DYNASTEE DYNamic Analysis, Simulation and Testing applied to the Energy and Environmental performance of buildings

Free On-line Training Webinars | 2-9-16-23-30 September 2020

Dynamic Calculation Methods for Building Energy Performance Assessment

Technical University of Denmark DTU | GCU Glasgow Caledonian University | Universidad del País Vasco / Euskal Herriko Unibertsitatea | Gobierno de España, Ministerio de Economía y Competitividad | Ciemat Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas | University of Salford Manchester

DYNASTEE

DYNASTEE DYNamic Analysis, Simulation and Testing applied to the Energy and Environmental performance of buildings

Free On-line Training Webinars | 2-9-16-23-30 September 2020

Peder Bacher

Technical University of Denmark DTU

Irati Uriarte

Universidad del País Vasco / Euskal Herriko Unibertsitatea

Webinar management

Maria Kapsalaki (INIVE, BE)

Valérie Leprince (INIVE, BE)

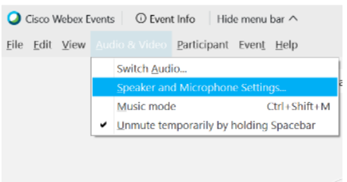
Disclaimer: The sole responsibility for the content of presentations and information given orally during DYNASTEE webinars lies with the authors. It does not necessarily reflect the opinion of DYNASTEE. Neither DYNASTEE nor the authors are responsible for any use that may be made of information contained therein.

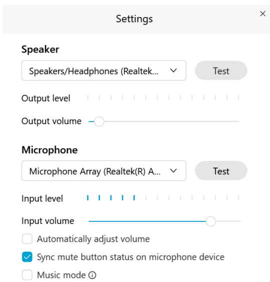
DYNASTEE DYNamic Analysis, Simulation and Testing applied to the Energy and Environmental performance of buildings

Free On-line Training Webinars | 2-9-16-23-30 September 2020

If you can't hear the webinar sound

Make sure that Audio Connection is on by clicking on Audio & Video / Speaker and Microphone Settings





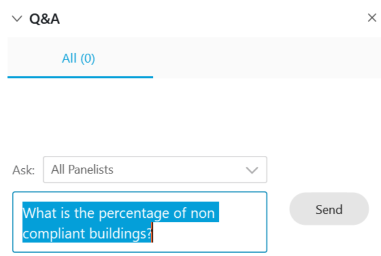
DYNASTEE DYNamic Analysis, Simulation and Testing applied to the Energy and Environmental performance of buildings

Free On-line Training Webinars | 2-9-16-23-30 September 2020

How to ask questions during the webinar

Locate the **Q&A box**

Select **All Panelists** | Type your question | Click on Send



Note: Please DO NOT use the chat box to ask your questions!

DYNASTEE DYNamic Analysis, Simulation and Testing applied to the Energy and Environmental performance of buildings

Free On-line Training Webinars | 2-9-16-23-30 September 2020

Organized by https://dynastee.info/ Facilitated by INIVE

NOTES:

- The webinar will be recorded and published at https://dynastee.info/ within a couple of weeks, along with the presentation slides.
- In case your questions have not been answered please send them to Peder Bacher (pbac@dtu.dk) or Irati Uriarte(irati.uriarte@ehu.eus);
- All remaining questions will be answered during the last webinar of September 30th

Introduction to statistical modelling of dynamical systems

Peder Bacher

DYNASTEE On-line Training for Dynamic Calculation Methods for Building Energy Performance Assessment

Webinar 2020

September 23, 2020

Overview

Data analysis and statistics

Statistical inference

- "Everything should be made as simple as possible, but not simpler" (Einstein)
- **Which model**? and **how complex** should it be? **Depends on data!**
- Statistics provide the techniques to:
  - **Estimate model parameters** and their uncertainties
  - Verify and argue that you have found the best model (or rather there is not one best model, so we call it a **suitable model**)

We can: Extract information and draw conclusions from data

We can: Train models for prediction and use them as basis for optimization

Time series analysis

Statistical modeling of dynamical systems

- Called time series analysis
- Tons of literature (and software):
  - Wiener, N. (1949). Extrapolation, Interpolation, and Smoothing of Stationary Time Series. The MIT Press
  - Box, G., Jenkins, G. (1976). *Time series analysis: forecasting and control*
  - ...
- Used in any thinkable application!

