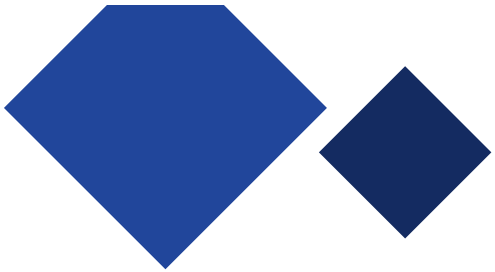
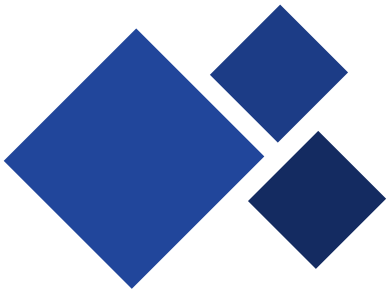


Lecture 11

Convolution and Fourier series in MatLab

Tomaso Erseghe



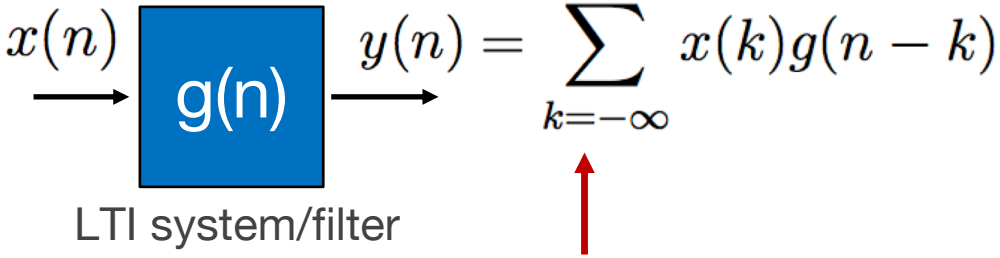
11.1 Convolution in MatLab

An overview

- Discrete-time convolution
- Approximate convolution in continuous-time

Discrete-time convolution

For limited-time signals

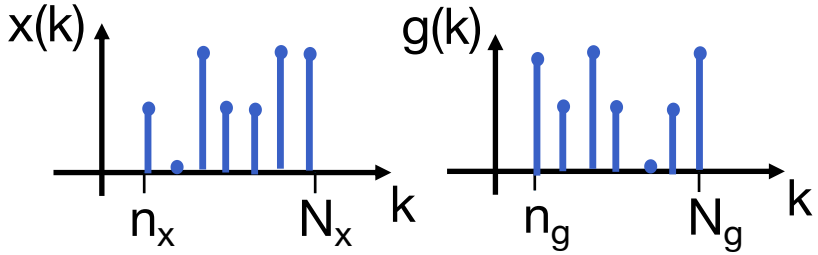


$$y(n) = \sum_{k=-\infty}^{\infty} x(k)g(n-k)$$

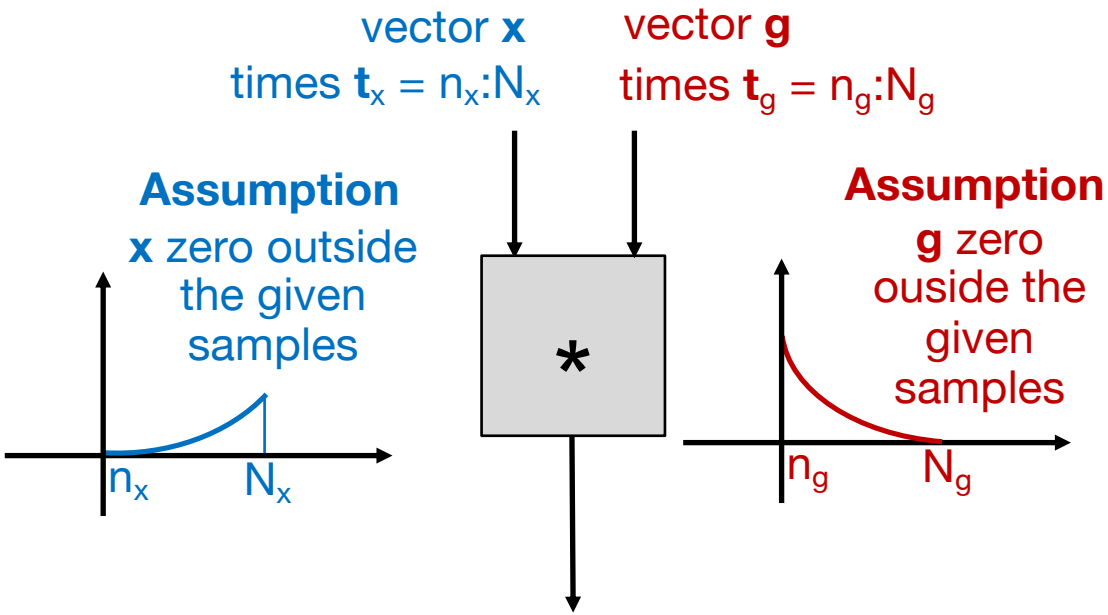
LTI system/filter

