Simple example of grey-box modeling of the heat dynamics of a wall with CTSM-R

Rune Juhl Niels Rode Kristensen Peder Bacher Jan Kloppenborg
Henrik Madsen

August 3, 2022

Chapter 1

Introduction

This document contains a simple example of using CTSM-R for modeling of a wall. Imagine that we want to estimate the parameters in a simple lumped model of the heat dynamics of a wall. The model is lumped into two parts: one representing the outer wall The system equations of the model is

$$dT_1 = \left(\frac{1}{C_1 R_2}(T_2 - T_1) + \frac{1}{C_1 R_1}(T_e - T_1)\right)dt + \sigma_1 d\omega_1 \tag{1.1}$$

$$dT_2 = \left(\frac{1}{C_2 R_3}(T_i - T_2) + \frac{1}{C_2 R_2}(T_1 - T_2)\right)dt + \sigma_2 d\omega_2 \tag{1.2}$$

where

- The state variables T_1 and T_2 represent the outer wall temperature and the inner wall temperature, respectively
- The inputs T_e and T_i are the measured outdoor temperature and the indoor temperature
- The parameters C_1 and C_2 represent the heat capacity of the outer wall and the inner wall
- R_1, R_2 and R_3 represent the thermal resistance between the

The RC-diagram in Figure 1.1 is an illustration of the model. The measurement equation is:

$$q_{t_k} = \frac{1}{R_3}(T_\mathrm{i} - T_2) + \sigma e_{t_k} \tag{1.3}$$

where the output variable q is the heat flux and $e_k \in N(0, \sigma)$. The true parameter values used for generating the data we will use are given in Table 1.1.

C_1	C_2	R_1	R_2	R_3	σ_{11}	σ_{22}	σ
100	50	1	2	0.5	0	0	0.01

Table 1.1: True values of the parameters of the wall heat dynamics model.

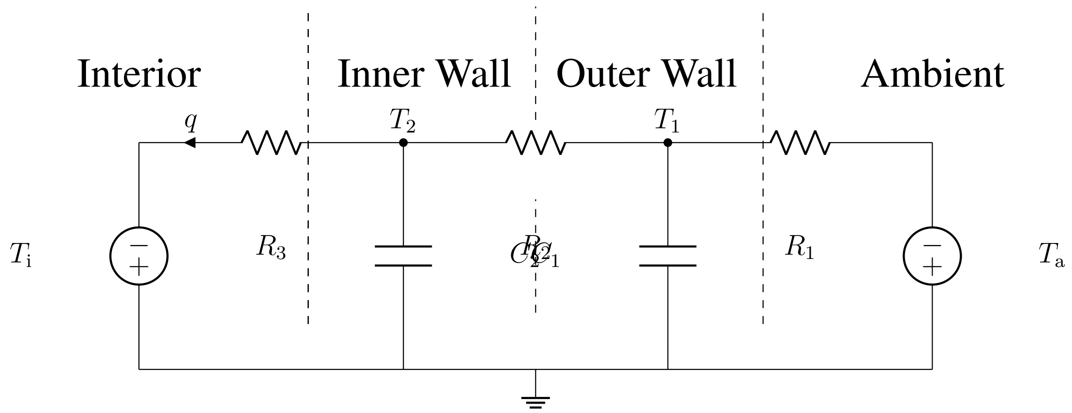


Figure 1.1: RC-diagram of the model.

Chapter 2

Model in CTSM-R

In this section the model is implemented CTSM-R and the results analyzed. Each step is accompanied with the R code for carrying out the calculations.
First some initialization

```
## Set the working directory
setwd(".")
## Use the ctsmr package
library(ctsmr)
```

Then the equations for the model is specified

```
## Initialize the model
model <- ctsm$new()
## Specify the system equations
model$addSystem(dT1 ~ ( 1/(C1*R2) * (T2-T1) + 1/(C1*R1) * (Te-T1) ) * dt + s11 * dw1)
model$addSystem(dT2 ~ ( 1/(C2*R3) * (Ti-T2) + 1/(C2*R2) * (T1-T2) ) * dt + s22 * dw2)
## The inputs
model$addInput("Te","Ti")
## The observation equation
model$addObs( q ~ 1/R3*(Ti-T2) )
model$setVariance(q ~ s)
```

The optimization which is carried out to estimate the parameters, needs initial values for the states for the first time point and initial values for the parameters

