



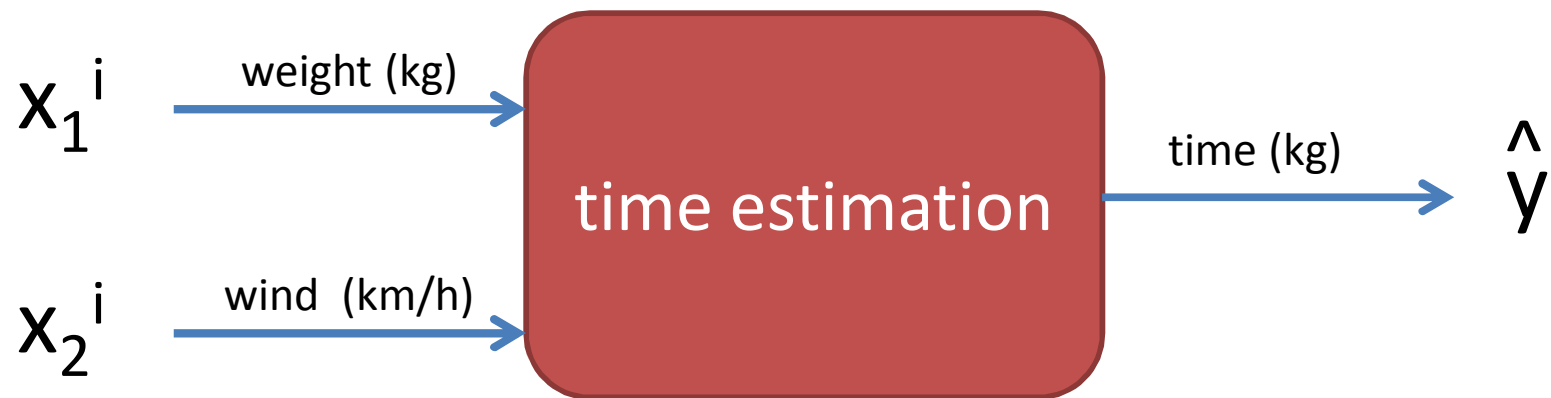
Predictive modelling with linear regression

Tunç Ali Kütükçüoğlu

CFA, Dipl. Ing. ETH

Email: tuncalik@finaquant.com

Linear regression model with two input parameters



Estimated time record (seconds) of an athlete running a 100-meter distance against the wind (speed in km/h) with a given weight (kg) attached to his belt. Estimation for the i 'th observation.

Historical data and estimation

i: observation	x1: weight (kg)	x2: wind (km/h)	y: time
1	1.0	20	15.5
2	1.5	25	18.0
3	1.0	40	17.2

Data from the first observation:

$$x_1^1 = 1.0 \text{ kg}, \quad x_2^1 = 20 \text{ km/h}, \quad y^1 = 15.5 \text{ sec}$$

Estimation \hat{y}^i as linear combination of two input parameters x_1^i and x_2^i :

$$\hat{y}^i = \beta_0 + \beta_1 \cdot x_1^i + \beta_2 \cdot x_2^i$$

Estimation error and SSE

Estimation error for i'th observation:

$$e^i = y^i - \hat{y}^i = y^i - (\beta_0 + \beta_1 \cdot x_1^i + \beta_2 \cdot x_2^i)$$

Sum of Square Errors (SSE):

$$SSE = \sum_{i=1}^N (e^i)^2$$

Training and Test errors, MSE: Mean Square Error

$$SSE_{train} = \sum_{i=1}^N (e^i)^2, \quad MSE_{train} = SSE_{train} \div N$$

$$SSE_{test} = \sum_{i=1}^M (e^i)^2, \quad MSE_{test} = SSE_{test} \div M$$

Historical data with matrix notation

Linear filter with two input parameters and N observations

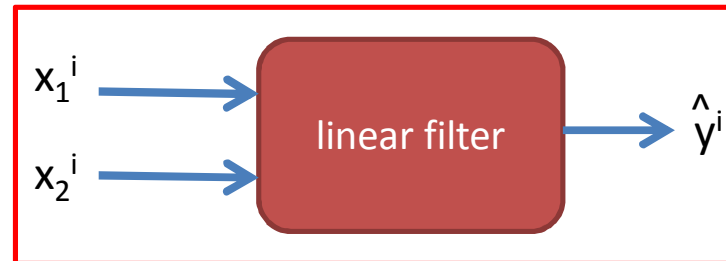
$$\text{input matrix } X = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1^1 & x_1^2 & \dots & x_1^N \\ x_2^1 & x_2^1 & \dots & x_2^N \end{bmatrix}$$

$$\text{output vector } Y = [y^1 \quad y^2 \quad \dots \quad y^N]^T$$

$$\text{estimation vector } \hat{Y} = [\hat{y}^1 \quad \hat{y}^2 \quad \dots \quad \hat{y}^N]^T$$

$$\text{coefficient vector } \beta = [\beta_0 \quad \beta_1 \quad \beta_2]^T$$

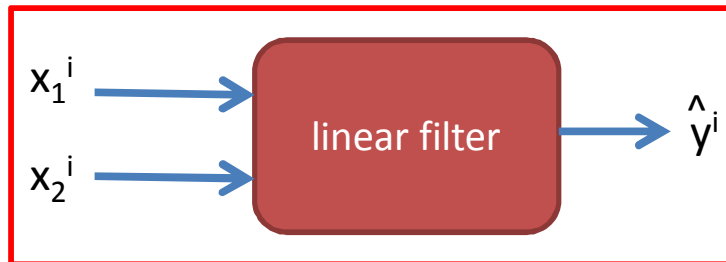
Calculating estimations with matrices



$$\hat{Y} = \begin{bmatrix} \hat{y}^1 \\ \hat{y}^2 \\ \dots \\ \hat{y}^N \end{bmatrix} = X^T \times \beta = \begin{bmatrix} 1 & x_1^1 & x_2^1 \\ 1 & x_1^2 & x_2^2 \\ \dots & \dots & \dots \\ 1 & x_1^N & x_2^N \end{bmatrix} \times \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 \cdot x_1^1 + \beta_2 \cdot x_2^1 \\ \beta_0 + \beta_1 \cdot x_1^2 + \beta_2 \cdot x_2^2 \\ \dots \\ \beta_0 + \beta_1 \cdot x_1^N + \beta_2 \cdot x_2^N \end{bmatrix}$$