Time series models

General types of models *(can all be tweaked!)*:

- Static model *no dynamics*
- ARMAX, *discrete models* based on transfer functions
- Grey-box, *continuous time models*, combination of physics and statistics (stochastic differential equations (SDEs))

Static model *(linear function)*

$$\text{Measurements} = \text{Function}(\text{Inputs}) + \text{Error}$$

Discrete ARX model *(Auto-Regressive with eXogenous input)*

$$\text{Measurements} = \text{TransferFun}(\text{Inputs}) + \text{Error}$$

Discrete ARMAX model *(Auto-Regressive Moving Average with eXogenous input)*

Statistical model validation: examine the residuals

Residuals from a *simple linear regression model*

$$Y_t = \beta_0 + \beta_1 x_t + \varepsilon_t$$

$$y_t = \hat{\beta}_0 + \hat{\beta}_1 x_t + \hat{\varepsilon}_t$$

$$y_t = \hat{y}_t + \hat{\varepsilon}_t$$

$$\hat{\varepsilon}_t = y_t - \hat{y}_t$$

$$\text{Residual}_t = \text{Observation}_t - \text{Prediction}_t$$

Two assumptions:

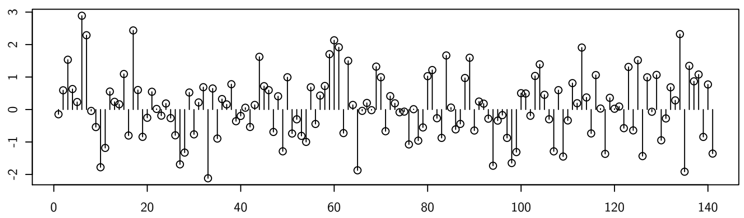
1. The error is normal distributed: $\varepsilon_t \sim N(0, \sigma^2)$ (less important with many obs.)
2. The error is *independent and identically distributed* (i.i.d.):
   - Check ε_t is not dependent on other variables

Do you know about:

- White noise?
- AutoCorrelation Function (ACF)?

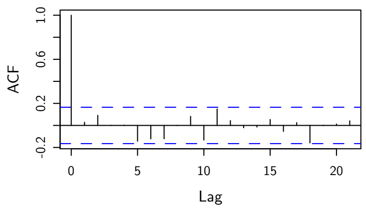
ACF of white noise

```
## Plot
x <- rnorm(141)
plot(x, type="n", xlab="Time", ylab="")
points(x)
lines(x, type='h')
```



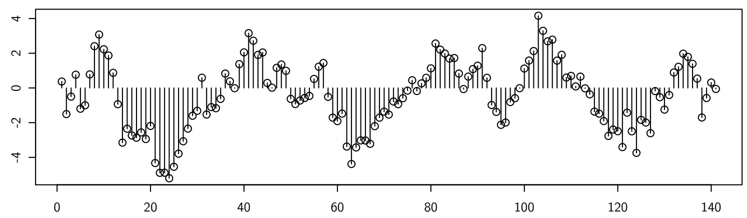
The ACF:

```
## Autocorrelation function
acf(x)
```



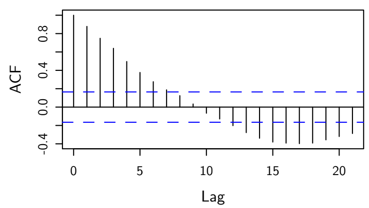
ACF of non-white noise

```
## Plot
x <- filter(rnorm(141), 0.9, "recursive")
plot(x, type="n", xlab="Time", ylab="")
points(x)
lines(x, type='h')
```



The ACF:

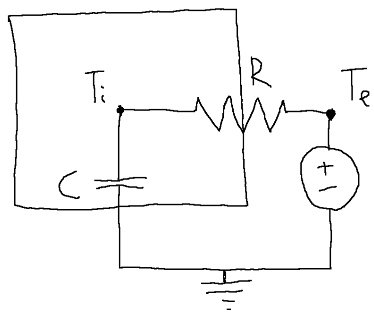
```
## Autocorrelation function
acf(x)
```



We want white noise!

