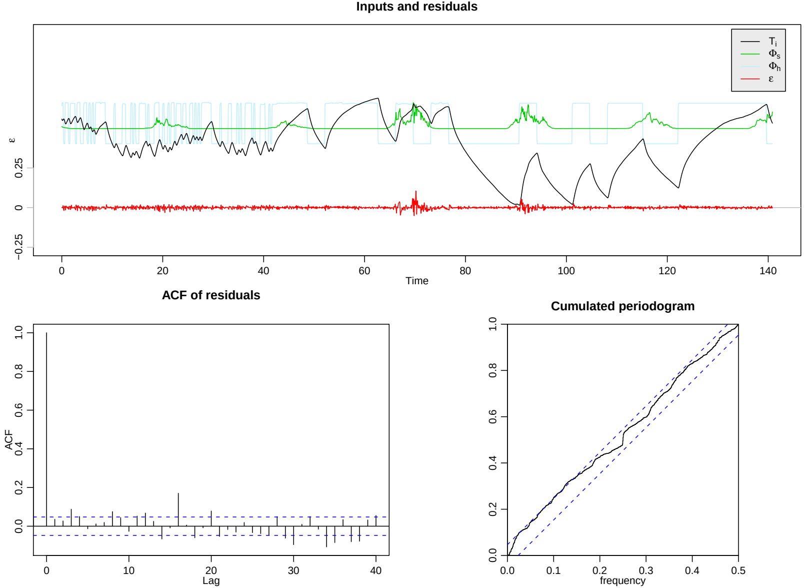
Evaluate the model selected in step two



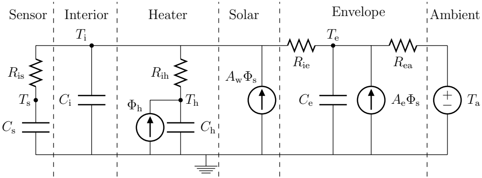
Evaluate the model selected in step three



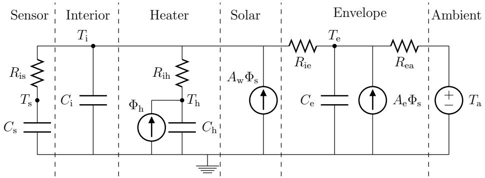
Evaluate the selected model in step four



Selected model



Selected model



Estimated parameters

$\hat{C}_i$	0.0928	(kWh/°C)
$\hat{C}_e$	3.32	-
$\hat{C}_h$	0.889	-
$\hat{C}_s$	0.0549	-
$\hat{R}_{ie}$	0.897	(^{c}ircC/kW)
$\hat{R}_{ea}$	4.38	
$\hat{R}_{ih}$	0.146	-
$\hat{R}_{is}$	1.89	-
$\hat{A}_w$	5.75	(m^2)
$\hat{A}_e$	3.87	-

Estimated time constants

$\hat{\tau}_1$	0.0102	hours
$\hat{\tau}_2$	0.105	-
$\hat{\tau}_3$	0.788	-
$\hat{\tau}_4$	19.3	-

Conclusions

- Applied Grey-box modelling, where a combination of *prior physical knowledge* and *data-driven modelling* is utilized
- Using a forward selection procedure with likelihood-ratio tests a suitable physical model is found
- The ability of the selected models to describe the heat dynamics are evaluated with the ACF, CP, and time series plots

Identifiability

Identifiability

Model identifiability is important for estimation in general (less important for prediction, very important for parameter interpretation).

There are two aspects of identifiability:

- **Structural identifiability:** the parameters in the model can never be estimated due to the structure of the model. Depends only on the model.
- **Practical identifiability:** there is not enough information in the data available to estimate the parameters in the model. Depends both on the model and the data.