Calculating estimation errors with matrices

Linear filter with two input parameters and N observations



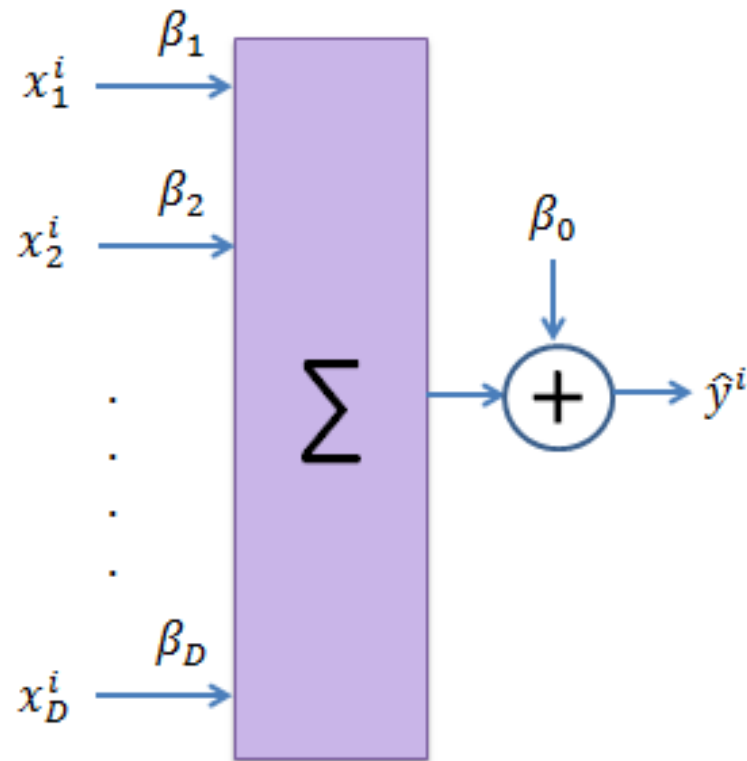
Estimation error E:

$$E = Y - \hat{Y} = \begin{bmatrix} y^1 - \hat{y}^1 \\ y^2 - \hat{y}^2 \\ \vdots \\ y^N - \hat{y}^N \end{bmatrix} = \begin{bmatrix} e^1 \\ e^2 \\ \vdots \\ e^N \end{bmatrix} \quad \square$$

Sum of Square Errors:

$$SSE = \sum_{i=1}^N (e^i)^2 = E^T \times E$$

General diagram of a linear filter with D input parameters



$$\hat{y}^i = \beta_0 + \sum_{d=1}^D \beta_d \cdot x_d^i$$
$$\hat{Y} = X^T \times \beta$$

Calculating the optimal coefficient vector β

Step 1: Formulate Sum of Square Errors (SSE) as a function of β

$$(1) SSE(\beta) = \sum_{i=1}^N (y^i - \hat{y}^i)^2 = \sum_{i=1}^N \{y^i - [(\bar{x}^i)^T \times \beta]\}^2$$

$$(2) SSE(\beta) = (Y - \hat{Y})^T \times (Y - \hat{Y}) = (Y - X^T \times \beta)^T \times (Y - X^T \times \beta)$$

Step 2: Set derivative to zero and solve the equation for the optimal coefficient vector β

$$\frac{\partial SSE(\beta)}{\partial \beta} = 2 \cdot X \times (X - X^T \times \beta) = 0$$

$$\Rightarrow \beta_{opt} = (X \times X^T)^{-1} \times X \times Y$$

Calculating optimal β with training data

Step 1: Calculate the optimal coefficient vector β_{opt} with training data

$$\beta_{opt} = (X_{train} \times X_{train}^T)^{-1} \times X_{train} \times Y$$

Step 2: Calculate estimation error with training data (training error)

$$E_{train} = Y_{train} - \hat{Y}_{train} = Y_{train} - X_{train}^T \times \beta_{opt}$$

$$SSE_{train} = E_{train}^T \times E_{train}$$

$$MSE_{train} = SSE_{train} \div N$$

Step 3: Calculate estimation error with test data (test error)

$$E_{test} = Y_{test} - \hat{Y}_{test} = Y_{test} - X_{test}^T \times \beta_{opt}$$

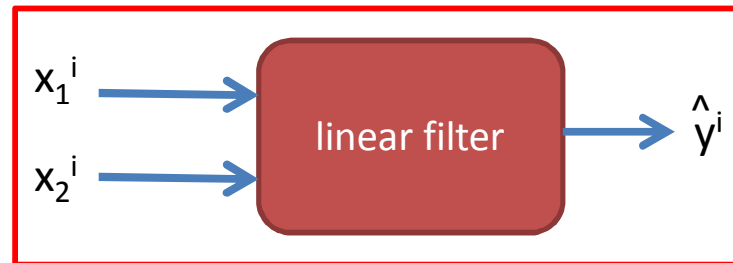
$$SSE_{test} = E_{test}^T \times E_{test}$$

$$MSE_{test} = SSE_{test} \div M$$



LR Estimation Model in Action

Athlete running against the wind with a weight at his belt



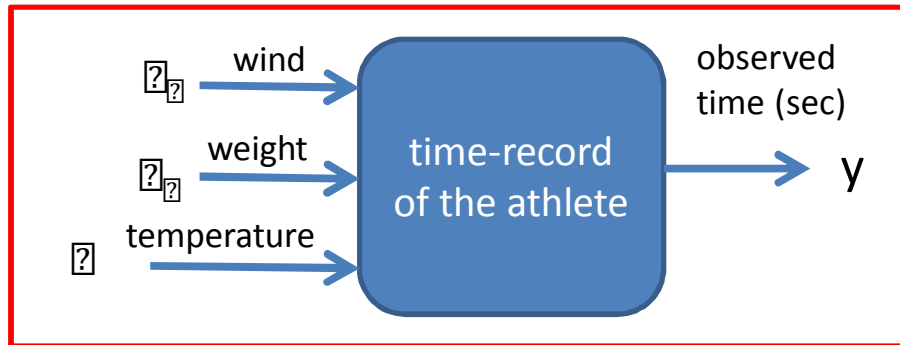
Related article @ finaquant.com:

Predictive Modelling with Linear Regression - 2

presented by:

Tunç Ali Kütükçüoğlu

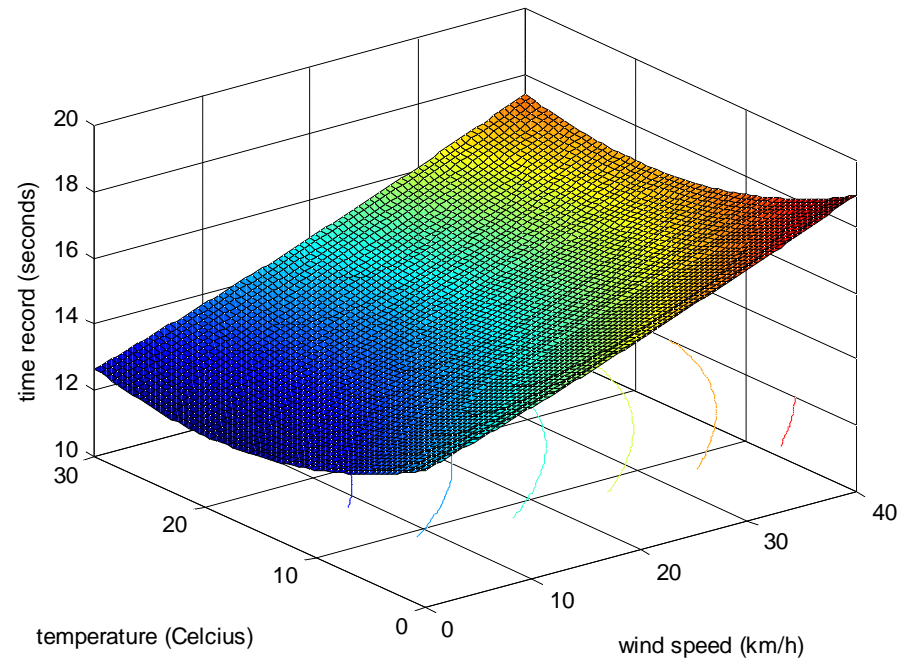
The Physical Function



$$\hat{y} = 12 + 0.1 \cdot \hat{x}_1 + 0.0005 \cdot \hat{x}_1^2 + 0.15 \cdot \hat{x}_2 + 0.003 \cdot \hat{x}_2^2 + 0.005 \cdot \hat{x}_3(\hat{x}_3 - 20)^2$$

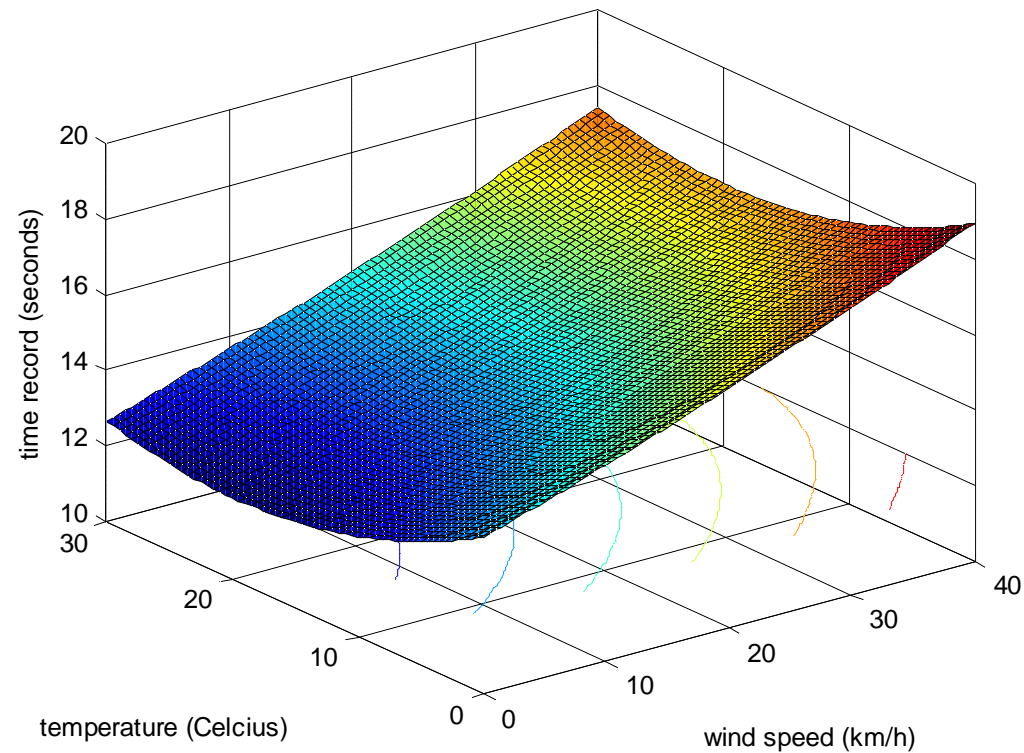
Physical Function: Time-record as a function of wind and temperature

The third parameter *temperature* (T) which is not captured by the LR estimation model represents generally all the unknown factors that add to the degree of uncertainty and estimation error in the time record of the athlete.