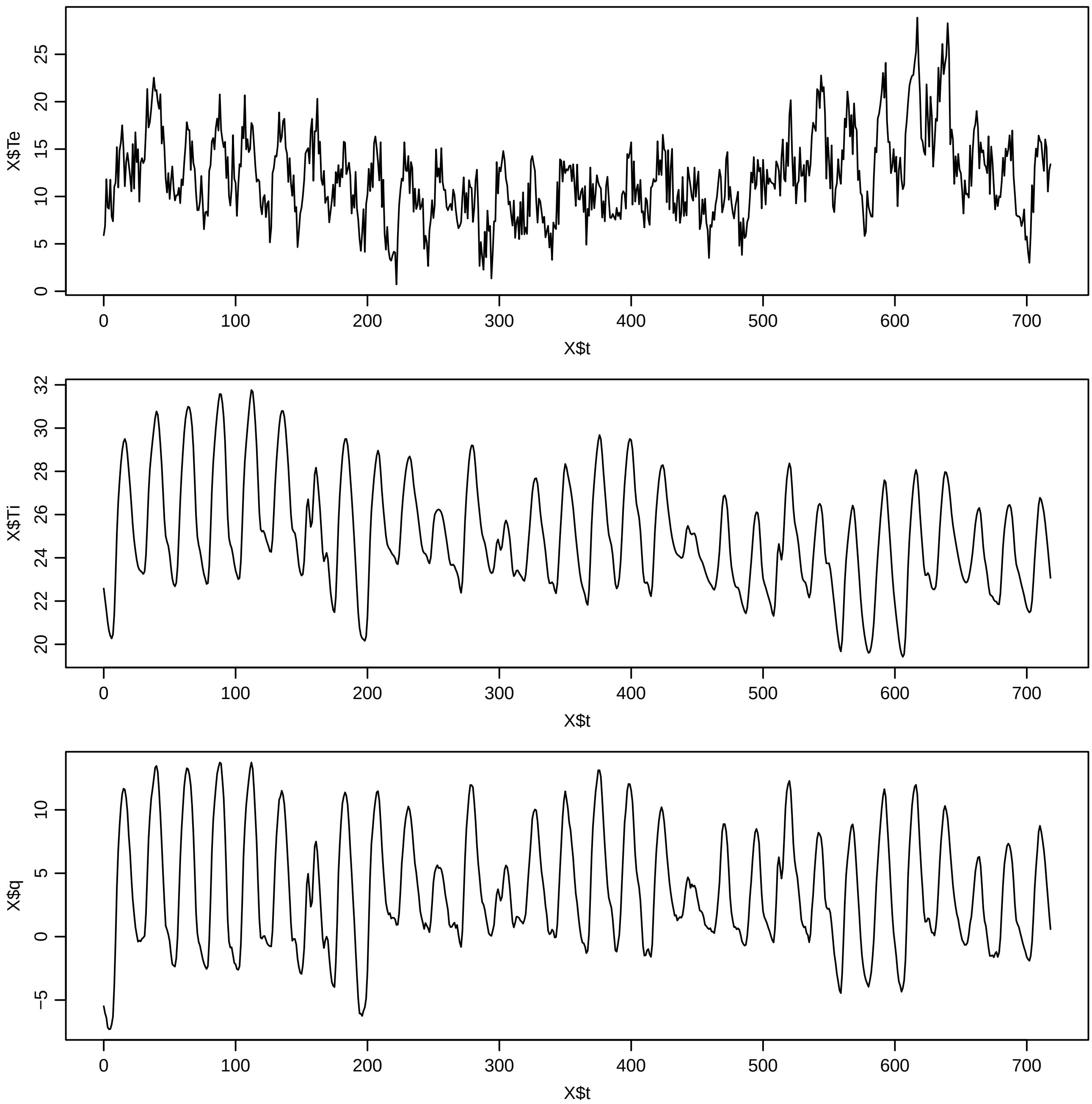
```
## Set initial values of the states for the first time point
model$setParameter( T10 = c(init=15   ,lb=10 ,ub=20 ) )
model$setParameter( T20 = c(init=25   ,lb=20 ,ub=30 ) )
## Set intial values
model$setParameter( C1 =  c(init=100  ,lb=0  ,ub=200) )
model$setParameter( R1 =  c(init=2    ,lb=0  ,ub=5  ) )
model$setParameter( R2 =  c(init=2    ,lb=0  ,ub=5  ) )
model$setParameter( C2 =  c(init=50   ,lb=0  ,ub=100) )
model$setParameter( R3 =  c(init=1    ,lb=0  ,ub=5  ) )
model$setParameter( s11 = c(init=0.01 ,lb=0  ,ub=1  ) )
model$setParameter( s22 = c(init=0.01 ,lb=0  ,ub=1  ) )
model$setParameter( s =   c(init=0.01 ,lb=0  ,ub=1  ) )
```