Structural identifiability

State space model (innovation form)

$$\frac{d\hat{X}(t)}{dt} = A\hat{X}(t) + BU(t) + K\epsilon(t)$$
$$Y(t) = C\hat{X}(t) + DU(t) + \epsilon(t)$$

Apply the bilateral Laplace transformation (and after some voodoo)

$$\begin{aligned} Y(s) &= C(sI - A)^{-1}BU(s) + C(sI - A)^{-1}K\epsilon(s) + DU(s) + \epsilon(s) \\ &= \left(C(sI - A)^{-1}B + D\right)U(s) + \left(C(sI - A)^{-1}K + I\right)\epsilon(s) \end{aligned}$$

Focus on the input related transfer function

$$H_i(s) = C(sI - A)^{-1}B + D \tag{2}$$

Analyse the identifiability of an SDE model of a Wall

A lumped RC model of the wall

$$dT_w = \frac{1}{C_w}\left(\frac{T_a - T_w}{R_{aw}} + \frac{T_i - T_w}{R_{wi}}\right)dt + d\omega_1(t)$$

$$dT_i = \frac{1}{C_i}\left(\frac{T_w - T_i}{R_{wi}}\right)dt + d\omega_2(t)$$

$$y_{t_k} = Ti_{t_k} + \sigma_{t_k}$$

Transfer function

Apply equation **??** to obtain the input transfer function

$$H_{input}(s) = \frac{\frac{1}{C_i\, C_w\, R_{aw}\, R_{wi}}}{s^2 + \frac{R_{aw}\, C_i + C_i\, R_{wi} + R_{aw}\, C_w}{C_i\, C_w\, R_{aw}\, R_{wi}} \cdot s + \frac{1}{C_i\, C_w\, R_{aw}\, R_{wi}}}$$

Transfer function

Apply equation **??** to obtain the input transfer function

$$H_{input}(s) = \frac{\frac{1}{C_i\,C_w\,R_{aw}\,R_{wi}}}{s^2 + \frac{R_{aw}\,C_i + C_i\,R_{wi} + R_{aw}\,C_w}{C_i\,C_w\,R_{aw}\,R_{wi}} \cdot s + \frac{1}{C_i\,C_w\,R_{aw}\,R_{wi}}}$$

And compare it to

$$H(s) = \frac{b_0}{s^2 + a1 \cdot s + a0}$$

Transfer function

Apply equation **??** to obtain the input transfer function

$$H_{input}(s) = \frac{\frac{1}{C_i\,C_w\,R_{aw}\,R_{wi}}}{s^2 + \frac{R_{aw}\,C_i + C_i\,R_{wi} + R_{aw}\,C_w}{C_i\,C_w\,R_{aw}\,R_{wi}} \cdot s + \frac{1}{C_i\,C_w\,R_{aw}\,R_{wi}}}$$

And compare it to

$$H(s) = \frac{b_0}{s^2 + a1 \cdot s + a0}$$

Only two independent equations

$$a_0 = \frac{1}{C_i\,C_w\,R_{aw}\,R_{wi}}$$
$$a_1 = \frac{R_{aw}\,C_i + C_i\,R_{wi} + R_{aw}\,C_w}{C_i\,C_w\,R_{aw}\,R_{wi}}$$

Fit all four parameters?