Limited only if limited signals

$e(x^*g) = [n_x+n_g, N_x+N_g]$

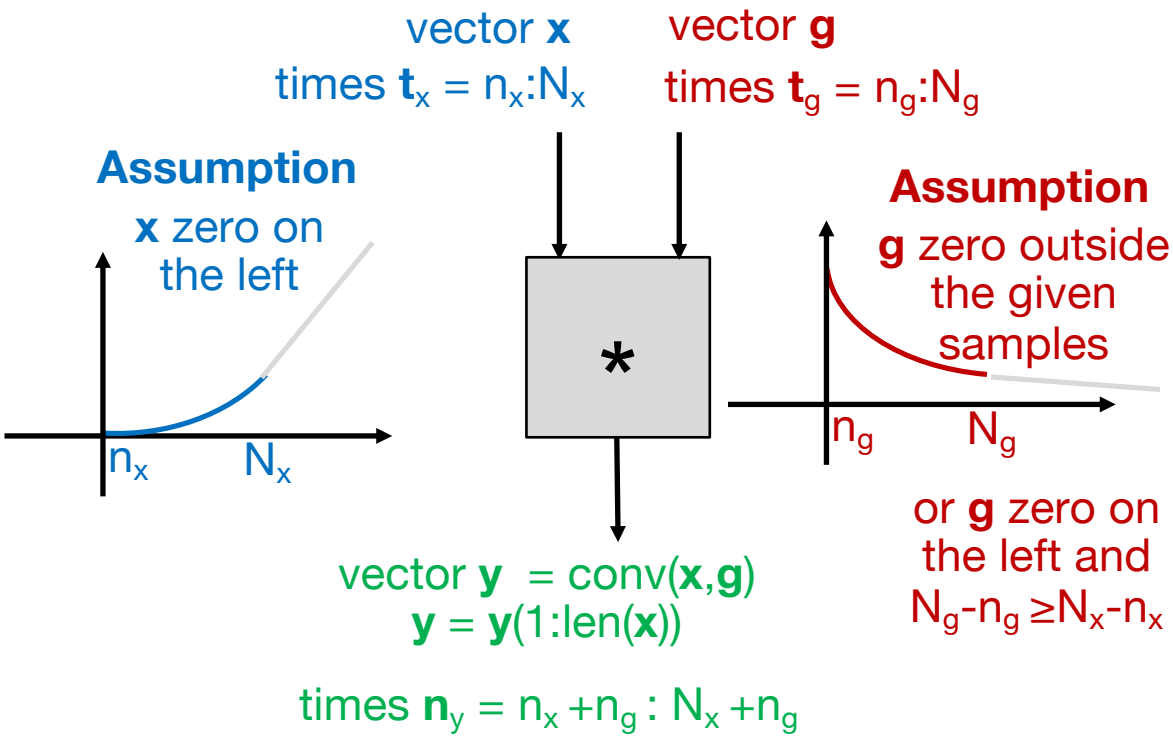


MatLab conv function



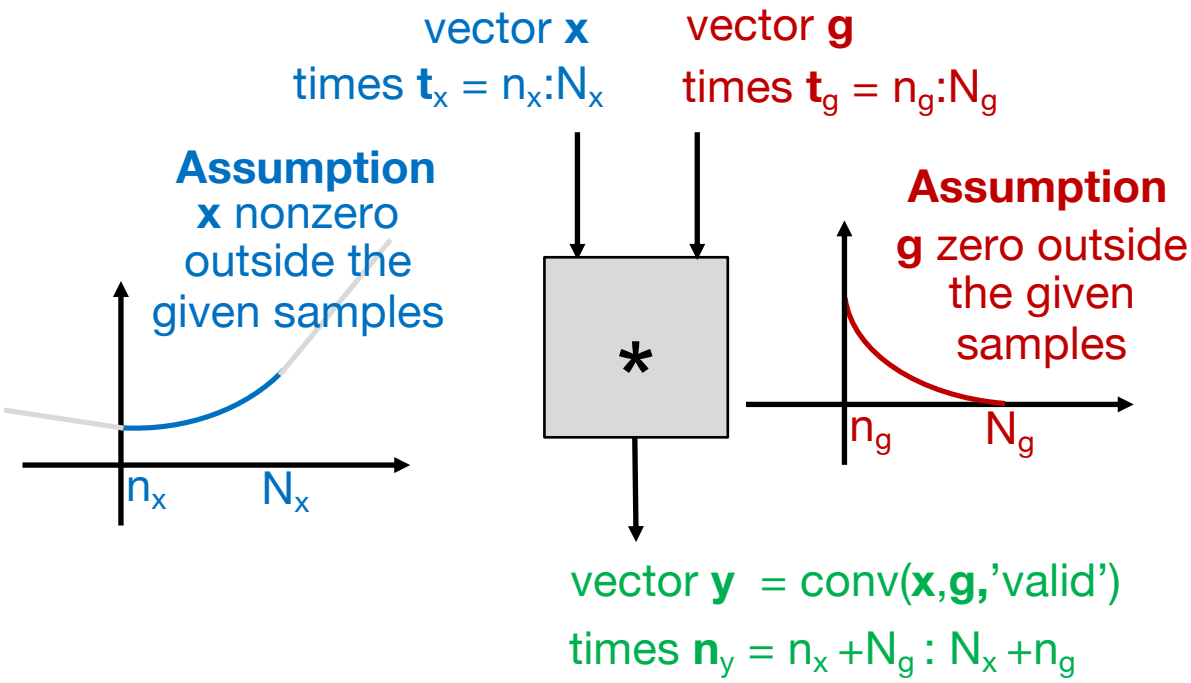
MatLab conv function

cut the result in case nonzero outside



MatLab conv function

‘valid’ = keeps an even smaller part



Continuous-time convolution

An approximation

samples of y

$$x * g(nT) = \int_{-\infty}^{\infty} x(u)g(nT-u)\,du$$

$$\simeq T \sum_{k=-\infty}^{\infty} x(kT)g(nT-kT)$$

sampling spacing: must be sufficiently (very) small

samples of x and g

y = T*conv(x,g)

$t_y = t_x + t_g : T_x + T_g$

Exercises

On the convolution in MatLab

Get acquainted with MatLab convolution operator **conv** and remind that when approximating a continuous-time convolution you will need to multiply by the sampling spacing T, to have **T*conv(x,g)**

Remember that the output of the convolution is not always valuable everywhere, e.g., in case signals are not zero outside the interval where samples are given

UNIVERSITÀ
DEGLI STUDI
DI PADOVA