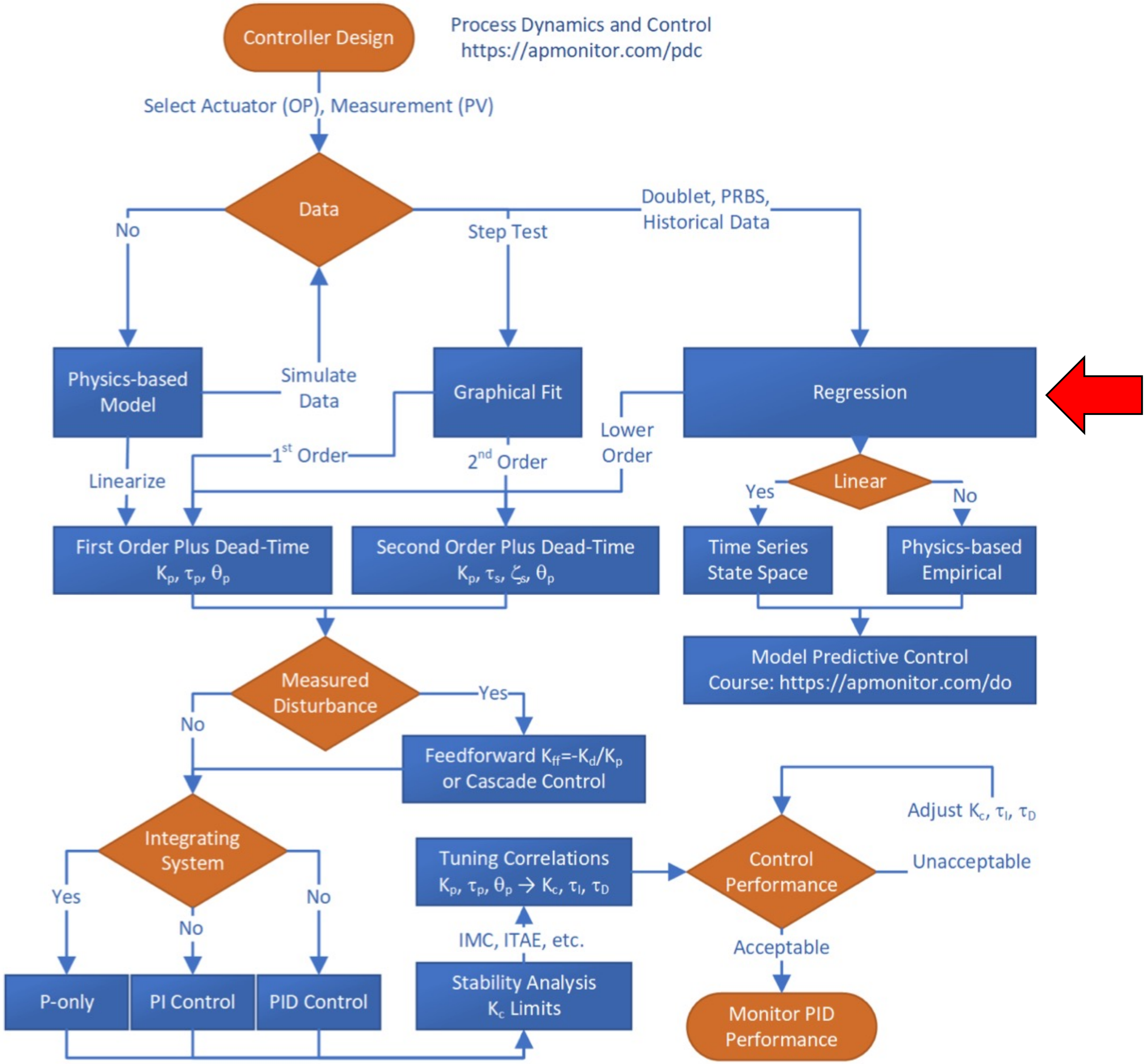
Data-driven model learning of dynamical systems

Review of linear regression and least squares estimation

Hugues GARNIER

hugues.garnier@univ-lorraine.fr

Model-based


Controller Design
Process Dynamics and Control
https://apmonitor.com/pdc
Select Actuator (OP), Measurement (PV)
Data
No
Step Test
Doublet, PRBS, Historical Data
Physics-based Model
Simulate Data
Graphical Fit
Regression
Linearize
1st Order
2nd Order
Lower Order
Linear
Yes
No
First Order Plus Dead-Time
K_p, τ_p, θ_p
Second Order Plus Dead-Time
$K_p, \tau_s, \zeta_s, \theta_p$
Time Series State Space
Physics-based Empirical
Model Predictive Control
Course: https://apmonitor.com/do
Measured Disturbance
Yes
No
Feedforward $K_{ff}=-K_d/K_p$
or Cascade Control
Integrating System
Yes
No
No
P-only
PI Control
PID Control
Tuning Correlations
$K_p, \tau_p, \theta_p \rightarrow K_c, \tau_I, \tau_D$
IMC, ITAE, etc.
Stability Analysis
K_c Limits
Control Performance
Adjust K_c, τ_I, τ_D
Unacceptable
Acceptable
Monitor PID Performance

Review of linear regression and least squares estimation

1. Least squares-based model estimation for static systems

2. Least squares-based model estimation for dynamical systems

Regression

- Prediction of variable y on the basis of information provided by other measured variables $\varphi_1, ..., \varphi_d$.

- Collect $\varphi = \begin{bmatrix} \varphi_1 \\ \vdots \\ \varphi_d \end{bmatrix}$.

- Problem: find function of the regressors $g(\varphi)$ that minimises the difference $y - g(\varphi)$ in some sense.
  So $\widehat{y} = g(\varphi)$ should be a good prediction of y.

- Example in a stochastic framework: minimise $E[y - g(\varphi)]^2$.

Linear regression

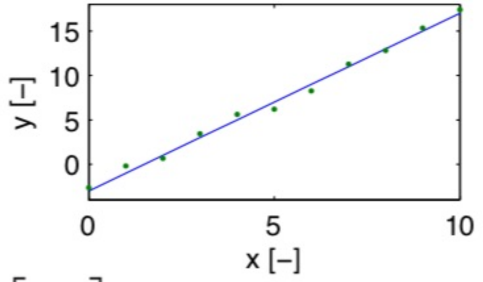
- Regression function $g(\varphi)$ is parameterised. It depends on a set of parameters
$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}.$$

- Special case: regression function $g(\varphi)$ is *linear in the parameters* θ. Note that this does **not** imply any linearity with respect to the variables from φ.

- Special case: $g(\varphi) = \theta_1\varphi_1 + \theta_2\varphi_2 + \ldots + \theta_d\varphi_d$
So $g(\varphi) = \varphi^T\theta$.

Linear regression - Examples

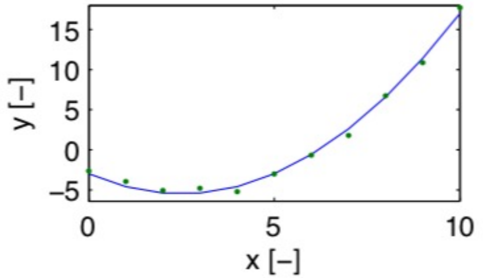
Linear regression — Examples:



- Linear fit $y = ax + b$.

  Then $g(\varphi) = \varphi^T \theta$ with input vector $\varphi = \begin{bmatrix} x \\ 1 \end{bmatrix}$

  and parameter vector $\theta = \begin{bmatrix} a \\ b \end{bmatrix}$. So: $g(\varphi) = \begin{bmatrix} x & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$.