The data is in the file `data.csv` which is read into the data.frame `X`

```
## Read the data
X <- read.csv("data.csv",sep=";",header=FALSE)
## Set the names of the columns in X
names(X) <- c("t","Te","Ti","q")
```

and the time series are plotted

```
## Plot the data
par(mfrow=c(3,1),mar=c(3,3.5,1,1),mgp=c(2,0.7,0)) # Setup plot
plot(X$t,X$Te,type="l")
plot(X$t,X$Ti,type="l")
plot(X$t,X$q,type="l")
```



The data is specified in the format which is used for the estimation ??This must be changed?? and the parameters are estimated

```
## Execute the estimation
fit <- model$estimate(X)
```

2.0.1 Model validation

The results are analyzed and the model is validated. First, the parameter estimates are printed

```r
## Print the parameters estimates and the Correlation Matrix of the estimates
summary(fit,extended=TRUE)

## Coefficients:
##          Estimate  Std. Error     t value    Pr(>|t|)     dF/dPar dPen/dPar
## T10   1.3134e+01  3.5200e-01  3.7313e+01  0.0000e+00 -5.7503e-06   -0.0008
## T20   2.5330e+01  1.9767e-02  1.2814e+03  0.0000e+00 -3.0244e-04    0.0017
## C1    1.0394e+02  4.5672e+00  2.2759e+01  0.0000e+00 -7.7459e-06    0.0002
## C2    4.9320e+01  2.2905e-01  2.1532e+02  0.0000e+00 -3.2173e-05    0.0002
## R1    9.6509e-01  6.2285e-02  1.5495e+01  0.0000e+00 -3.3369e-06    0.0000
## R2    2.0215e+00  6.4971e-02  3.1113e+01  0.0000e+00 -2.6395e-06    0.0001
## R3    5.0929e-01  4.5443e-04  1.1207e+03  0.0000e+00  1.3823e-04    0.0000
## s     1.0330e-02  4.8738e-04  2.1195e+01  0.0000e+00 -5.4052e-07    0.0000
## s11   5.0709e-16  1.5436e-13  3.2852e-03  9.9738e-01  0.0000e+00    0.0000
## s22   9.0874e-09  9.8934e-07  9.1853e-03  9.9267e-01  2.8498e-10    0.0000
##
## Correlation of coefficients:
##       T10   T20   C1    C2    R1    R2    R3    s     s11
## T20 -0.03
## C1  -0.89 -0.12
## C2  -0.15 -0.32  0.09
## R1   0.97  0.08 -0.95 -0.16
## R2  -0.98 -0.07  0.95  0.15 -1.00
## R3  -0.30  0.05  0.24 -0.01 -0.28  0.28
## s    0.00  0.02  0.00 -0.04  0.01  0.00 -0.05
## s11  0.77  0.16 -0.86 -0.14  0.82 -0.81 -0.21  0.01
## s22  0.77  0.16 -0.86 -0.14  0.82 -0.82 -0.21  0.01  1.00
```

The following four important points are checked (see the section: Model Validation in the CTSM-R User Guide)

- That the p-value of the t-tests (i.e. `Pr(>|t|)`) is below 0.05 for all parameters. This is not the case for σ_1 and σ_2, which is very fine, since they are actually equal to zero in the model used to generate the data.
- That the derivative of the objective function with respect to each parameter (i.e. `dF/dPar`) is close to zero
- That the derivative of the penalty function with respect to each parameter (i.e. `dPen/dPar`) is not significant compared to `dF/dPar`).
- That Correlation Matrix do not have any off-diagonal values close to -1 or 1. Clearly, the correlation of `1.00` (this is a rounded value, the real value is below 1) between `R1` and `R2` is an indication of over-parametrization of the model and thereby an indication of that a simpler model should be used. However, for the present example, where simulated data was generated with the exact same model as estimated, the parameters can be verified exact and it is found (below) that the estimates are close to the true values.

