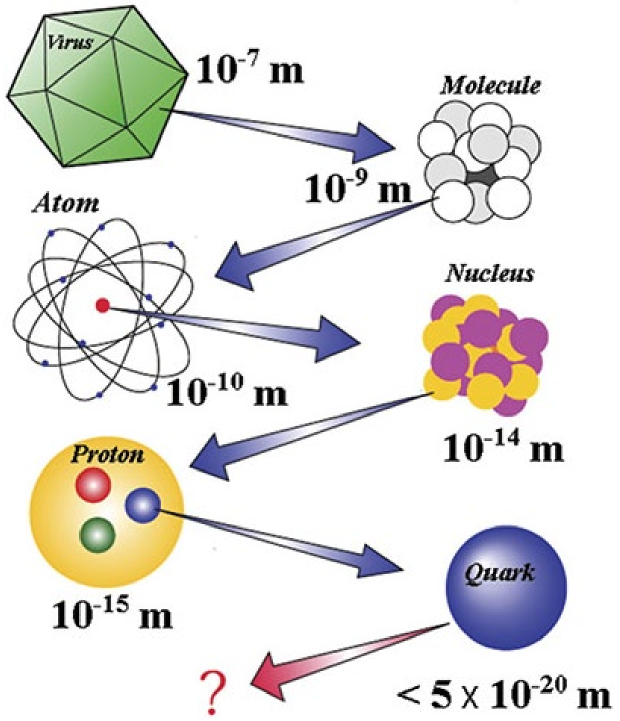
Fitting Digital Signals in Real-Time using the Magic of Convolutions

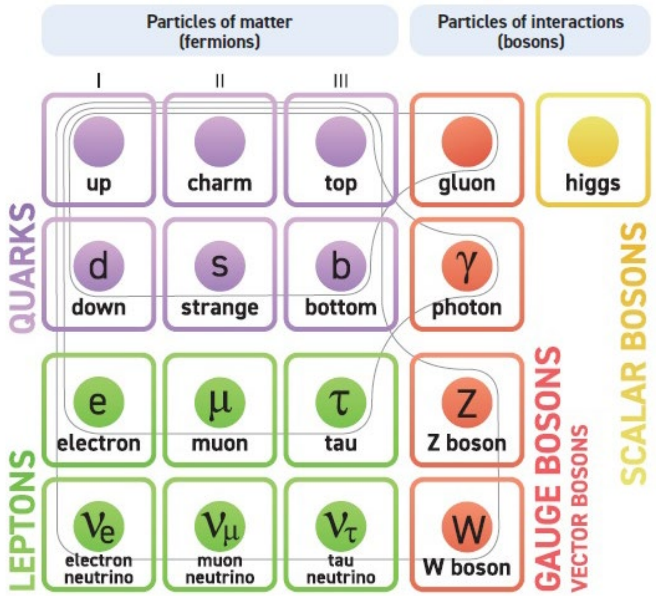
Christopher Crawford, University of Kentucky