- Quadratic function $y = c_2 x^2 + c_1 x + c_0$.

  Then $g(\varphi) = \varphi^T \theta$ with input vector $\varphi = \begin{bmatrix} x^2 \\ x \\ 1 \end{bmatrix}$

  and parameter vector $\theta = \begin{bmatrix} c_2 \\ c_1 \\ c_0 \end{bmatrix}$. So: $g(\varphi) = \begin{bmatrix} x^2 & x & 1 \end{bmatrix} \begin{bmatrix} c_2 \\ c_1 \\ c_0 \end{bmatrix}$.

UNIVERSITÉ DE LORRAINE

POLYTECH NANCY

Least squares estimate

- N measurements $y(t), \varphi(t), \qquad t = 1, ..., N.$

- Minimise $V_N(\theta) = \dfrac{1}{N}\sum_{t=1}^{N}[y(t) - g(\varphi(t))]^2.$

- So a suitable θ is $\widehat{\theta}_N = \arg\min V_N(\theta).$

- Linear case $V_N(\theta) = \dfrac{1}{N}\sum_{t=1}^{N}[y(t) - \varphi^T(t)\theta]^2.$

Least squares estimate (1)

- In the linear case the "cost" function $V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}[y(t) - \varphi^T(t)\theta]^2$ is a quadratic function of θ.

- It can be minimised analytically: All partial derivatives $\frac{\partial V_N(\theta)}{\partial \theta}$ have to be zero in the minimum:
  $\frac{1}{N}\sum_{t=1}^{N} 2\varphi(t)[y(t) - \varphi^T(t)\theta] = 0$
  The solution of this set of equations is the parameter estimate $\widehat{\theta}_N$.

UNIVERSITÉ DE LORRAINE

POLYTECH NANCY

Least squares estimate (2)

- A *global* minimum is found for $\widehat{\theta}_N$ that satisfies a set of linear equations, the *normal equations*

$$\left[\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right]\widehat{\theta}_N = \frac{1}{N}\sum_{t=1}^{N}\varphi(t)y(t).$$

- If the matrix on the left is invertible, the LSE is

$$\widehat{\theta}_N = \left[\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right]^{-1}\frac{1}{N}\sum_{t=1}^{N}\varphi(t)y(t).$$

Least squares estimate
Recommended matrix formulation

- Collect the output measurements in the vector $Y_N = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}$,

  and the inputs in the $N \times d$ regression matrix $\Phi_N = \begin{bmatrix} \varphi^T(1) \\ \vdots \\ \varphi^T(N) \end{bmatrix}$.

- Normal equations: $\left[\Phi_N^T \Phi_N\right] \hat{\theta}_N = \Phi_N^T Y_N$.

- Estimate $\boxed{\hat{\theta}_N = \Phi_N^{\dagger} Y_N}$

  (Moore-Penrose) *pseudoinverse* of Φ_N: $\Phi_N^{\dagger} = \left[\Phi_N^T \Phi_N\right]^{-1} \Phi_N^T$.

  Note: $\Phi_N^{\dagger} \Phi_N = I$.

Linear least-squares estimate in Matlab

Solution x of overdetermined $Ax = b$ with rectangular matrix A,
so more equations than unknowns, or
more rows than columns, or
A is m-by-n with $m > n$ and full rank n

Then least squares solution $\hat{x} = A^{\dagger}b$

In Matlab:

```
x = A\b;                 % Preferred
x = pinv(A)*b;
x = inv(A'*A)*A'*b;
```

Example 1: The "well-known" linear fit $y = ax + b$

Measurements x_i and y_i for $i = 1, \ldots, N$.

Cost function $V_N = \frac{1}{N}\sum(y_i - ax_i - b)^2$.

1) "Manual" solution: $\dfrac{\partial V_N}{\partial a} = 0$ and $\dfrac{\partial V_N}{\partial b} = 0$, so

$$\left\{\begin{array}{rcl} \sum -x_i(y_i - ax_i - b) & = & 0 \\ \sum -(y_i - ax_i - b) & = & 0 \end{array}\right. \Leftrightarrow \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & \sum 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$$

Parameter estimate: $\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & \sum 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$

```
Matlab code
x=[…];
y=[…];
theta_hat=inv([sum(x.^2) sum(x); ...
               sum(x) length(x)])*[sum(x.*y);sum(y)]
```

Example 1: The "well-known" linear fit $y = ax + b$

Matrix formulation

Measurements x_i and y_i for $i = 1, \ldots, N$.

Cost function $V_N = \frac{1}{N}\sum(y_i - ax_i - b)^2$.

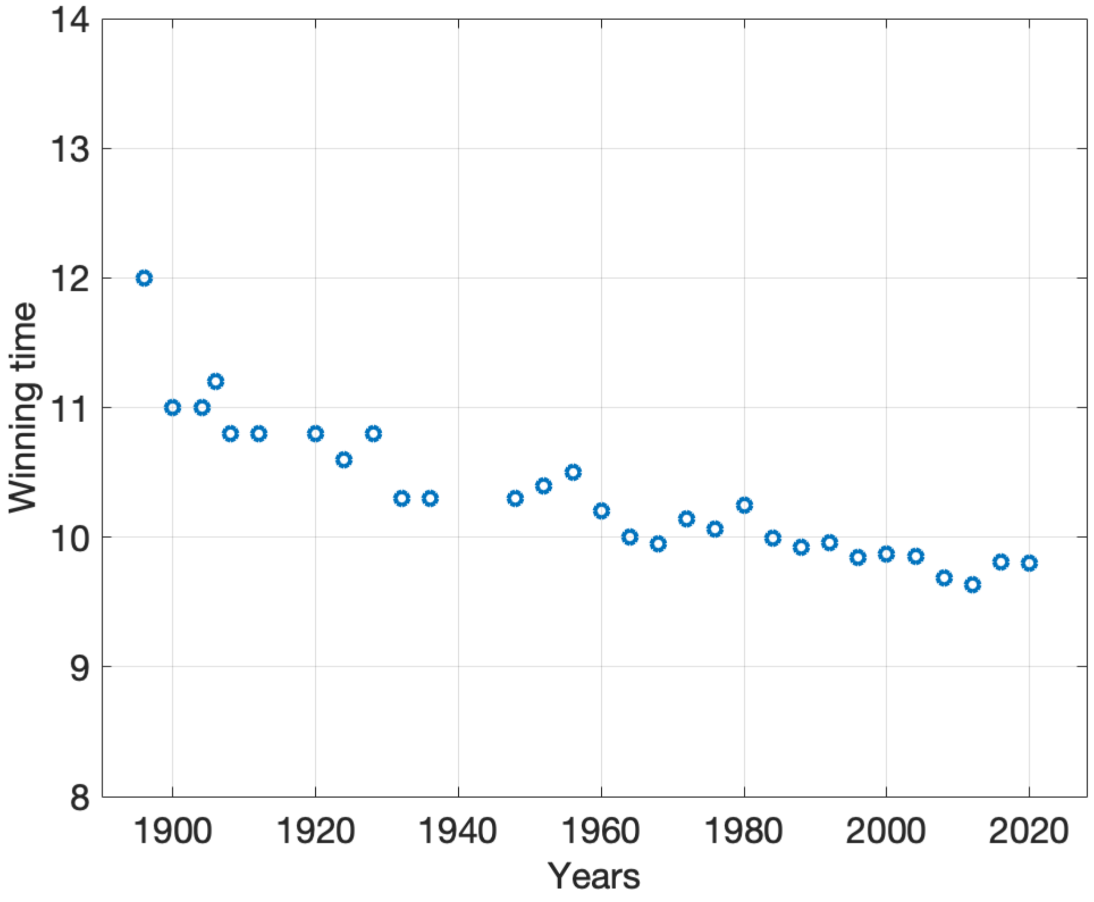
2) Matrix solution: $Y_N = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}$, $\Phi_N = \begin{bmatrix} x(1) & 1 \\ \vdots & \vdots \\ x(N) & 1 \end{bmatrix}$ and $\theta = \begin{bmatrix} a \\ b \end{bmatrix}$.