- We fit the model and then analyze the residuals
- If they are *not* white noise, then we can still improve the model!

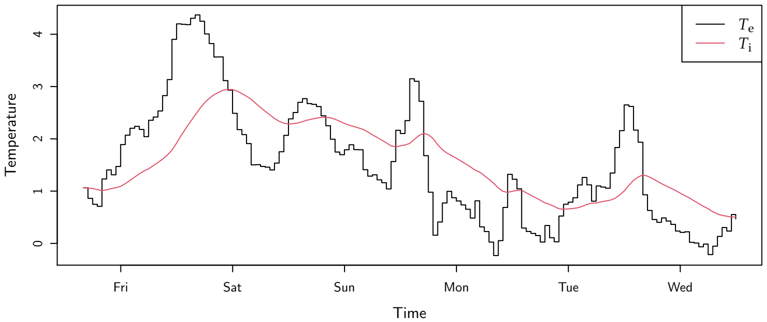
Simplest first order RC-system



Simplest RC-system

- T_t^{e} external and T_t^{i} internal temperature at time $t = [1, 2, \dots, n]$
- ODE model

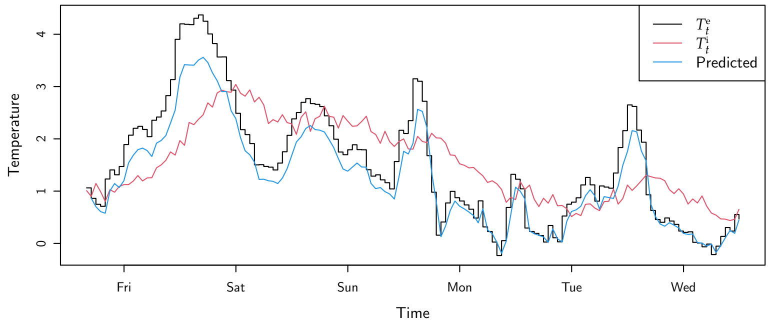
$$\frac{dT_{\mathrm{i}}}{dt} = \frac{1}{RC}(T_{\mathrm{e}} - T_{\mathrm{i}})$$



Try a static model

- A simple linear regression model (ε_t is the error)
- Not describing dynamics

$$T_t^{\mathrm{i}} = \omega_{\mathrm{e}} T_t^{\mathrm{e}} + \varepsilon_t$$

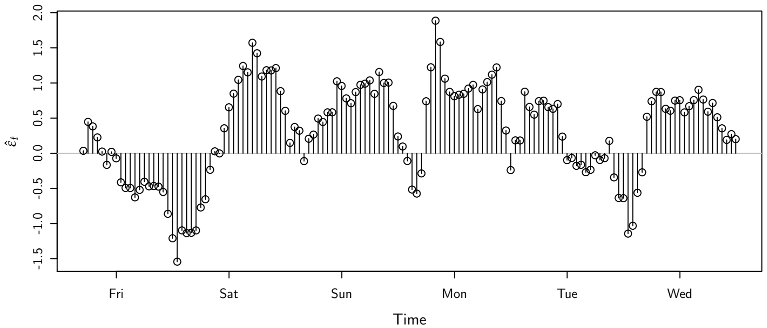


Model validation: check i.i.d. of residuals

Are residuals like white noise?