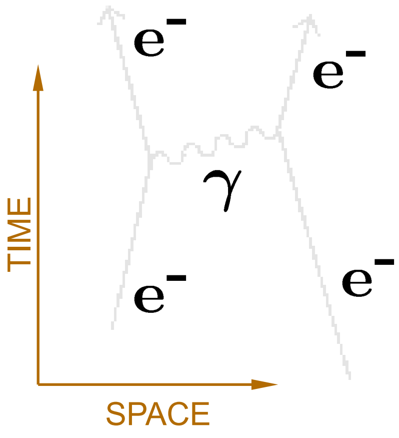
Research Computing and Data Seminar

UK Center for Computational Sciences, KY, 2024-02-13

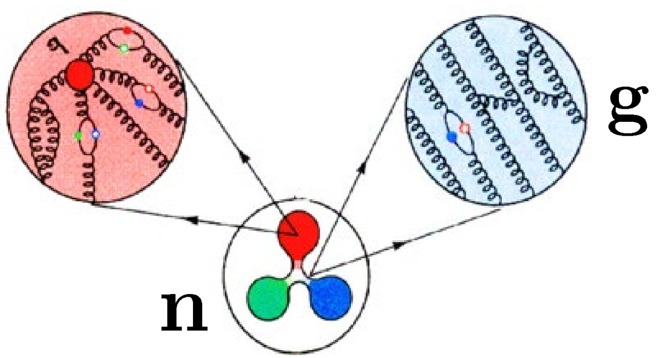
Standard Model (SM) of Particle Physics





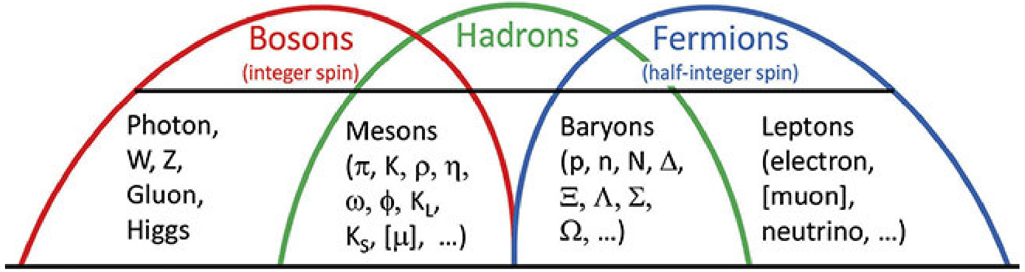


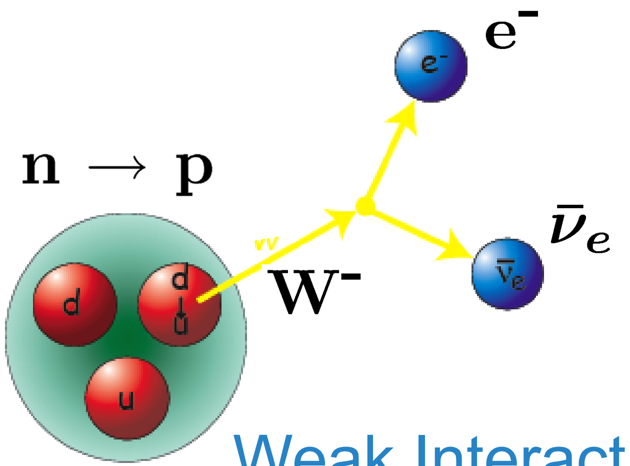
E&M Interaction



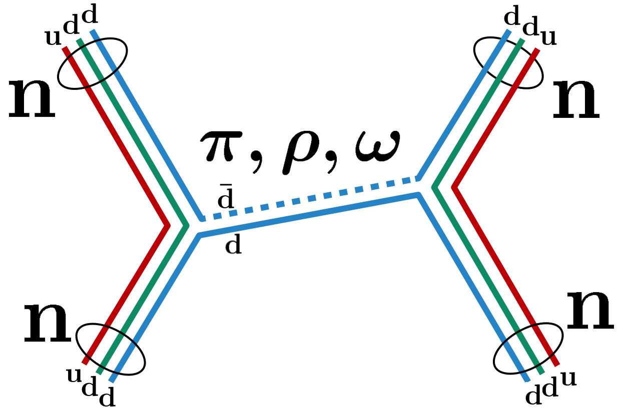
Strong Interaction

The Physics Teacher **56**, 204 (2018) https://doi.org/10.1119/1.5028231



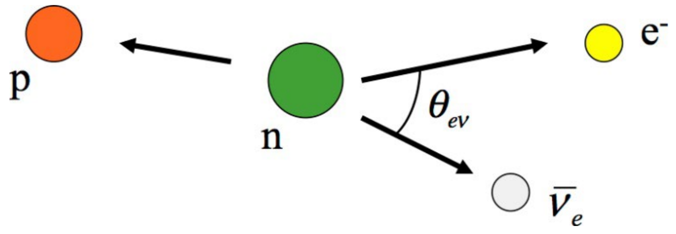


Weak Interaction



Hadronic Interaction
(residual nuclear force)