Cost function (in vector form) $V_N = \frac{1}{N}||Y_N - \Phi_N\theta||_2^2$.

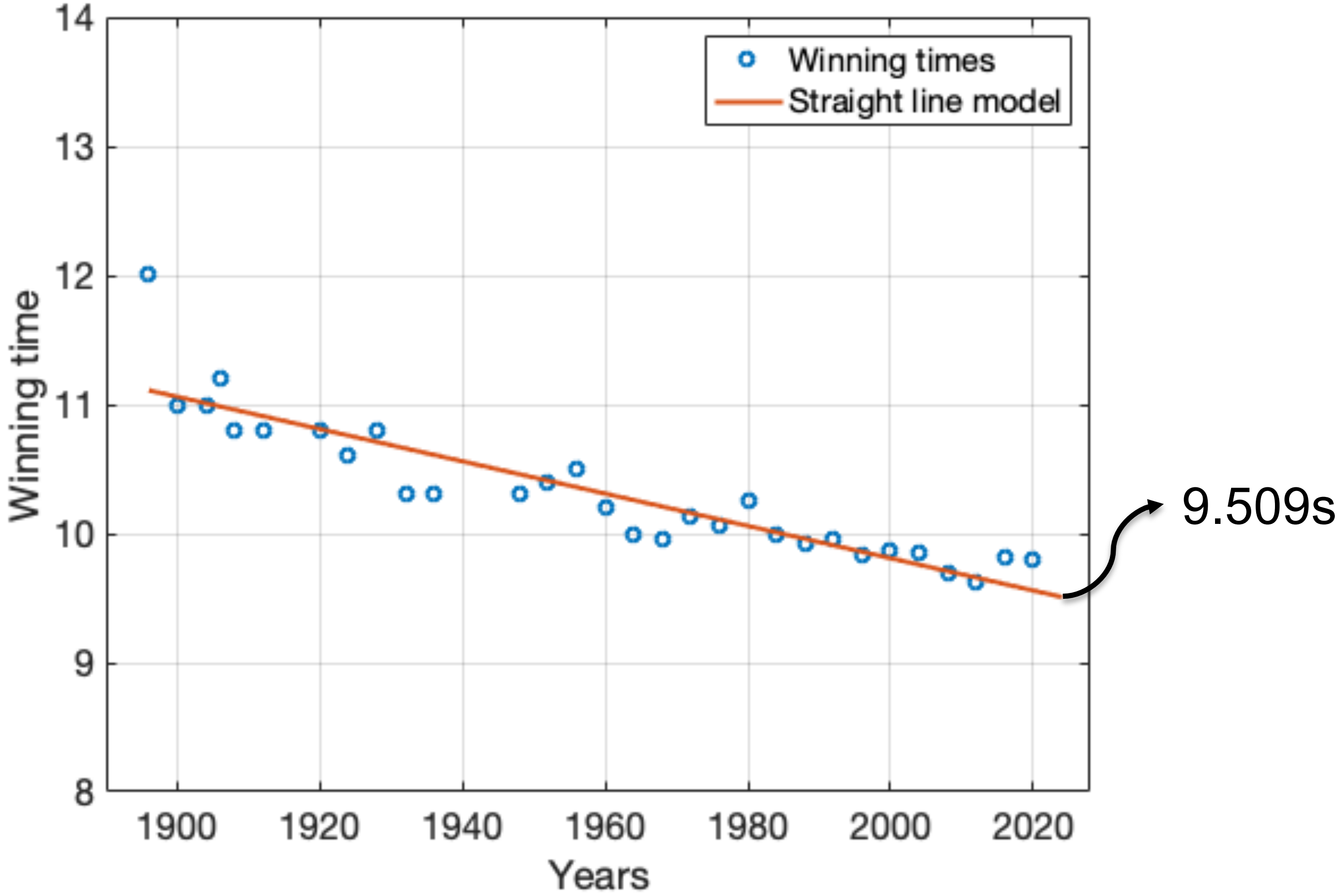
Estimate $\hat{\theta}_N = \Phi_N^{\dagger} Y_N = \left[\Phi_N^T \Phi_N\right]^{-1} \Phi_N^T Y_N$.

```
Matlab code
x=[…]'; % column vector
y=[…]'; % column vector
Phi_N=[x ones(length(x),1)];
Y_N=y;
Theta_hat=inv([Phi_N'*Phi_N]*Phi_N'*Y
Theta_hat=Phi_N\Y % recommended implementation
```

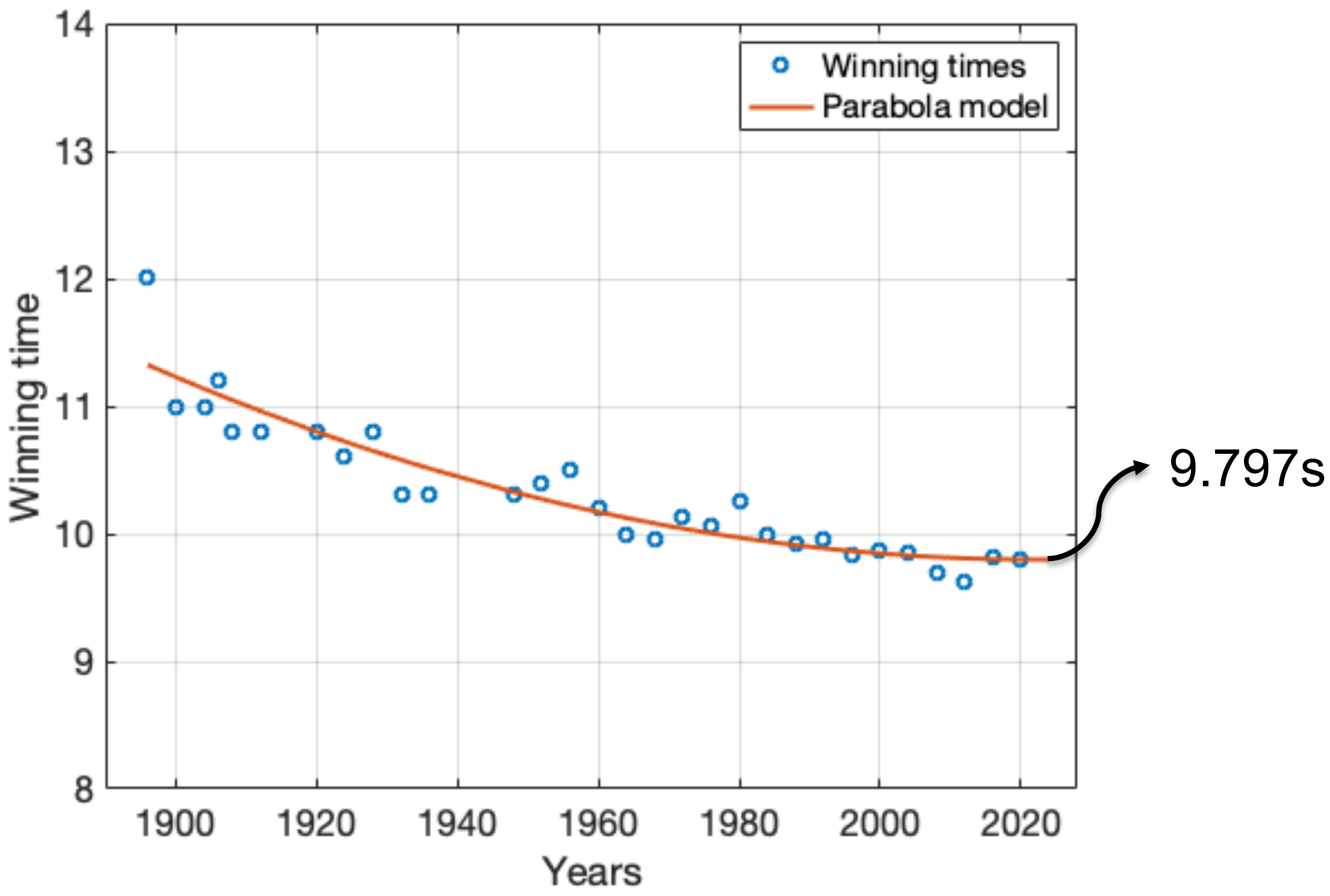
First case study - Linear trend model of the winning men's 100 m time at the Summer Olympics