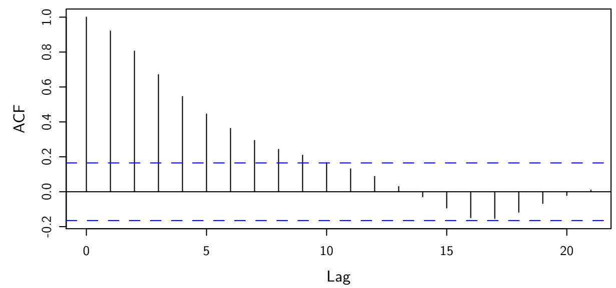
- Check if they are *independent and identically distributed*
- Is $\hat{\varepsilon}_t$ independent of $\hat{\varepsilon}_{t-k}$ for all t and k?

Nope! There is a pattern left...



Model validation: Test for i.i.d. with ACF

TEST if residuals independent of each other using the *Auto Correlation Function*?



It's not white nose! How do we find a better model?

Discretize the ODE

$$\frac{dT_i}{dt} = \frac{1}{RC}(T_e - T_i)$$

It has the solution

$$T_i(t+\Delta t) = T_e(t) + e^{-\frac{\Delta t}{RC}}\left(T_i(t) - T_e(t)\right)$$

if $\Delta t = 1$ and T_e is constant between the sample points then

$$T^i_{t+1} = e^{-\frac{1}{RC}} T^i_t + (1 - e^{-\frac{1}{RC}}) T^e_t$$

since $e^{-\frac{1}{RC}}$ is between 0 and 1, then write it as

$$T^i_{t+1} = \phi_1 T^i_t + \omega_1 T^e_t$$

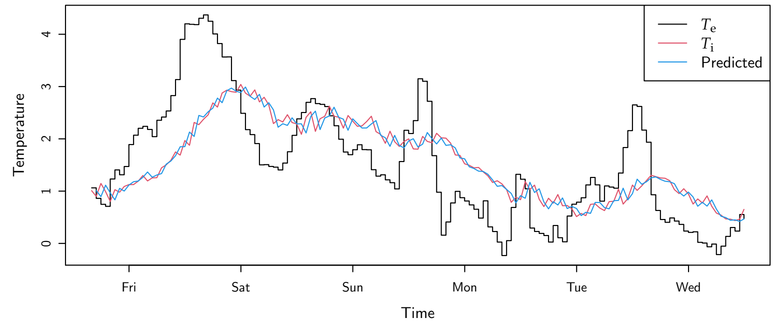
where ϕ_1 and ω_1 are between 0 and 1.

Add a noise term and we have the ARX model

$$T^i_{t+1} = \phi_1 T^i_t \quad + \omega_1 T^e_t \quad + \varepsilon_{t+1} T^i_t \qquad = \phi_1 T^i_{t-1} + \omega_1 T^e_{t-1} + \varepsilon_t$$

An ARX model

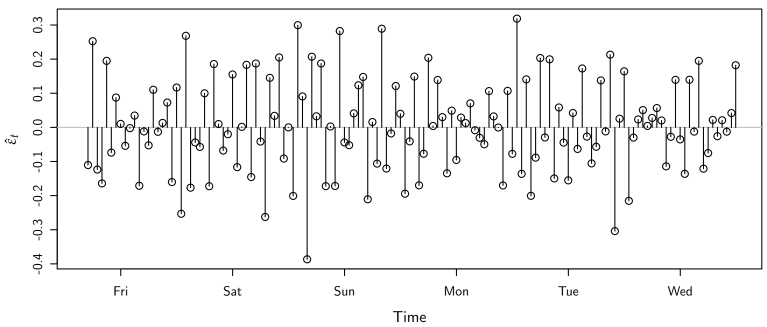
$$T^i_t = \phi_1 T^i_{t-1} + \omega_1 T^e_{t-1} + \varepsilon_t$$