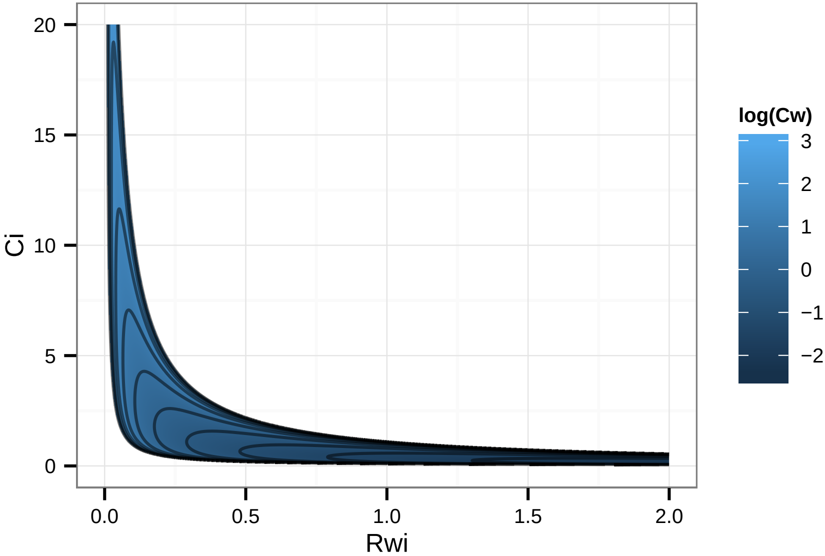
Solve two equations for four parameters.

$$
\begin{aligned}
Ci &= Ci \\
Rwi &= Rwi \\
Cw &= -\frac{C_i}{{C_i}^2 {R_{wi}}^2 a_0 - a_1\, C_i\, R_{wi} + 1} \\
R_{aw} &= -\frac{{C_i}^2 {R_{wi}}^2 a_0 - a_1\, C_i\, R_{wi} + 1}{{C_i}^2 R_{wi}\, a_0}
\end{aligned}
$$

Note: a_0 and a_1 are known when simulating data.