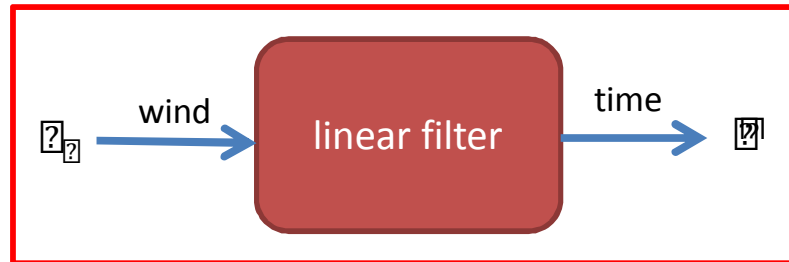
The Physical Function

Physical Function: Time-record as a function of wind and temperature

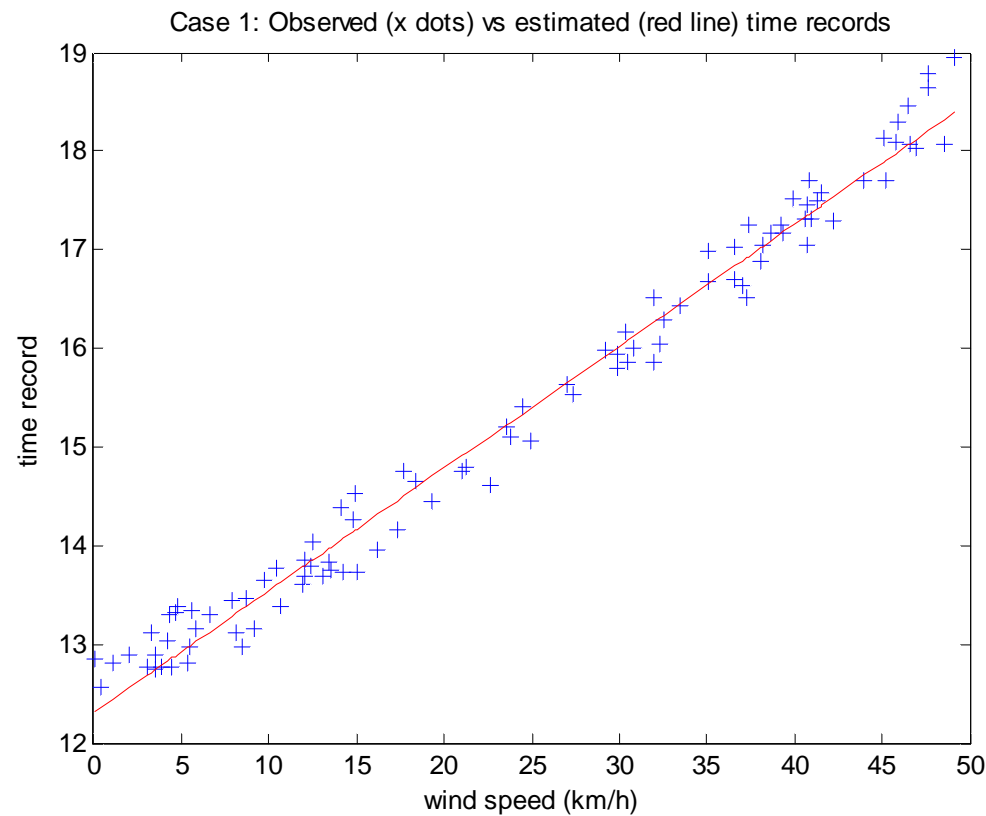


$$t = 12 + 0.1 \cdot x_1 + 0.0005 \cdot x_1^2 + 0.15 \cdot x_2 + 0.003 \cdot x_2^2 + 0.005 \cdot \text{abs}(T - 20)^2$$

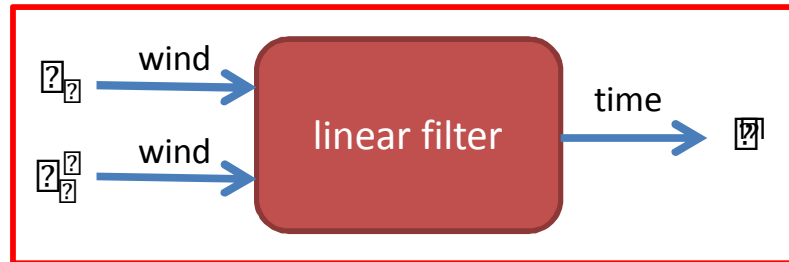
Case 1: Single parameter time estimation



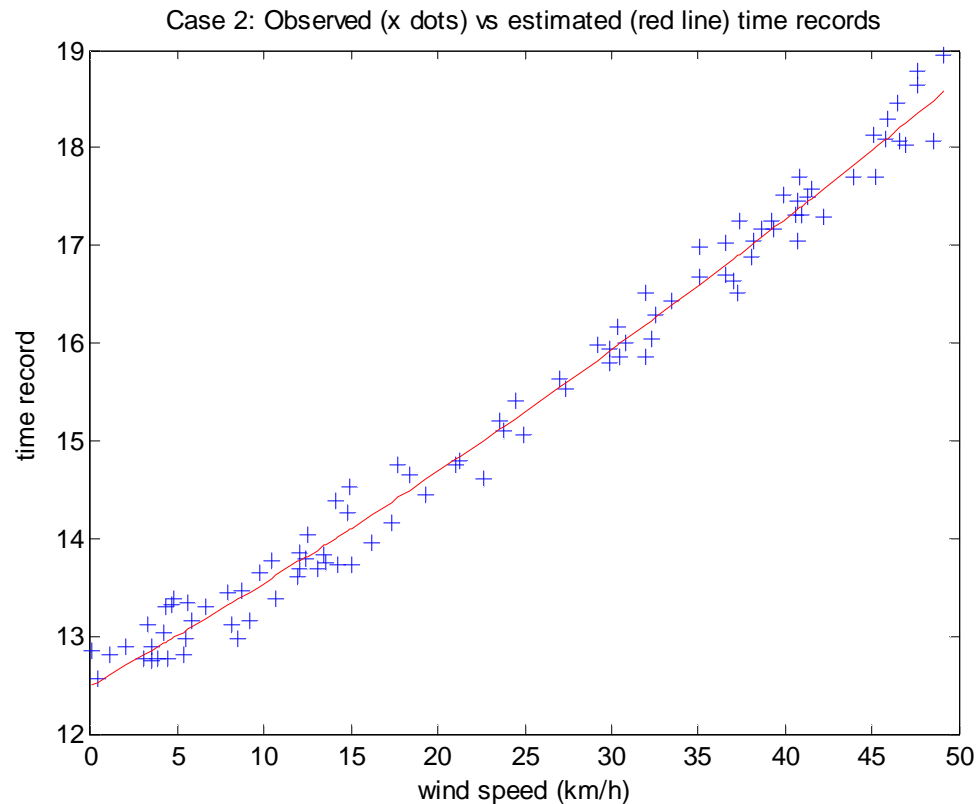
$$\hat{y} = \beta_0 + \beta_1 \cdot x_1$$



