Estimation by means of observers (Software Sensors)

Driss Boutat

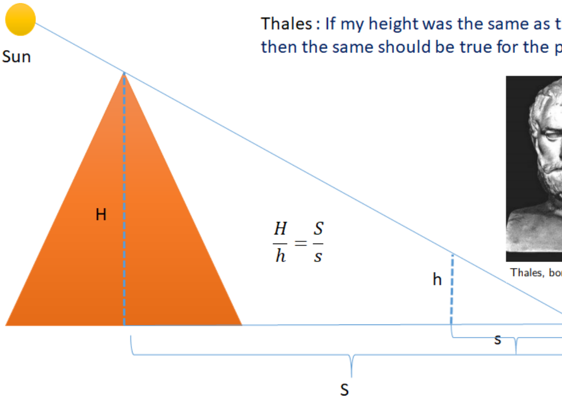
INSA Centre Val de Loire

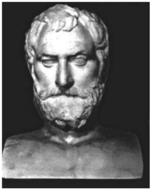
Fédération ICVL, JIRC 2018

Outline

Estimation is an old technique

Thales : If my height was the same as the length of my shadow, then the same should be true for the pyramid.



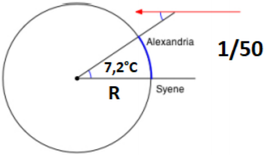


Thales, born around 624 B.C.,

In this case the model is : $H = \frac{S}{s} \times h.$

Estimation is an old technique

Eratosthenes circa 275–194 B.C.



distance between Syene and Alexandria is 5000 stadia

1 Stadium = 157.5 meters

Estimate of the circumference of the Earth 39 375 km

In this case the model is : $R = \frac{1}{\text{angle in radians}} \times \text{arc length}$.

Build predictive models (static case)

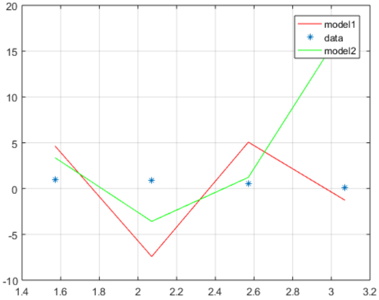
Machine learning algorithms for predictive modeling: from former input variables X_f and the corresponding output variables Y_f

$$F : \quad X_f \quad \rightarrow \quad Y_f.$$

First, build a function (learn a model) that maps the inputs to outputs $F(X_f) \simeq Y_f$.
Then, use the model F to make predictions of outputs Y_n for new input variables X_n.
Often a model F is a set of parameters to be estimated. It relies on optimisation algorithm such as: least squares, gradient descent, genetic algorithms, ...to estimate those parameters.

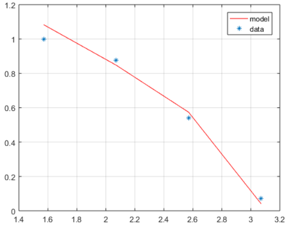
finite data samples

The truth reality cannot only be perceived through finite data samples or from simplified model built from the information lied in the data.



A valid model

We have a new information that tells us that the phenomena is periodic. This helped us to build a good model.



Models: dynamics case

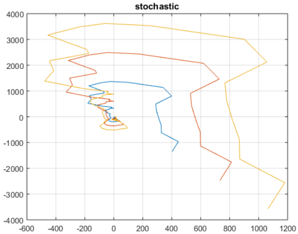
To supervise and predict the behavior of real objects or systems, we must describe them with a physical or mathematical representation that is closest to the observed data.

$$\dot{x} = f(x) + g(x)u,$$

where x is the state of our system, $\dot{x} = \frac{dx}{dt}$ the time derivative, $f(x)$ is vector field (velocity), $g(x)$ is the control direction and u is the control.

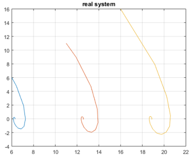
Behavior of the real system

Dynamics character: its state evolves in time and is random.

