

We assume that have a pyhysical system which can be described as general non-linear system in the form

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \mathbf{u})\end{aligned}$$

(To my knowledge) For every continous system description we can derive a discrete system description in a way that $\mathbf{x}_k = \mathbf{x}(t_0 + k\Delta t)$ for the free system $\mathbf{f}(\mathbf{x}, \mathbf{0})$. So every time-discrete state vector is exactly the sampled state vector of the continuous system. This is because of the markov property of any correct state description. So because the system is deterministic with a given current state vector any future state vectors can be calculated. If we have additionally a model to describe how the time-discrete values of \mathbf{u}_k can be related to the values of the continous input $\mathbf{u}(t)$, we are able to transform a between continous and discrete time domain without any loss of information. A possible model for the input could be Zero-Order-Hold.

A discrete time system description

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k)\end{aligned}$$

This system uses a state description which is together with the current input sufficient to describe the state in the next time step. This can more abstractly described as a markov property as

$$P[x_{k+1}|x_k, u_k] = P[x_{k+1}|x_k, u_k, x_{k-1}, u_{k-1}, \dots, x_0, u_0]$$

If we want to model a real system in such a manner the task is to find a proper state description of the real system. So if we measure the states of our model in the real system the deviation of the prediction to the actual measurements of the next state of the real system should be very small. Or the markov property should hold almost completely. A system description can be extended by defining the state of a new model to be the concatenation of past states (Augmented State vector)

$$z_k = [x_k, x_{k-1}, \dots, x_{k-N}]$$

In addition the description of the current input should always describe some quantities which are modifiable at the current state so the analogous process is not sensible for inputs. Rather we can do state augmentation by adding past inputs to the current state. So we can construct a large augmented state vector of the form

$$z_k = \begin{bmatrix} x_k \\ x_{k-1} \\ \dots \\ x_{k-N} \\ u_{k-1} \\ \dots \\ u_{k-N} \end{bmatrix}$$

Note that in this description the original state x_k and input u_k are vectors!

From this large system description it would be interesting to find a more dense state description of the system by some machine learning procedure.

From Ground Truth system description to Maximum System description

If we state the original system description was the relevant one the extended system is just the concatenation of the f function of the original system. If we look at the ground truth system description the omission of states should strongly increase the error of the prediction while the addition of states should not change the error at all (redundant information) (analogous as markov property: omission should break the markov property while addition should keep it)

Desribing static transformtations

By using the output equation $y_k = h(x_k, u_k)$ we can also describe fixed mathematical transformations like coordinate transformations if the output is just a function of the input

$$y_k = g(u_k)$$