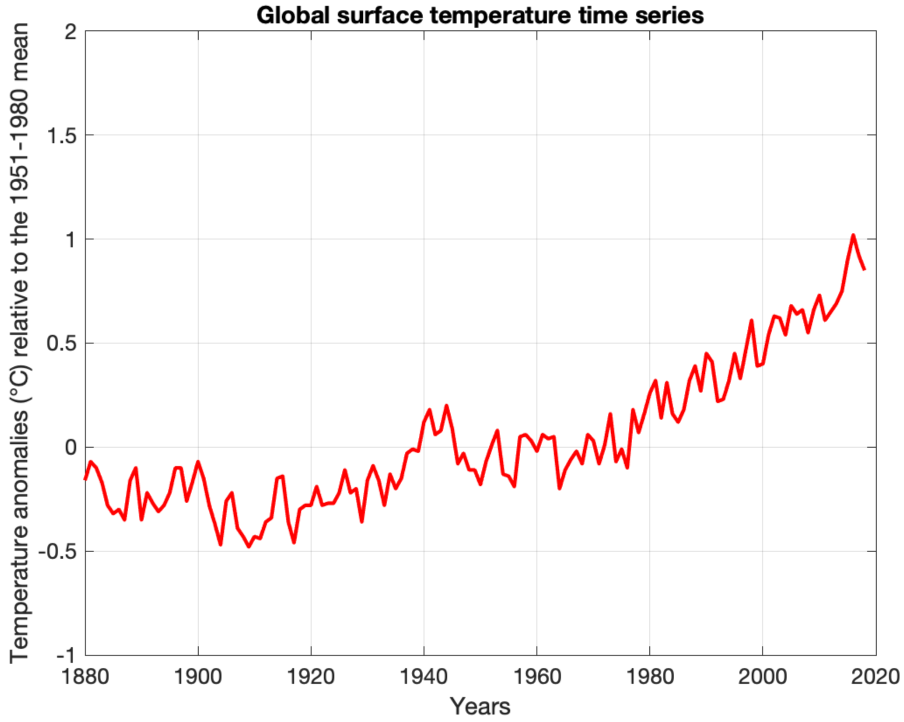
First case study - Linear trend model of the winning men's 100 m time at the Summer Olympics



First case study - Parabola trend model
of the winning men's 100 m time at the Summer Olympics



Second case study
Trend model for global surface temperature time series



Global warming seems to be clearly accelerating from 1980 onwards

- Global surface temperature time series shown against time is the temperature anomalies (in ° Celsius) relative to the 1951-1980 mean. The series is called GISTEMP after its producer, the NASA, New York, USA
- For more elaborated trend models, see paper by Manfred Mudelsee, *Trend analysis of climate time series: A review of methods*, Earth-Science Reviews 2019

Estimation of a linear trend model

$$V\left(\theta,Z^N\right)=\frac{1}{N}\sum_{k=1}^{N}\left(y(t_k)-(\alpha\times t_k+\beta)\right)^2 \qquad \theta=\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$$

- At the minimum of the criterion, its first derivative with respect to *q* is null:

$$\frac{\partial V(\theta,Z^N)}{\partial\alpha}=\frac{2}{N}\sum_{k=1}^{N}-t_k\left(y(t_k)-(\alpha\times t_k+\beta)\right)=0$$

$$\frac{\partial V(\theta,Z^N)}{\partial\beta}=\frac{2}{N}\sum_{k=1}^{N}-\left(y(t_k)-(\alpha\times t_k+\beta)\right)=0$$

$$\begin{bmatrix}\sum_{k=1}^{N}t_k^2 & \sum_{k=1}^{N}t_k\\ \sum_{k=1}^{N}t_k & \sum_{k=1}^{N}N\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix}=\begin{bmatrix}\sum_{k=1}^{N}t_k y(t_k)\\ \sum_{k=1}^{N}y(t_k)\end{bmatrix}$$

- The least squares estimates are given by :

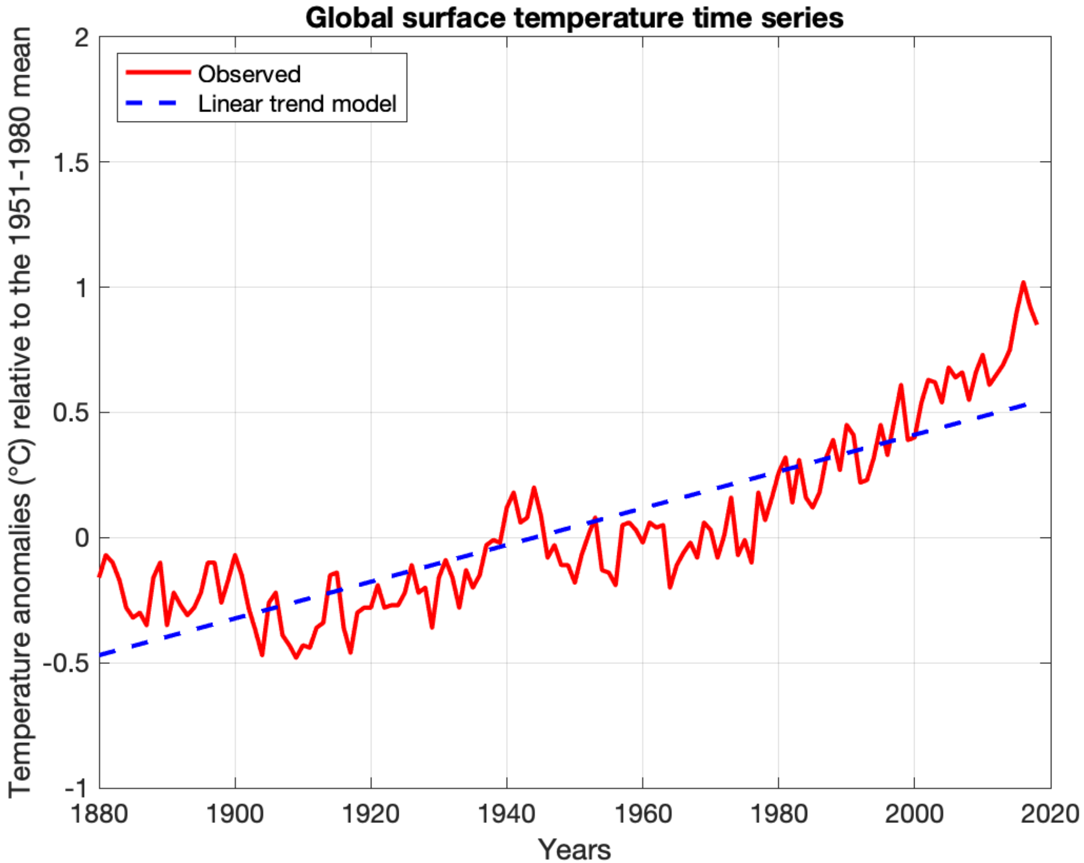
$$\begin{bmatrix}\hat{\alpha}\\ \hat{\beta}\end{bmatrix}=\begin{bmatrix}\sum_{k=1}^{N}t_k^2 & \sum_{k=1}^{N}t_k\\ \sum_{k=1}^{N}t_k & N\end{bmatrix}^{-1}\begin{bmatrix}\sum_{k=1}^{N}t_k y(t_k)\\ \sum_{k=1}^{N}y(t_k)\end{bmatrix}$$