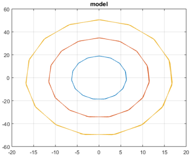


Simplified model

Approximations can lead to wrong results without random noise:

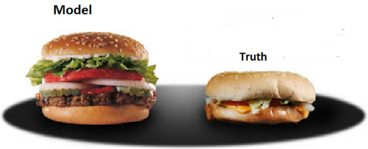


Linear approximation:



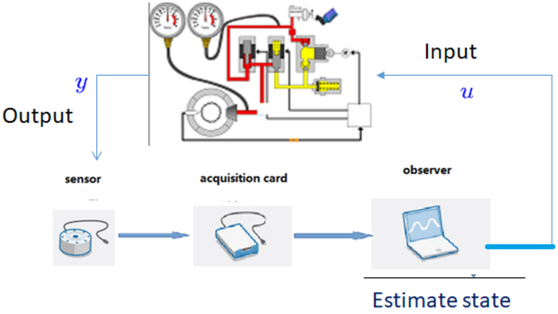
Simplified model

Far from reality



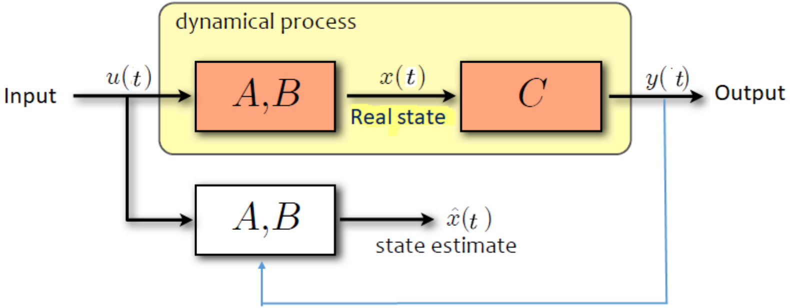
Problem estimation: software sensors

Derive more information from the measurements.



Build predictive models (dynamics case)

We master linear dynamical systems : $\dot{x} = Ax + Bu \quad y = Cx$



Observer design is software sensor that consists of building a copy of system's model $+$ an innovation term $K(y - \hat{y})$. There are many software that compute the gain K: Pole placement design.

Luenberger's observer

Canonical Normal form (Brunovsky)

$$\begin{cases} \dot{z} = A_0 z + \beta(y,u) \\ y = C_0 z = z_n \end{cases}$$

$$A_0 = \begin{pmatrix} 0 & 0 & .. & 0 & 0 \\ 1 & 0 & .. & .. & 0 \\ 0 & 1 & 0 & .. & .. \\ .. & .. & .. & .. & .. \\ 0 & 0 & .. & 1 & 0 \end{pmatrix}$$

Luenberger's Observer

$$\dot{\hat{z}} = A_0 \hat{z} + \beta(y,u) + K(y - \hat{y})$$

$$C_0 = \begin{pmatrix} 0 & 0 & .. & 0 & 1 \end{pmatrix}$$

Error Dynamics

$$e = z - \hat{z}$$
$$\dot{e} = (A_0 - KC_0)e$$

$$KC_0 = \begin{pmatrix} 0 & .. & .. & .. & k_0 \\ 0 & .. & .. & .. & k_1 \\ 0 & .. & .. & .. & .. \\ .. & .. & .. & .. & .. \\ 0 & .. & .. & .. & k_{n-1} \end{pmatrix}$$

K to be chosen such that the error goes to 0

$$e(t) = z(t) - \hat{z}(t) \xrightarrow[t \to +\infty]{} 0$$

$$K = \begin{pmatrix} k_0 & k_1 & .. & k_{n-1} & k_{n-1} \end{pmatrix}$$

Kalman's observer

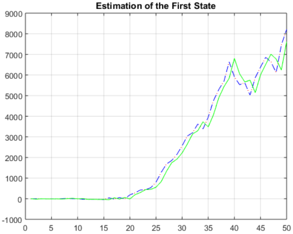
Consider the following dynamical system modelled as follows:

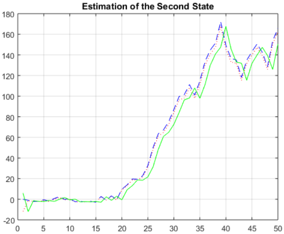
$$\begin{aligned} \dot{x} &= Ax + w \\ y &= Cx + v \end{aligned}$$

where x is the state and y is the measurement, w and v are the noises with covariances Q and R respectively. We will assume that noises w and v have zero-mean and are uncorrelated. Kalman's observer is given by:

$$\begin{aligned} \dot{x} &= Ax + K(y - \hat{y}) \\ K &= PCR^{-1} \\ \dot{P} &= AP + PA^T + Q - KRK^T \end{aligned}$$

Kalman's observer





Nonlinear dynamics: an example

Consider the following dynamical system:

$$\begin{aligned} \dot{x}_1 &= x_2 x_1 = y\dot{y} \\ \dot{x}_2 &= x_1 \\ y &= x_2 \end{aligned}$$

If the measurement is affected by noise: $y = \bar{y} + \epsilon \cos(\omega t)$ then its time derivative $\dot{y} = \dot{\bar{y}} - \omega\epsilon \sin(\omega t)$ can amplify the noise.
To overcome this problem: we will find conditions that allow a nonlinear dynamical system to be "linear" in a new coordinate system.

Nonlinear dynamics: an example

Consider the following dynamical system:

$$
\begin{aligned}
\dot{x}_1 &= x_2 x_1 = y\dot{y} \\
\dot{x}_2 &= x_1 \\
y &= x_2
\end{aligned}
$$

The following change of coordinates $z_1 = x_1 - \frac{1}{2}x_2^2$ and $z_2 = x_2$ leads to the following dynamics:

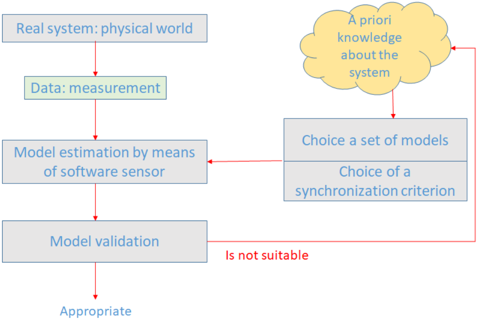
$$
\begin{aligned}
\dot{z}_1 &= 0 \\
\dot{z}_2 &= z_1 + \frac{y^2}{2}.
\end{aligned}
$$

we have transformed the nonlinear term which contains the derivative of the measurement into a nonlinear term which is only a function of the measurement.

Some remarks

- If the model is faithful to reality then the observer leads to good results.
- If we are sure enough of our observer design and the results are bad then we can conclude that our model is not good.
- The observer can be viewed as a tool that performs a synchronisation between the system and its model.

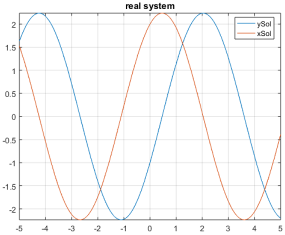
The observer-based dynamics model learning procedure



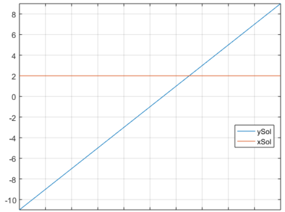
The observer-based dynamic model learning procedure

Example

Real Dynamics:

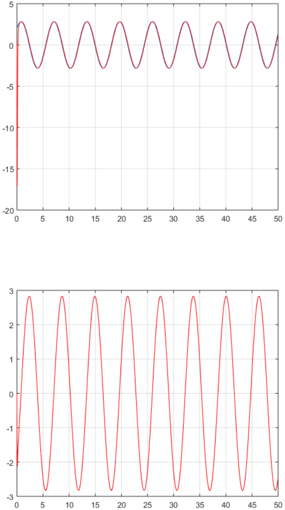


Initial model:



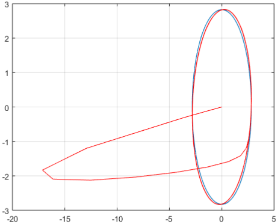
Example

Synchronisation



Example

Synchronisation



Reference

Do not overindulgence artificial intelligence.