ARX model

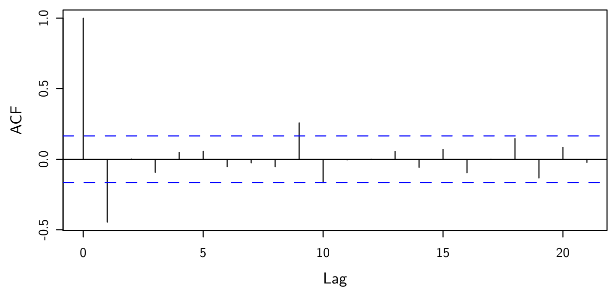
The residuals

$$\hat{\varepsilon}_t = T^i_t - \frac{\hat{\omega}_1 \mathrm{B}}{1 - \hat{\phi}_1 \mathrm{B}} T^e_t$$



Check for i.i.d. of residuals

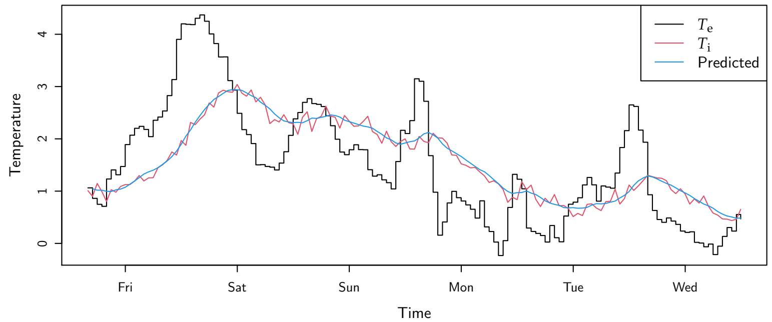
Is it likely that this is white noise? Almost!



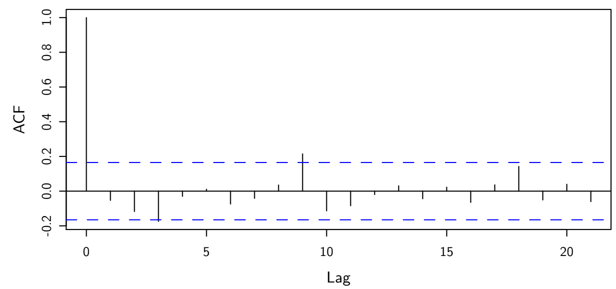
Actually we miss an MA part!

An ARMAX model

$$T^{\mathrm{i}}_t = \phi_1 T^{\mathrm{i}}_{t-1} + \omega_1 T^{\mathrm{e}}_t + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$



Validate the model with the residuals ACF



Now we have *white noise residuals*, that is want to have after applying the model!

Note that we are validating the *one-step prediction* residuals: $\hat{\varepsilon}_{t+1} = y_{t+1} - \hat{y}_{t+1|t}$
$\hat{\varepsilon}_t = y_t - \hat{y}_{t|t-1}$

Auto-regressive (AR) model

AR model of order 1

$$Y_t = \phi_1 Y_{t-1} + \varepsilon_t$$

ARX model of order 1

$$Y_t = \phi_1 Y_{t-1} + \omega_1 X_{t-1} + \varepsilon_t$$

ARMAX model of order 1

$$Y_t = \phi_1 Y_{t-1} + \omega_1 X_{t-1} + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

where $\varepsilon_t \sim N(0, \sigma^2)$ and i.i.d.
Use either X or U as the input (just a variable name in the generalized form).

Discrete linear time series models

AR model of order p

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t$$

ARX model of order p

$$\begin{aligned} Y_t = \phi_1 Y_{t-1} + \ldots + \phi_p Y_{t-p} \\ + \omega_1 X_{t-1} + \ldots + \omega_p X_{t-p} \\ + \varepsilon_t \end{aligned}$$

ARMAX model of order p

$$\begin{aligned} Y_t = \phi_1 Y_{t-1} + \ldots + \phi_p Y_{t-p} \\ + \omega_1 X_{t-1} + \ldots + \omega_p X_{t-p} \\ + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_p \varepsilon_{t-p} \end{aligned}$$

where $\varepsilon_t \sim N(0, \sigma^2)$ and i.i.d.
Doesn't need to have same order p for the AR, X and MA parts.