Sous Matlab

```
theta_hat = inv([sum(years.^2) sum(years);
            sum(years) N])*...
            [sum(years.*T);sum(T)]
T_hat = theta(1)* years + theta(2);
plot(years,T,'o',years,T_hat)
```

Estimation of a linear trend model by LS applied to the global surface temperature

$$V\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, Z^N\right) = \frac{1}{N}\sum_{k=1}^{N}\Big(y(t_k) - (\alpha \times t_k + \beta)\Big)^2$$

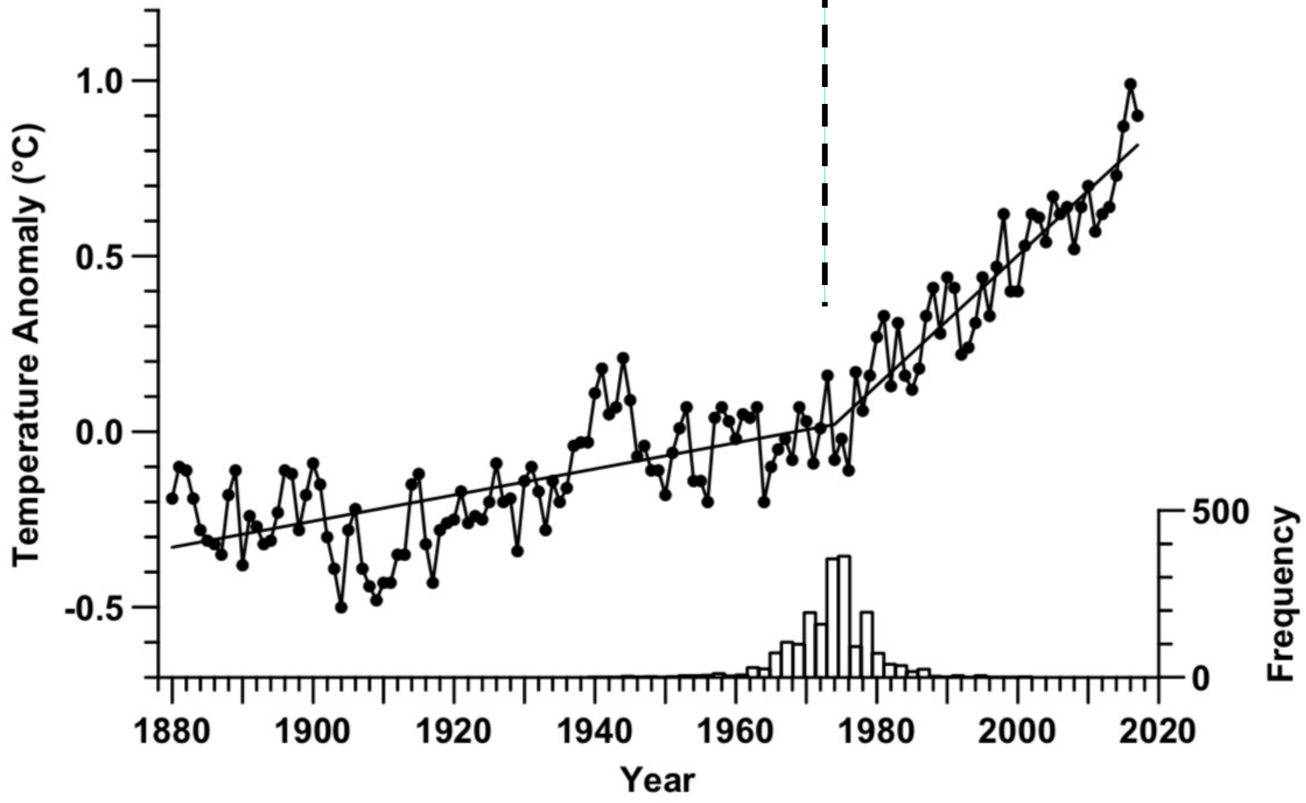


We can do better !

Estimation of piecewise linear trend models applied to the global surface temperature

$$V\left(\begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix}, Z^{N_1}\right)=\frac{1}{N}\sum_{k=1}^{N_1}\left(y(t_k)-(\alpha_1\times t_k+\beta_1)\right)^2$$

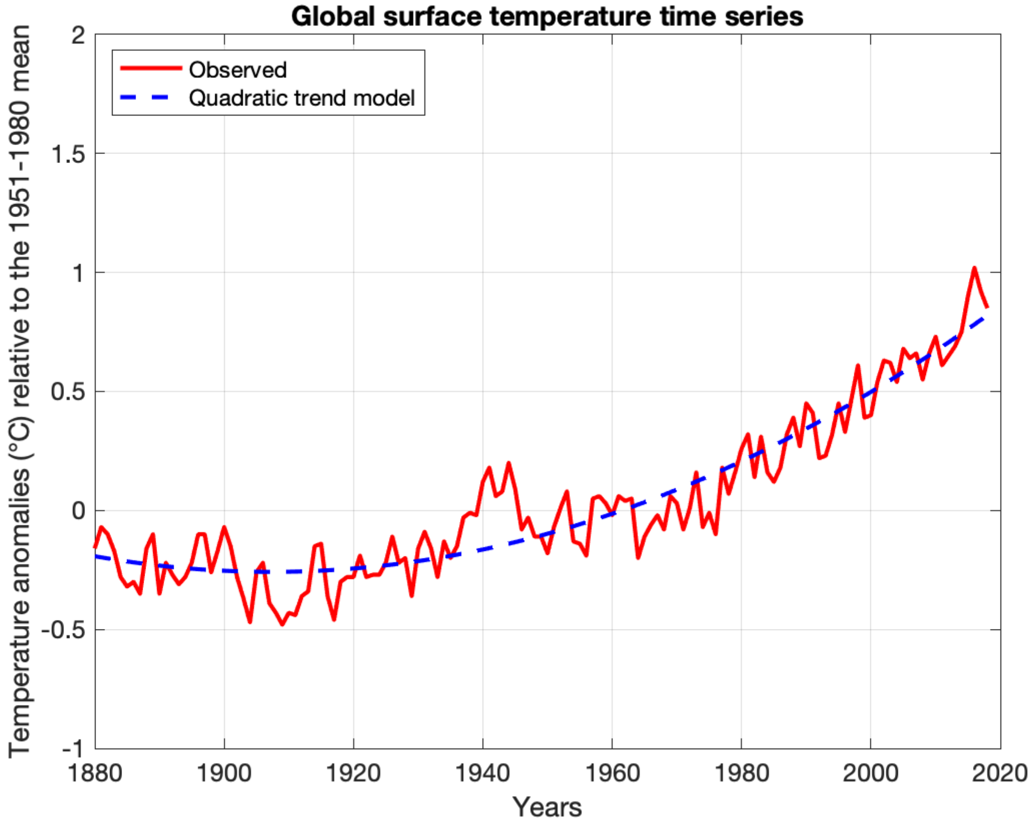
$$\left(\begin{bmatrix}\alpha_2\\ \beta_2\end{bmatrix}, Z^{N_2-N_1-1}\right)=\frac{1}{N}\sum_{k=N_1+1}^{N_2}\left(y(t_k)-(\alpha_2\times t_k+\beta_2)\right)^2$$