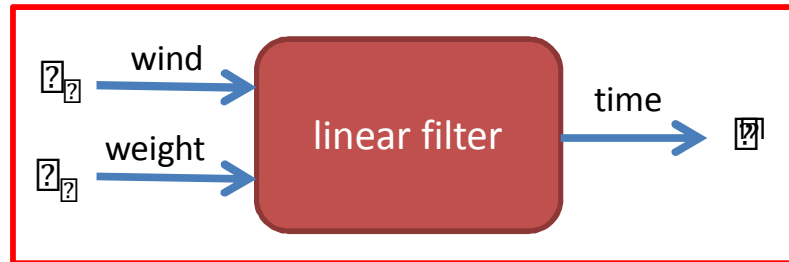
Case 2: Single-parameter 2nd degree polynomial time estimation



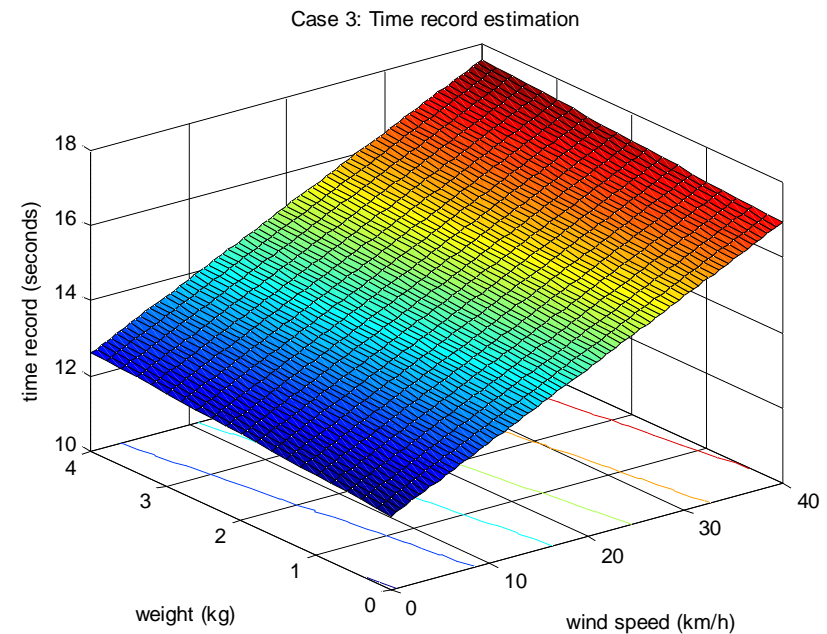
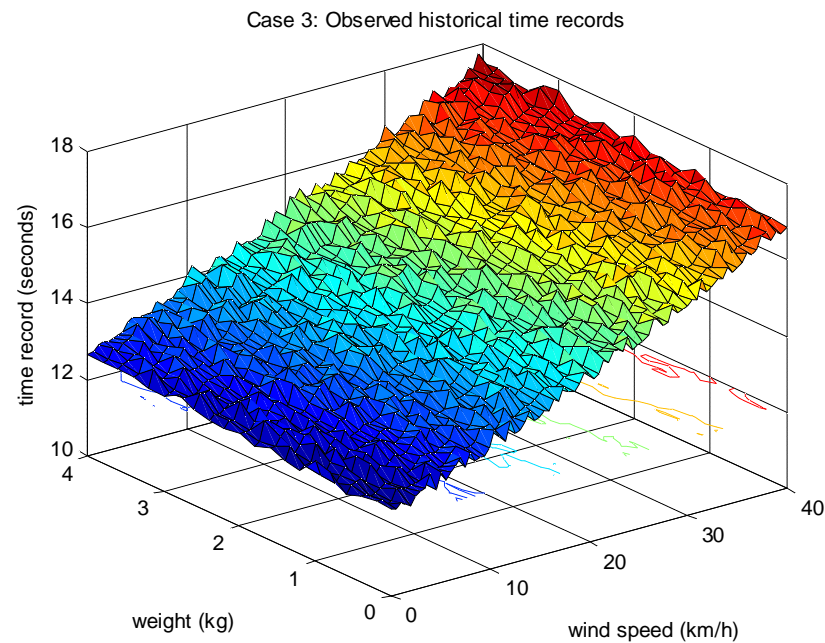
$$\hat{y} = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_1^2$$