Discrete linear time series models

AR model

$$\phi(B)Y_t = \varepsilon_t$$

ARX model

$$\phi(B)Y_t = \omega(B)X_t + \varepsilon_t$$

ARMAX model

$$\phi(B)Y_t = \omega(B)X_t + \theta(B)\varepsilon_t$$

- $\varepsilon_t \sim N(0, \sigma^2)$ and i.i.d.
- B is the back-shift operator $\mathrm{B}^k Y_t = Y_{t-k}$
- $\phi(B) = 1 + \phi_1 B + \phi_2 B^2 + \ldots + \phi_p B^p$
- $\omega(B) = \omega_1 B + \omega_2 B^2 + \ldots + \omega_p B^p$
- $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q$

Discrete linear time series models

On transfer function form

ARMAX model

$$Y_t = \frac{\omega(B)}{\phi(B)} X_t + \frac{\theta(B)}{\phi(B)} \varepsilon_t$$
$$\Leftrightarrow$$
$$Y_t = H_\omega(B) X_t + H_\theta(B)\varepsilon_t$$

where $H_\omega(B)$ and $H_\theta(B)$ are a transfer functions

How to estimate parameters in discrete TS models

Fit (in R)

- ARX models with linear regression (closed form optimization, always give the optimum, in R `lm()`)
- ARMA in R is in `arima()`
- ARMAX in R can be fitted with the `marima` and several other packages

And we can tweak and also make non-linear discrete models in many ways!

Continuous time series models

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

ctsmr

Continuous Time Stochastic Modelling in R

more correctly

Continuous-Discrete Time Stochastic Modelling in R

Grey-box modelling

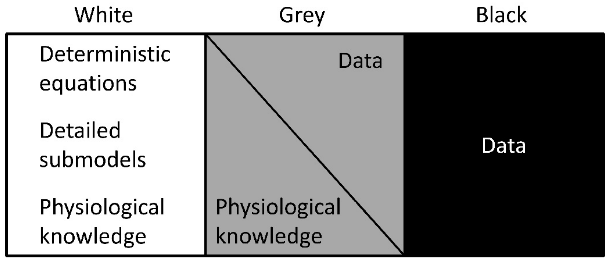


Figure: Ak et al. 2012

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

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

$$Y_k = h(X_{t_k}, u_t, t, \theta) + e_k \quad , \quad e_k \in N(0, S(u_k, t_k, \theta))$$

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

Write up the physical model!

This is easier to work with (if you know the physics behind the system)!

The ODE

$$\frac{dT_i}{dt} = \frac{1}{RC}(T_e - T_i)$$

Just needs a diffusion term to make into the *system equation*

$$\overset{\text{state}}{dT_i} = \overset{\text{drift term}}{\frac{1}{RC}(T_e - T_i)dt} + \overset{\text{diffusion term}}{\sigma_i d\omega}$$

and together with the *measurement equation*

$$\overset{\text{observation}}{Y_{T_i,k}} = \overset{\text{state}}{T_{i,t_k}} + \overset{\text{error}}{e_k} \quad , \quad e_k \in N(0, \sigma) \text{ and i.i.d.}$$

it forms a *grey-box model.*

Wuuups

This particular models are actually unidentifiable!!

R and C cannot be separated (change one, then the other accordingly and the model prediction is equal (same goes for ϕ_1 and ω_1))

The time constant $RC = \tau$ is used instead

$$dT_i = \frac{1}{RC\tau}(T_e - T_i)dt + \sigma_i d\omega$$

Define a GB model

Install the `ctsm-r` package from `ctsm.info`.