This is much better !

Estimation of a quadratic trend model applied to the global surface temperature

$$V\left(\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix}, Z^N\right) = \frac{1}{N}\sum_{k=1}^{N}\left(y(t_k) - (\alpha t_k^2 + \beta t_k + \gamma)\right)^2$$



This is also much better than the basic linear trend model !

UNIVERSITÉ DE LORRAINE

POLYTECH NANCY

Standard model accuracy measure: RMSE

- Given $\{y_1, \ldots, y_N\}$ actual observations of some data $\{y_t\}$, and let $\hat{y}_t$ be the simulated model value at time t

- We can calculate the *residuals* or *forecast errors*
$$E_N = Y_N - \hat{Y}_N = Y_N - \Phi_N \, \hat{\theta}$$

- A standard accuracy measure based on the residuals is the Root Mean Square Error (RMSE)
$$\text{RMSE}= \frac{1}{N} \left\| Y_N - \Phi_N \ \hat{\theta} \right\|_2$$

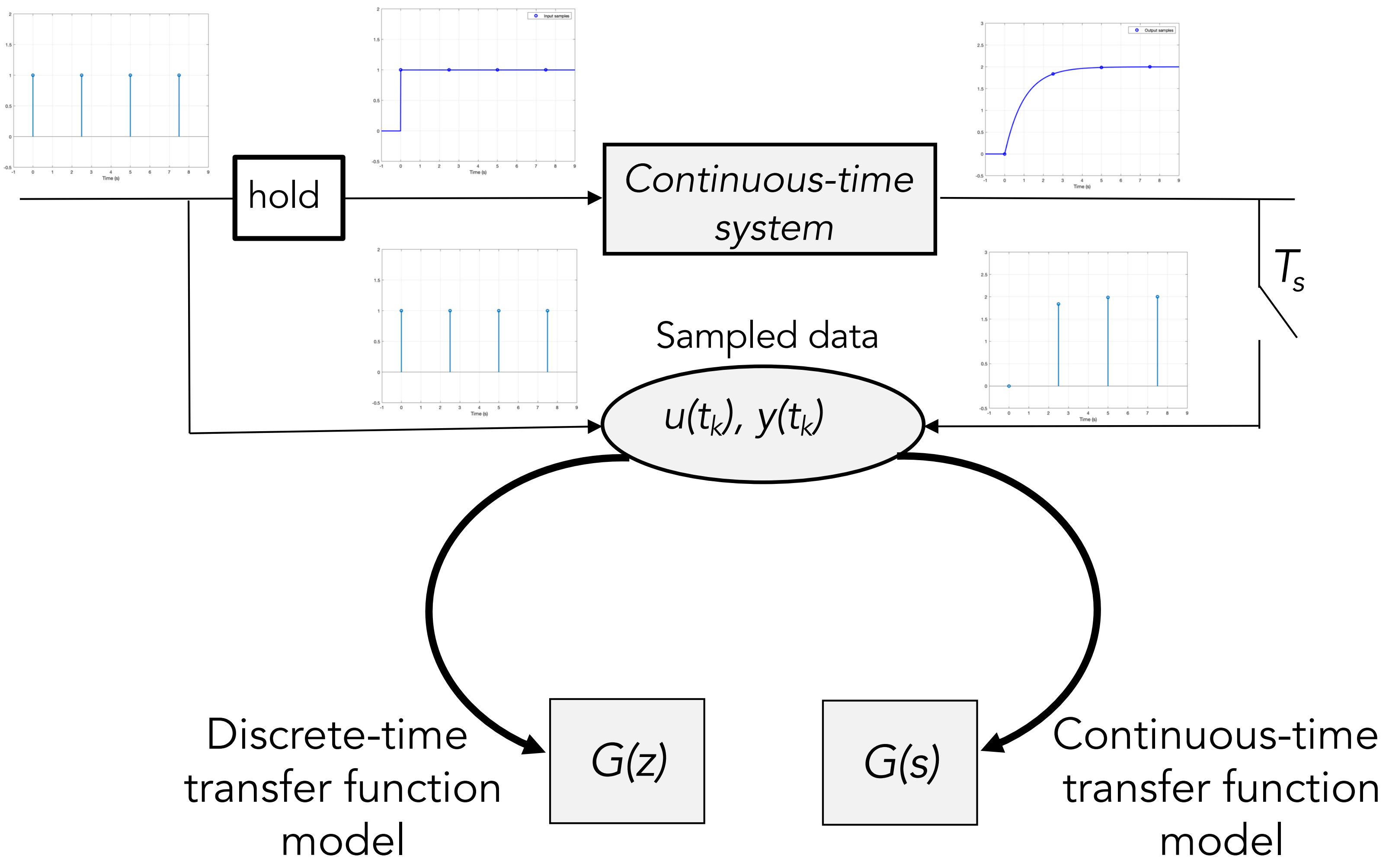
which calculates the Euclidean norm of the residuals, *i.e.*, the square root of the sum of the squares of all the residuals