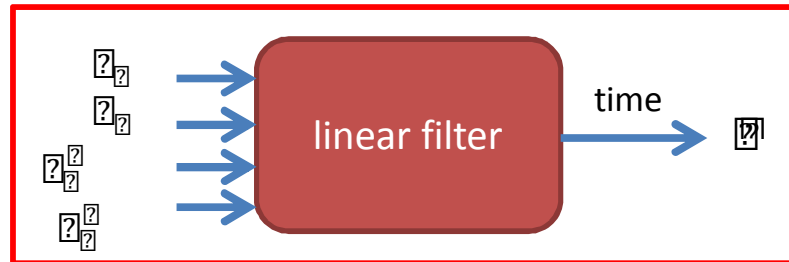
Case 3: Two-parameter time estimation



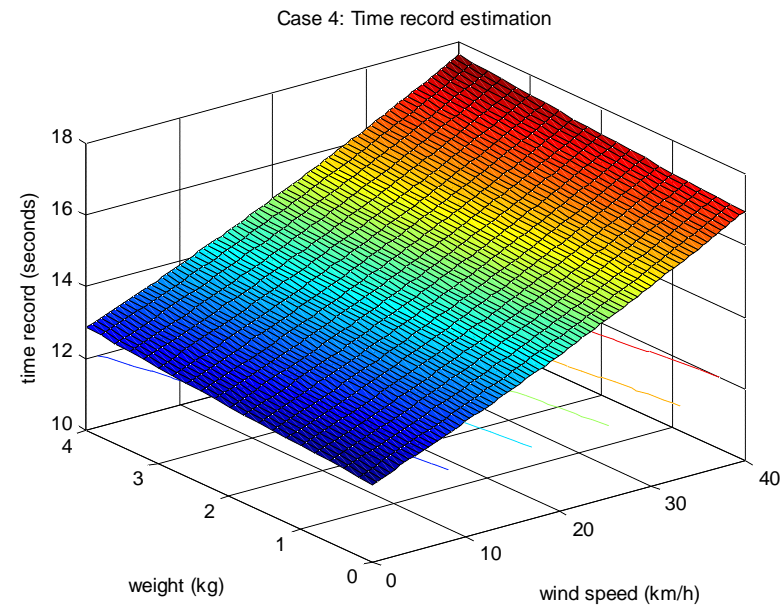
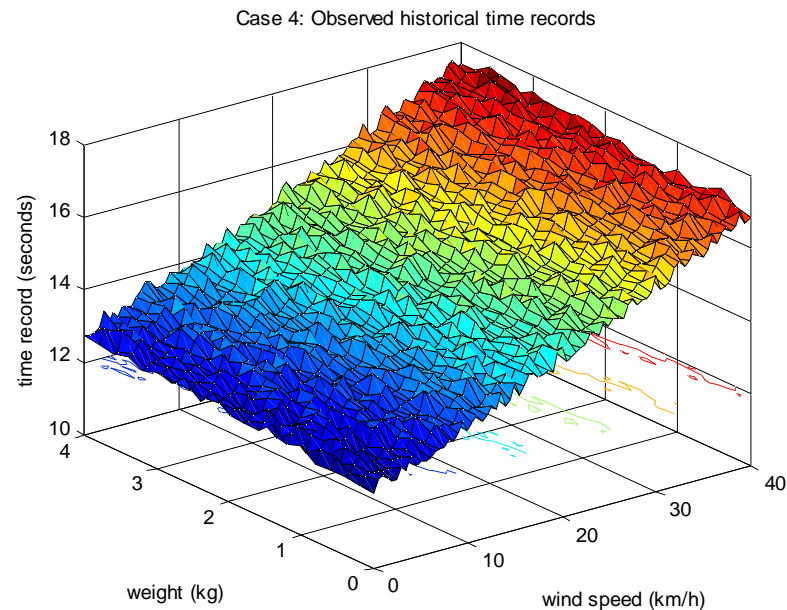
$$\hat{y} = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2$$



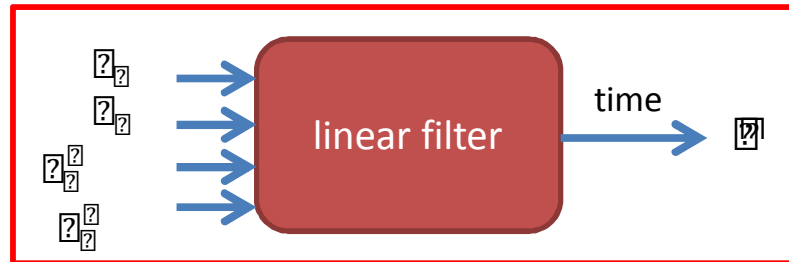
Case 4: Two-parameter polynomial time estimation



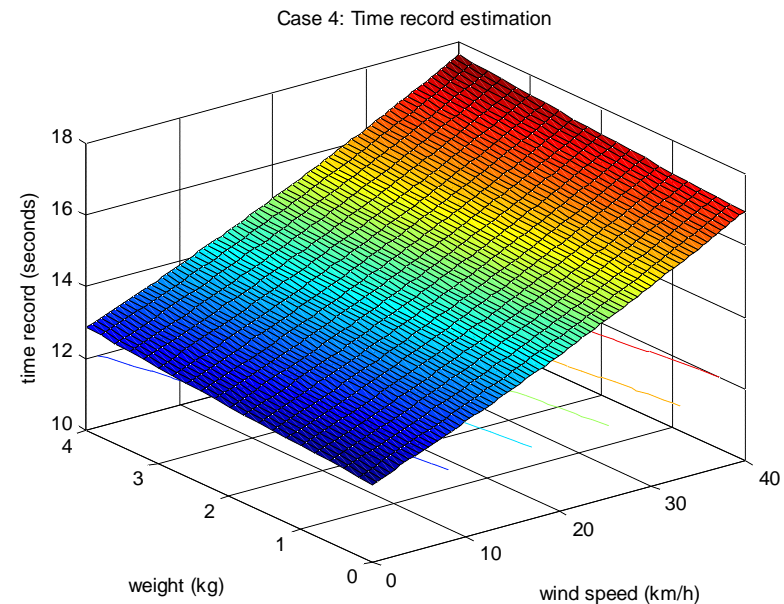
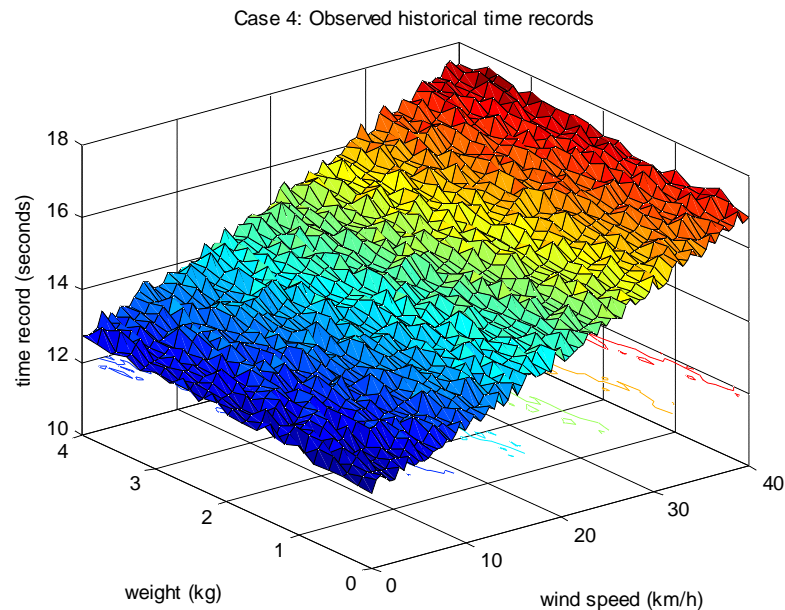
$$\hat{y} = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \beta_3 \cdot x_1^2 + \beta_4 \cdot x_2^2$$