Comparing the true values with estimated values and 95% confidence bounds

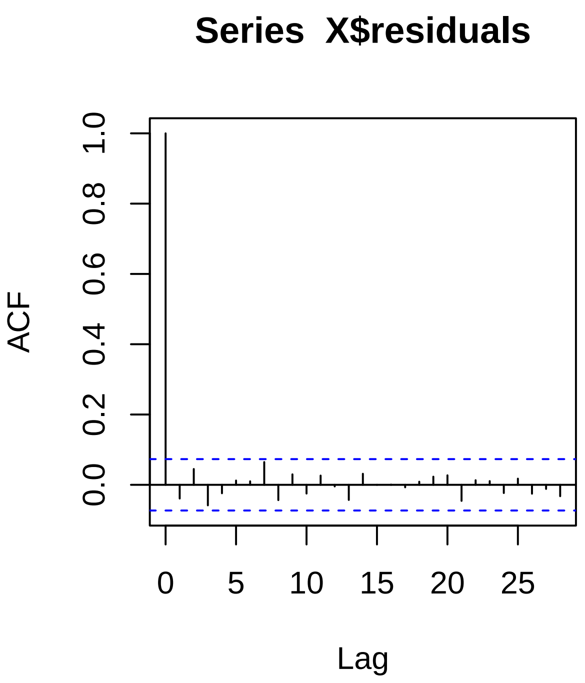
```r
## The parameter estimates in a data.frame
Prm <- data.frame(xm=fit$xm)
row.names(Prm) <- names(fit$xm)
## The true parameter values
Prm$xmtrue <- c(13,25,100,50,1,2,0.5,0.01,0,0)
## Approximately lower 95% confidence bound
```

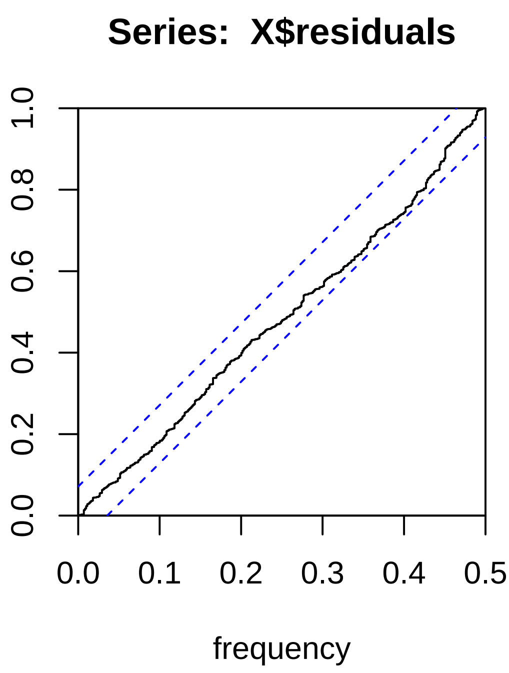
```
Prm$lower <- (fit$xm - 2*fit$sd)
## Approximately upper 95% confidence bound
Prm$upper <- (fit$xm + 2*fit$sd)
## Print it all
##round(Prm,digits=2)
format(Prm,digits=4)

##             xm xmtrue      lower     upper
## T10 1.313e+01  13.00  1.243e+01 1.384e+01
## T20 2.533e+01  25.00  2.529e+01 2.537e+01
## C1  1.039e+02 100.00  9.481e+01 1.131e+02
## C2  4.932e+01  50.00  4.886e+01 4.978e+01
## R1  9.651e-01   1.00  8.405e-01 1.090e+00
## R2  2.021e+00   2.00  1.892e+00 2.151e+00
## R3  5.093e-01   0.50  5.084e-01 5.102e-01
## s   1.033e-02   0.01  9.355e-03 1.130e-02
## s11 5.071e-16   0.00 -3.082e-13 3.092e-13
## s22 9.087e-09   0.00 -1.970e-06 1.988e-06
```

it is found that the estimated values are close to the true values and for most of the (except `C2` and `R3`) contained within the confidence bands.
The one-step ahead predictions are calculated and the white-noise properties of the residuals (one-step ahead prediction residuals) are analyzed with the auto-correlation function and the cumulated periodogram