Define the model:

```
## Generate a new object of class ctsm
model <- ctsm$new()
## Add a system equation and thereby also a state
model$addSystem(dTi ~ ( 1/tau*(Te-Ti) )*dt + exp(p11)*dw1)
## Set the names of the inputs
model$addInput(Te)
## Set the observation equation: Ti is the state, yTi is the measured output
model$addObs(yTi ~ Ti)
## Set the variance of the measurement error
model$setVariance(yTi ~ exp(e11))
```


Fit the model

Set initial values and bounds for the estimation:

```
## Set the initial value (for the optimization) of the value of the state at the starting time point
model$setParameter(  Ti = c(init=5     ,lb=-5  ,ub=20 ) )
## Set the initial value for the optimization
model$setParameter( tau = c(init=10   ,lb=1E-2 ,ub=200 ) )
model$setParameter( p11 = c(init=0.01 ,lb=-30  ,ub=10 ) )
model$setParameter( e11 = c(init=0.01 ,lb=-50  ,ub=10 ) )
```

Run the parameter estimation:

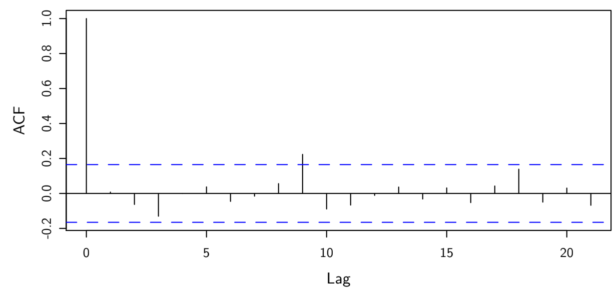
```
fit <- model$estimate(X)
```


Validate the model

Check the *one-step prediction* residuals:

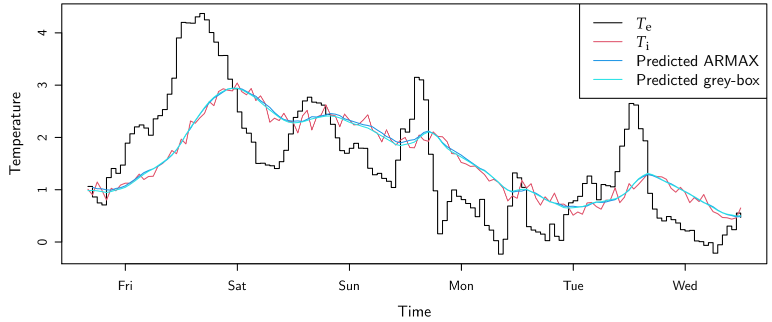
```
# Teke the one-step predictions from the fit
val <- predict(fit)[[1]]
# Calculate the residuals
residualsgb <- unlist(X$yTi - val$output$pred)

# The autocorrelation function
acf(residualsgb)
```



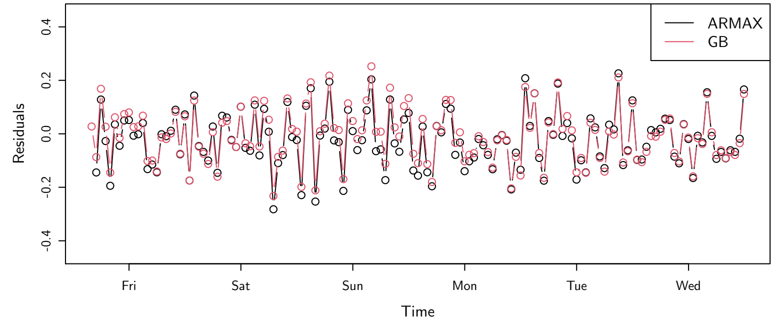
Discrete ARMAX is equivalent to continuous SDE model

One-step predictions of ARMAX and grey-box are almost equal:



Discrete ARMAX is equivalent to continuous SDE model!

Plot the ARMAX and GB residuals:



Parameter estimation with the likelihood

An example, we have:

- A model with two parameters $Y_i \sim N(\mu, \sigma^2)$
- n observations $(y_1, y_2, \ldots, y_n)$

The likelihood is defined by the joint probability density function (pdf) of the observations

$$L(\mu, \sigma) = p(y_1, y_2, \ldots, y_n | \mu, \sigma)$$

Hence, the model defines the pdf as a function of the parameters (the observations are not varying).