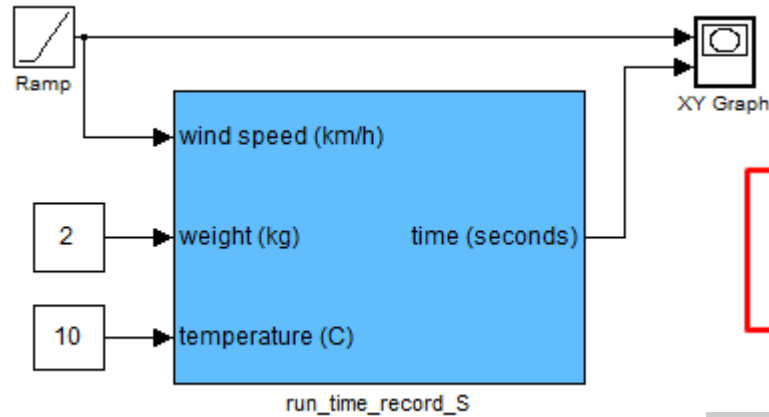
Case 4: Two-parameter polynomial time estimation



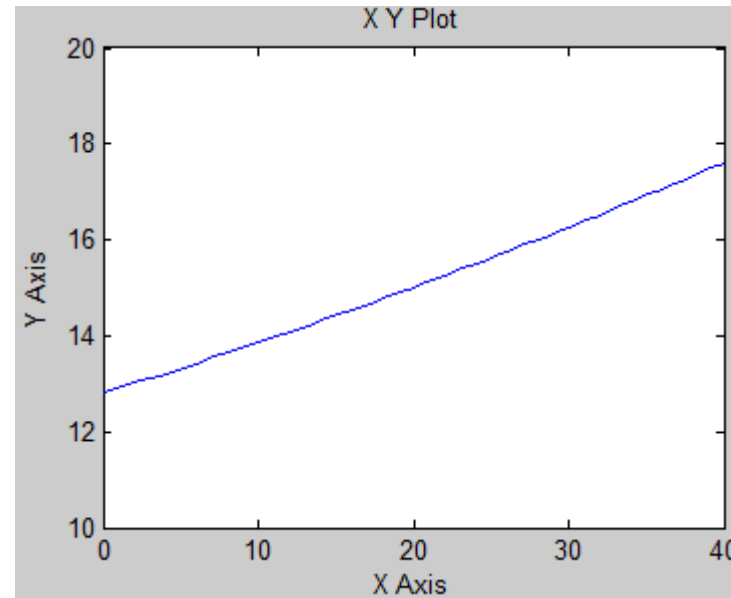
$$\hat{y} = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \beta_3 \cdot x_1^2 + \beta_4 \cdot x_2^2$$



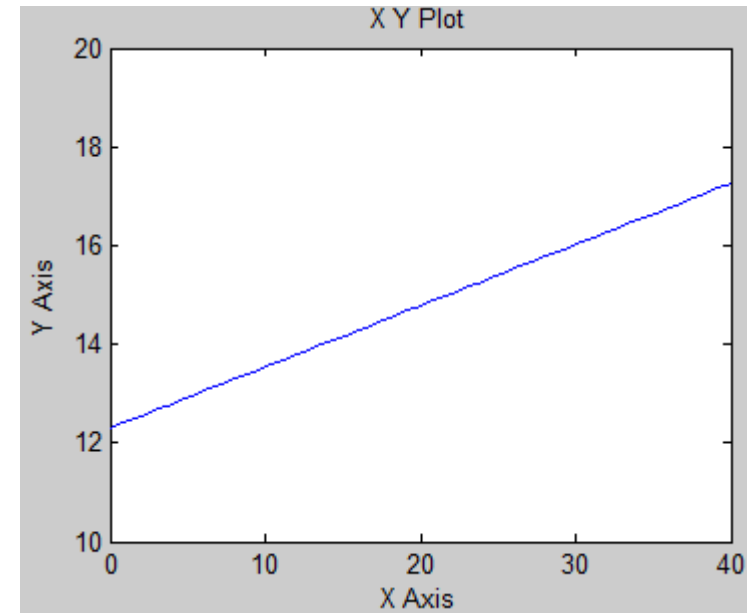
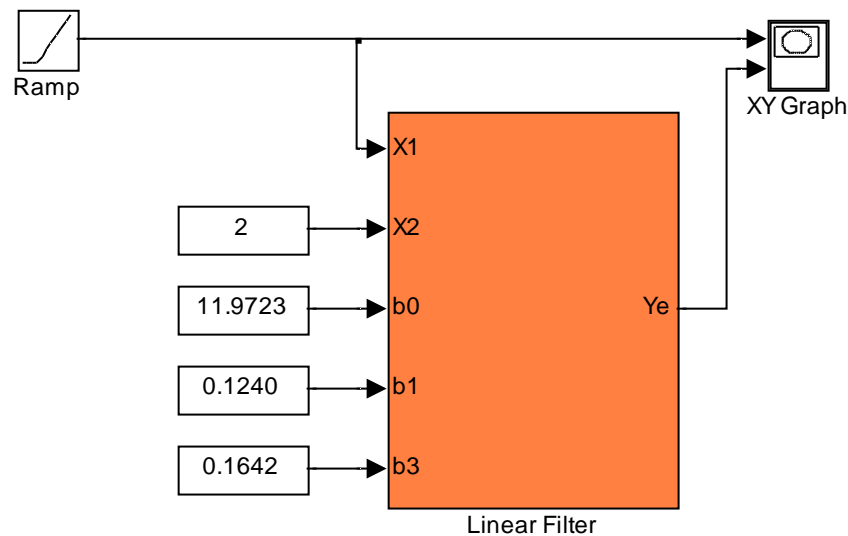
Physical function to generate historical data (matlab-Simulink)