Review of linear regression and least squares estimation

1. Least squares-based model estimation for static systems

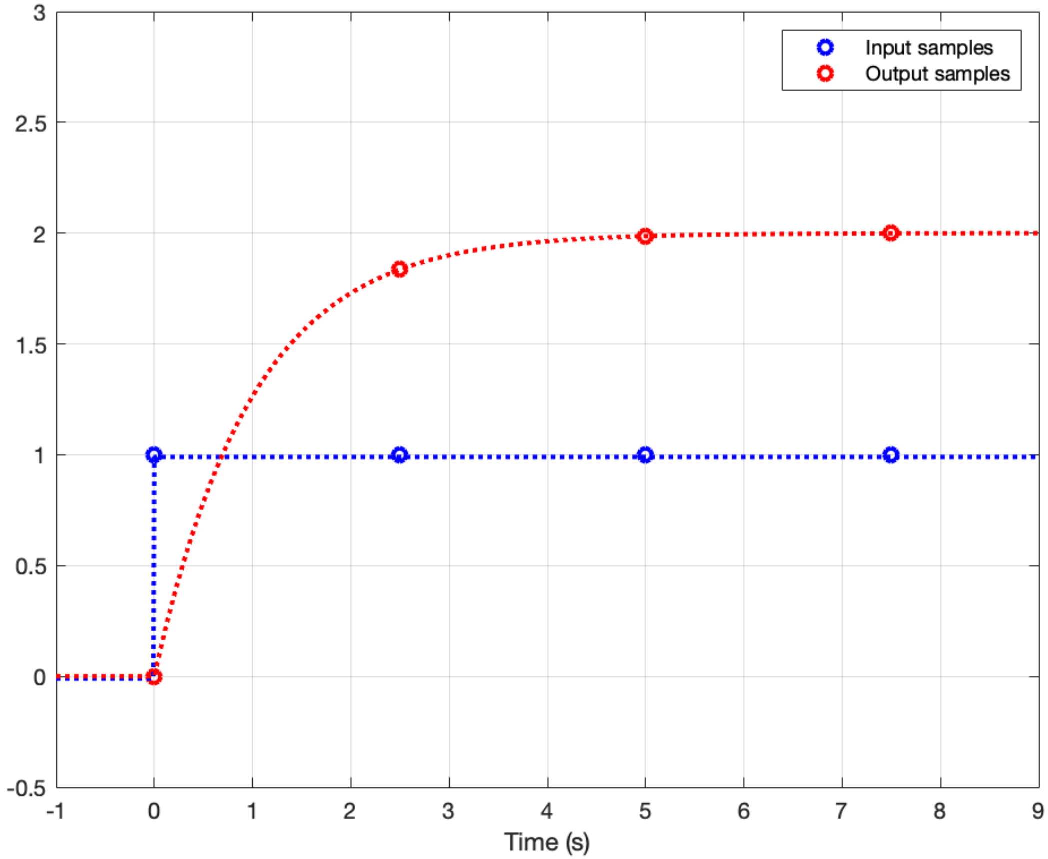
2. Least squares-based model estimation for dynamical systems

Transfer function model learning of a dynamic system by using basic linear regression



Transfer function model learning of a dynamic system by using basic linear regression

Goal: determine a continuous-time or discrete-time transfer function model of the dynamic system from step response data by using basic linear regression



Continuous-time model learning of a dynamic system by using basic linear regression

- Laplace transfer function model choice:

$$\boxed{G(s) = \frac{Y(s)}{U(s)} = \frac{b}{s+a}}$$

- We seek to estimate the values of *a* and *b* that best fit the step response data by using basic linear regression (least squares)

- In the time-domain, the continuous-time model takes the form of a differential equation

$$(s+a)Y(s) = bU(s)$$

$$sY(s) + aY(s) = bU(s)$$

$$\boxed{\dot{y}(t) + ay(t) = bu(t)}$$

Output time-derivative

Continuous-time model learning of a dynamic system by using basic linear regression

- At time-instant t_k

$$\dot{y}(t_k) + ay(t_k) = bu(t_k)$$

or

$$\dot{y}(t_k) = -ay(t_k) + bu(t_k)$$

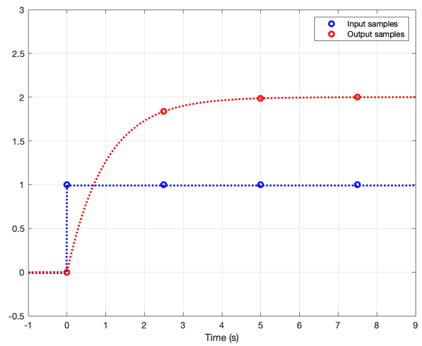
- From the $N=4$ sampled measurements, we can write a set of 4 equations

$$\dot{y}(t_0) = -ay(t_0) + bu(t_0)$$

$$\dot{y}(t_1) = -ay(t_1) + bu(t_1)$$

$$\dot{y}(t_2) = -ay(t_2) + bu(t_2)$$

$$\dot{y}(t_3) = -ay(t_3) + bu(t_3)$$



Continuous-time model learning of a dynamic system by using basic linear regression

- The 4 equations can be written in matrix form

$$\begin{bmatrix} \dot{y}(t_0) \\ \dot{y}(t_1) \\ \dot{y}(t_2) \\ \dot{y}(t_3) \end{bmatrix} = \begin{bmatrix} -y(t_0) & u(t_0) \\ -y(t_1) & u(t_1) \\ -y(t_2) & u(t_2) \\ -y(t_3) & u(t_3) \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$Y \quad = \quad \Phi \quad \theta$$

$$\hat{\theta} = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = [\Phi^T \Phi]^{-1} \Phi^T Y$$

Continuous-time model learning of a dynamic system by using basic linear regression

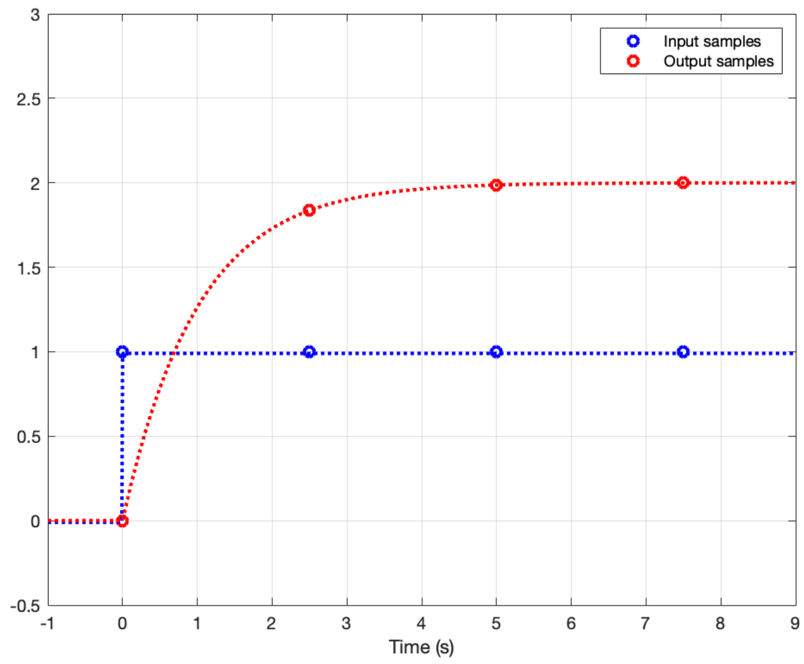
- The 4 equations can be written in matrix form

$$\begin{bmatrix} 2 \\ 0.1642 \\ 0.0135 \\ 0.0011 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1.8358 & 1 \\ -1.9865 & 1 \\ -1.9989 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

$$Y \quad = \quad \Phi \quad \theta$$

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T Y$$

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \Rightarrow \hat{G}(s) = \frac{2}{s+1}$$



H. Garnier

Discrete-time model learning of a dynamic system by using basic linear regression

- Zero-order hold equivalent of the Laplace transfer function model choice:

$$G(s) = \frac{b}{s+a} \qquad \boxed{G(z) = \frac{Y(z)}{U(z)} = (1 - z^{-1})Z\left(\frac{G(s)}{s}\right) = \frac{b_1 z^{-1}}{1+a_1 z^{-1}}}$$

The parameters a_1 and b_1 depend on T_s

$$\begin{cases} a_1 = -e^{-aT_s} = -0.0821 \\ b_1 = \dfrac{b}{a}(1 + a_1) = 1.8358 \end{cases}$$

- We seek to estimate the values of a_1 and b_1 that best fit the step response data by using basic linear regression (least squares)