C_w is a function of other parameters

Below is the feasible C_w parameters: $C_w > 0$

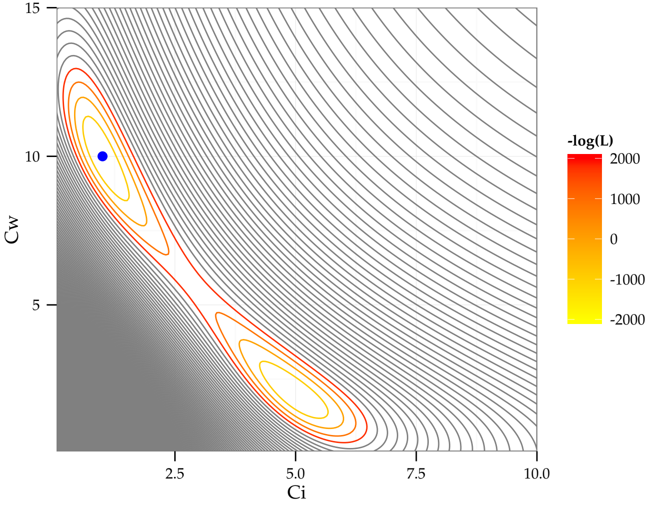


Estimate two parameters

We can estimate two.. So try fixing R_{wi} and R_{aw}

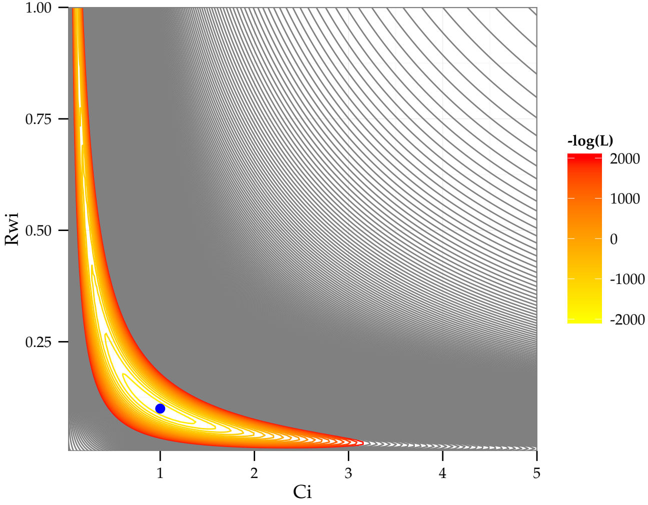
Estimate two parameters

We can estimate two.. So try fixing R_{wi} and R_{aw}



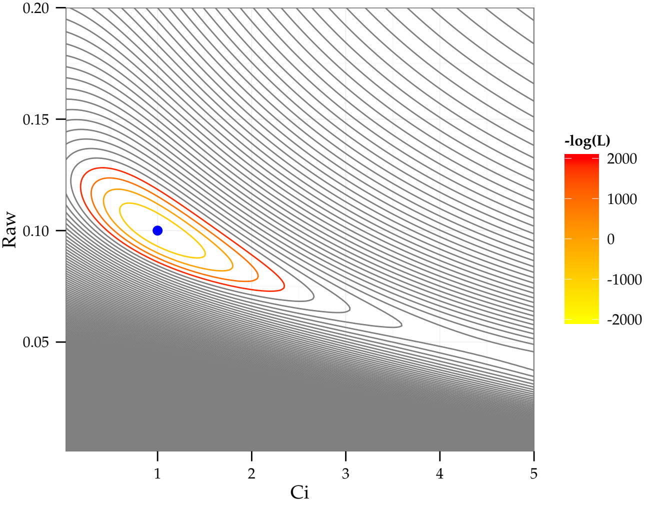
Estimate two parameters

We can estimate two.. So try fixing C_w and R_{aw}



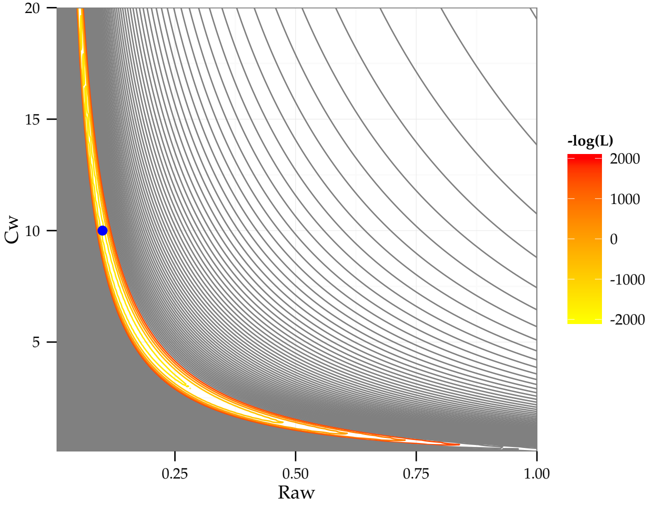
Estimate two parameters

We can estimate two.. So try fixing C_w and R_{wi}



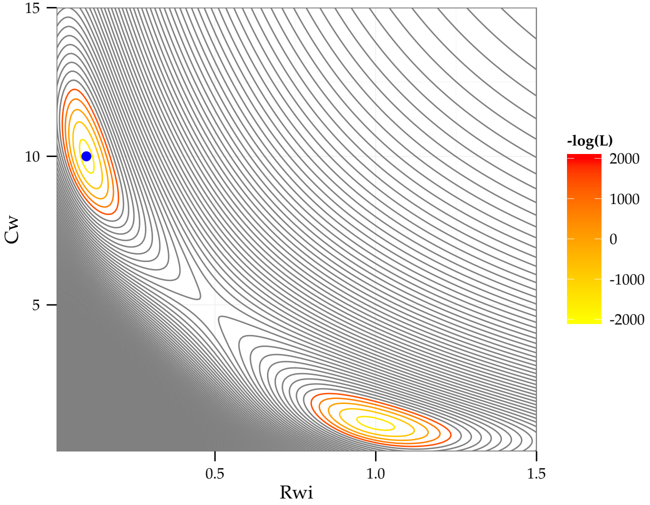
Estimate two parameters

We can estimate two.. So try fixing R_{wi} and C_i



Estimate two parameters

We can estimate two.. So try fixing R_{aw} and C_i



Estimate two parameters

We can estimate two.. So try fixing C_i and C_w