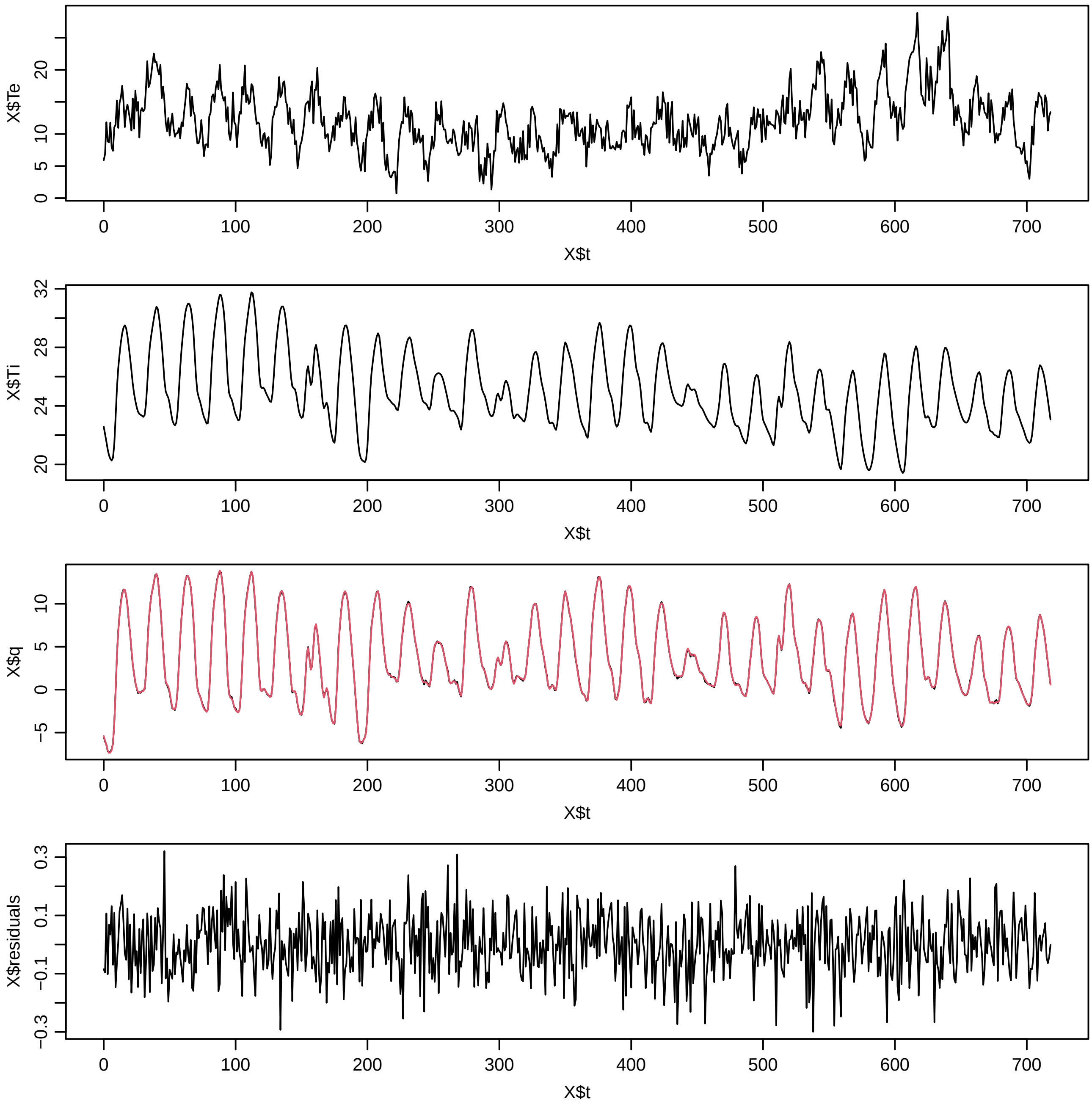
```
## Calculate the one-step ahead predictions
tmp <- predict(fit, newdata=X)
## Calculate the residuals and put them with the data in a data.frame X
X$qHat <- tmp$output$pred$q
X$residuals <- X$q - X$qHat
## Auto-correlation function and cumulated periodogram
par(mfrow=c(1,2)) # Setup plot
acf(X$residuals)
cpgram(X$residuals)
```





Finally, it is important to verify the model by studying the time series plots of the inputs, output, one-step ahead predictions and the residuals

```
## Plot of inputs, output, one-step ahead predicted output, and residuals
par(mfrow=c(4,1),mar=c(3,3.5,1,1),mgp=c(2,0.7,0)) # Setup plot
plot(X$t,X$Te,type="l")
plot(X$t,X$Ti,type="l")
plot(X$t,X$q,type="l")
lines(X$t,X$qHat,col=2)
plot(X$t,X$residuals,type="l")
```



It is found that there is no distinct patterns in the residuals, which could be related to unmodeled dynamics of the wall or related to any of the inputs.