- In the time-domain, the discrete-time model takes the form of a difference equation

$$(1 + a_1 z^{-1})Y(z) = b_1 z^{-1} U(z)$$

$$Y(z) + a_1 z^{-1} Y(z) = b_1 z^{-1} U(z)$$

$$\boxed{y(t_k) + a_1 y(t_{k-1}) = b_1 u(t_{k-1})}$$

H. Garnier

Discrete-time model learning of a dynamic system by using basic linear regression

- At time-instant t_k

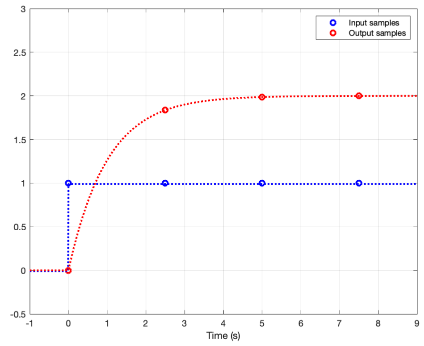
$$y(t_k) + a_1 y(t_{k-1}) = b_1 u(t_{k-1})$$

or

$$y(t_k) = -a_1 y(t_{k-1}) + b_1 u(t_{k-1})$$

- From the $N=4$ sampled measurements, we can write a set of 3 equations *only* because of the time-shift in the difference equation

$$y(t_1) = -a_1 y(t_0) + b_1 u(t_0)$$
$$y(t_2) = -a_1 y(t_1) + b_1 u(t_1)$$
$$y(t_3) = -a_1 y(t_2) + b_1 u(t_2)$$



Discrete-time model learning of a dynamic system by using basic linear regression

- The 3 equations can be written in matrix form

$$\begin{bmatrix} y(t_1) \\ y(t_2) \\ y(t_3) \end{bmatrix} = \begin{bmatrix} -y(t_0) & u(t_0) \\ -y(t_1) & u(t_1) \\ -y(t_2) & u(t_2) \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix}$$

$$Y \quad = \quad \Phi \quad \theta$$

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T Y$$

Discrete-time model learning of a dynamic system by using basic linear regression

- The 3 equations can be written in matrix form

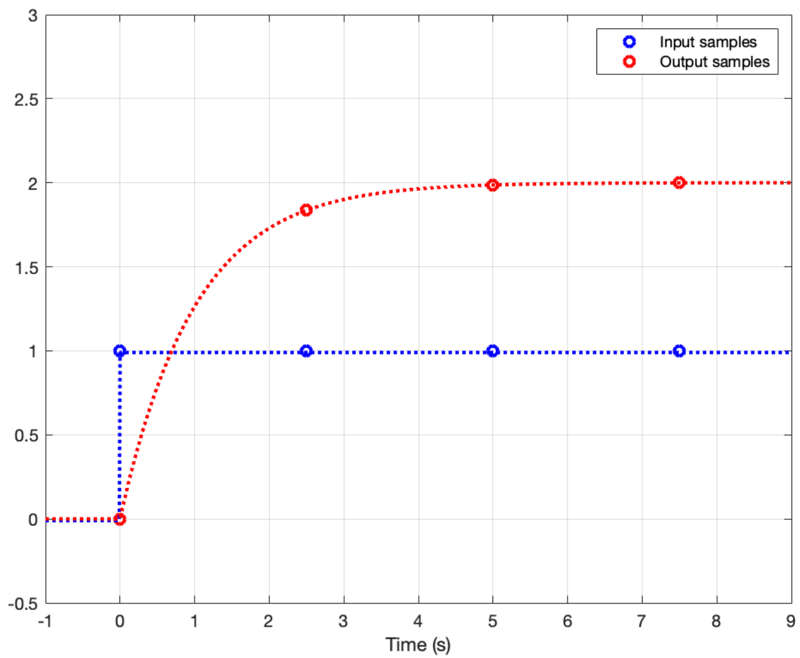
$$\begin{bmatrix}1.8358\\1.9865\\1.9989\end{bmatrix}=\begin{bmatrix}0 & 1\\-1.8358 & 1\\-1.9865 & 1\end{bmatrix}\begin{bmatrix}a_1\\b_1\end{bmatrix}$$

$$Y \quad = \quad \Phi \quad \theta$$

$$\hat{\theta} = [\Phi^T\Phi]^{-1}\Phi^T Y$$

$$\begin{bmatrix}\hat{a}_1\\\hat{b}_1\end{bmatrix}=\begin{bmatrix}-0.0821\\1.8358\end{bmatrix}$$

$$\Longrightarrow \hat{G}(z)=\frac{\hat{b}_1 z^{-1}}{1+\hat{a}_1 z^{-1}}=\frac{1.8358z^{-1}}{1-0.0821z^{-1}}$$



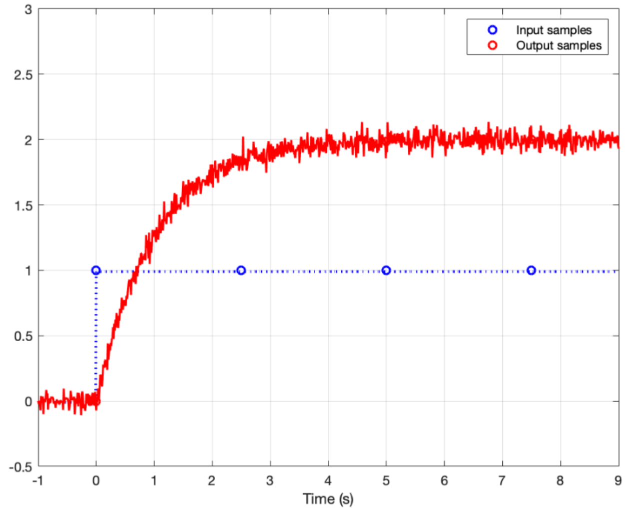
Use of simple linear regression for transfer function model learning - Take-home message

- In the ideal noise-free measurement case, it works fine
  - the continuous or discrete-time transfer function model parameter can be estimated by linear regression (least squares)

- In practice, the simple LS method breaks down for two main reasons
  - The output measurement is not perfectly known. It is contaminated by noise

    $\Longrightarrow$ Incorrect LS estimates whatever the continuous or discrete-time model form

  - The input and output time-derivatives required in the continuous-time model form are usually not measured



Advantages of continuous-time models

✓ **CT models** have certain advantages in relation to their equivalent DT models

- Are more intuitive to control engineers in their every-day practice
- Many practical control design are still based on CT models
- CT models are often preferred for fault detection
  - reveal faults more directly than their DT counterparts
- Parameter values are independent of T_s

$$G_o(p) = \frac{1}{p^2+p+1} \xrightarrow{\text{ZOH}} \begin{cases} T_s = 0.1\text{s}; & G_{T_s}(q^{-1}) = \dfrac{0.0048q^{-1}+0.0047q^{-2}}{1-1.8953q^{-1}+0.9048q^{-2}} \\ T_s = 1\text{s}; & G_{T_s}(q^{-1}) = \dfrac{0.3403q^{-1}+0.2417q^{-2}}{1-0.7849q^{-1}+0.3679q^{-2}} \end{cases}$$

H. Garnier

Continuous-time methods presents some advantages

- ✓ ***CT methods*** present many advantages in relation to their equivalent DT methods
  - include ***inherent data prefiltering***
  - are well-suited to ***fast sampling*** situations
  - are well-adapted to identify ***stiff*** systems
  - can cope easily with ***irregularly*** sampled data

- ✓ In the following of the course, we will focus on continuous-time model learning