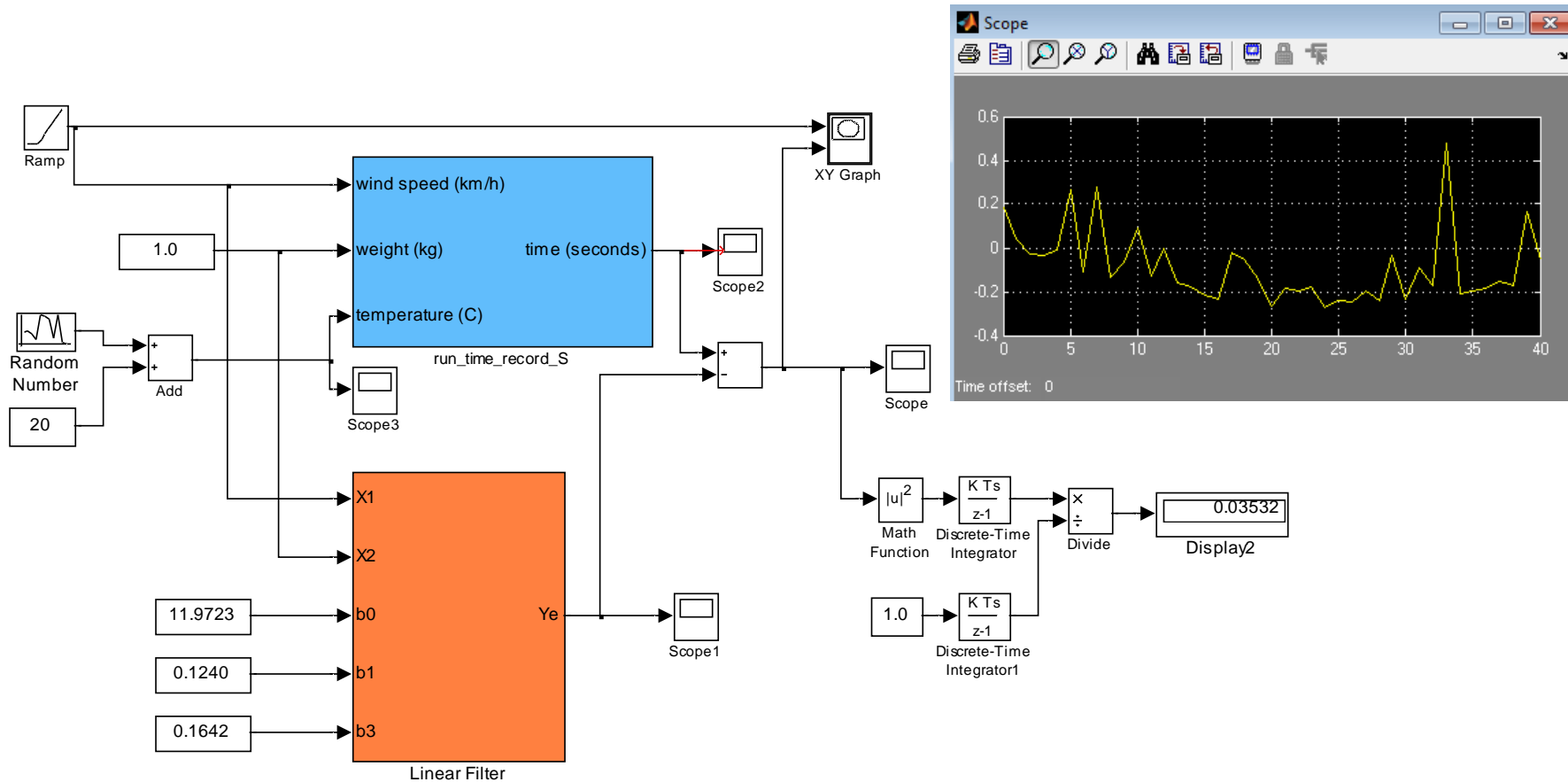
$$t = 12 + 0.1 \cdot x_1 + 0.0005 \cdot x_1^2 + 0.15 \cdot x_2 + 0.003 \cdot x_2^2 + 0.005 \cdot \text{abs}(T - 20)^2$$



Two-parameter estimation model in action (matlab-Simulink)



Two function blocks together: Estimation error



A Comparison of Mean Square Errors (MSE)

Case	MSE training	MSE test
1) Single parameter estimation	0.0588	0.0702
2) Single parameter polynomial estimation	0.0527	0.0610
3) Two parameter estimation	0.0243	0.0223
4) Two parameter polynomial estimation	0.0179	0.0163

As expected:

- Two parameter estimations result in smaller estimation errors because the physical function includes both factors, wind and weight, as input parameters.
- Polynomial estimations result in smaller estimation errors because the physical function also includes the squares of the factors wind and weight.

Download demo scripts for matlab & R

Download executable scripts that demonstrate four estimation cases at finaquant.com/download

- script_time_estimation_model_with_LR.m (matlab)
- script_time_estimation_model_with_LR.R

How to run these scripts?

- 1) Make sure that these scripts are in the current (working) directory
- 2) Enter command:

matlab> simple_time_estimation_model_with_LR

R> source('simple_time_estimation_model_with_LR.R')