Independence of the observations simplifies it to

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

Maximum likelihood estimation

Maximum likelihood estimation (MLE)

Parameter estimation by maximizing the likelihood function

$$\hat{\theta} = \arg\max_{\theta \in \Theta} (L(\theta))$$

Due to numerical properties we always minimize the negative log-likelihood

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

So in the example $\theta = (\mu, \sigma)$

Likelihood for time correlated data

Given a time series of measurements $\mathcal{Y}_N$

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

Essentially, $p(y_k|\mathcal{Y}_{k-1}, \theta)$ is the pdf of the one-step ahead prediction

Thus assuming independence of the one-step predictions (so i.i.d. error)

Likelihood for time correlated data

If Gaussian

$$\begin{aligned} \hat{y}_{k|k-1} &= E[y_k|\mathcal{Y}_{k-1}, \theta] \\ P_{k|k-1} &= V[y_k|\mathcal{Y}_{k-1}, \theta] \\ \varepsilon_k &= y_k - \hat{y}_{k|k-1} \end{aligned}$$

then the likelihood is

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

Kalman filter

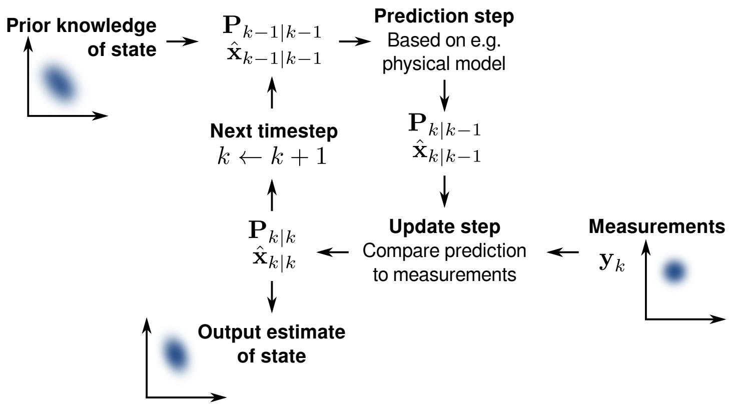


Figure: "Basic concept of Kalman filtering" by Petteri Aimonen. Wikipedia

Grey-box model MLE

Steps for maximum likelihood estimation of a grey-box model

1. Load data
2. Define a model
3. Define initial values and parameter bounds
4. Run an optimizer to find the parameter values maximizing the likelihood (run the Kalman filter many times)
5. Interpret and validate the result:
   - Check the optimizer convergence (e.g. no parameters at bounds)
   - Check estimated values and statistics
   - Validate the model by analyzing residuals

Show an example in R

Model complexity

The big question!!

How to select a *suitable* model complexity, neither underfitted nor overfitted!
Both which inputs, the structure. Number of parameters increase complexity.

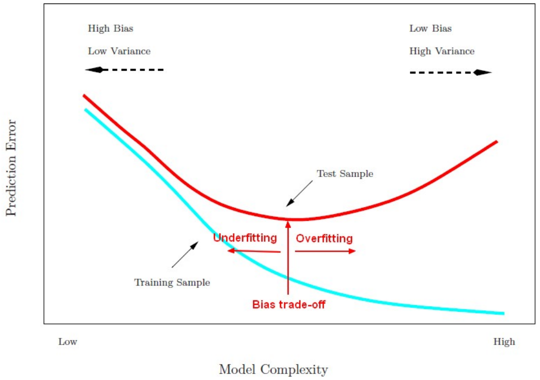


Figure from https://gerardnico.com/data_mining/bias_trade-off.

Model selection

The suitable model is a *compromise*:

- Not too complex (overfitted) and not too simple (underfitted).
- Use statistical tests to find out which model is better:
  - Nested models, use e.g. F-test or *likelihood ratio-test*
  - Un-nested models, use e.g. AIC or BIC

Different strategies:

- Forward selection: Start with the simplest model and extend step-wise
- Backward selection: Start with the full model and remove terms step-wise

ctsmr R package

See the website ctsm.info

- Installation needs compilers
- Documentation and examples
- Nice tricks
- Literature list with overview of studies where ctsm has been used