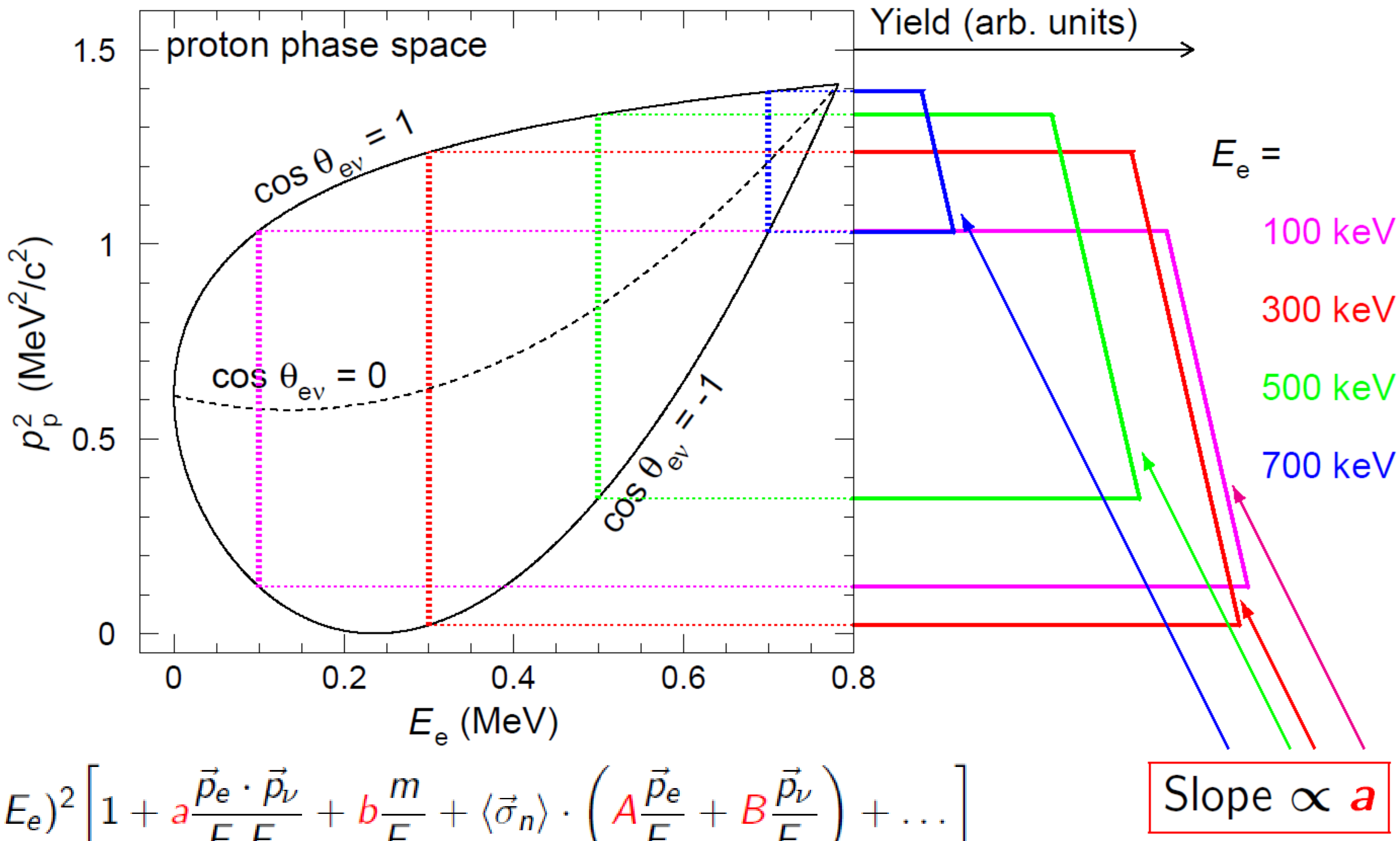
The Nab experiment



$$\vec{p}_p + \vec{p}_e + \vec{p}_\nu = 0$$

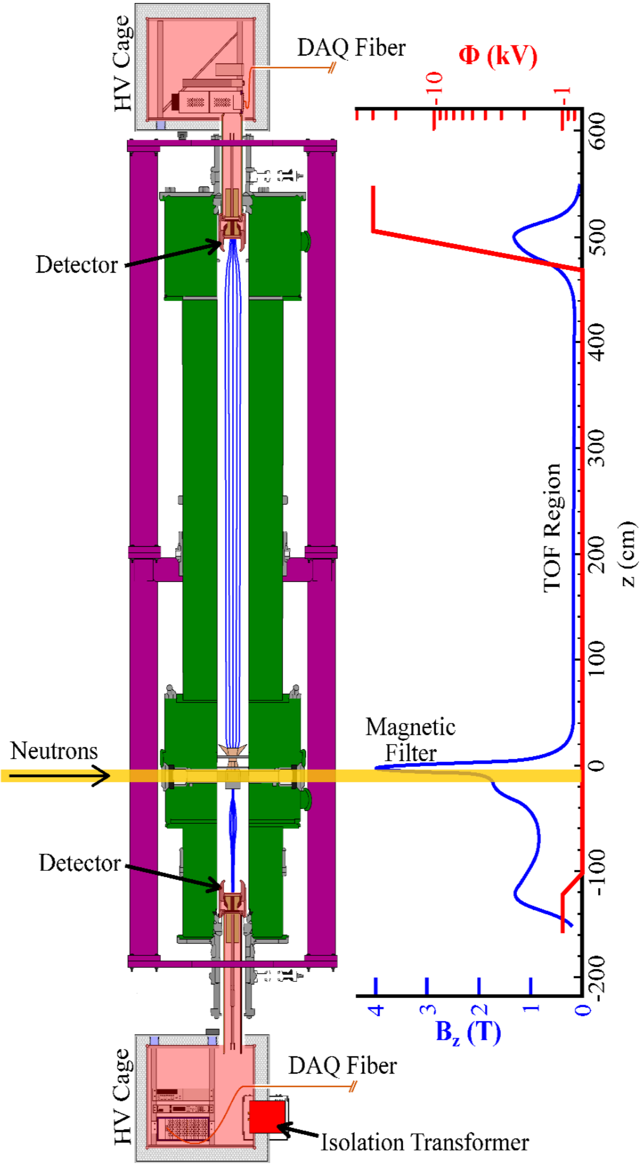
$$p_\nu = E_0 - E_e$$

$$p_p^2 = p_e^2 + 2p_e p_\nu \cos\theta_{e\nu} + p_\nu^2$$

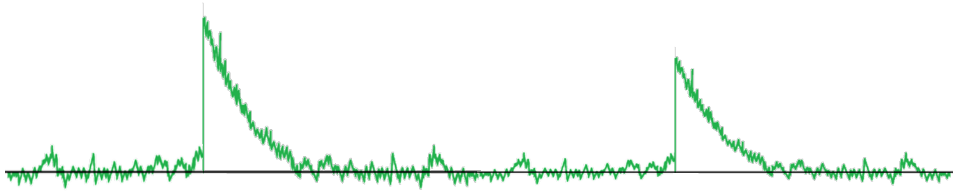


$$\frac{dw}{dE_e d\Omega_e d\Omega_\nu} \simeq p_e E_e (E_0 - E_e)^2 \left[1 + a\frac{\vec{p}_e \cdot \vec{p}_\nu}{E_e E_\nu} + b\frac{m}{E_e} + \langle \vec{\sigma}_n \rangle \cdot \left(A\frac{\vec{p}_e}{E_e} + B\frac{\vec{p}_\nu}{E_\nu}\right) + \dots\right]$$

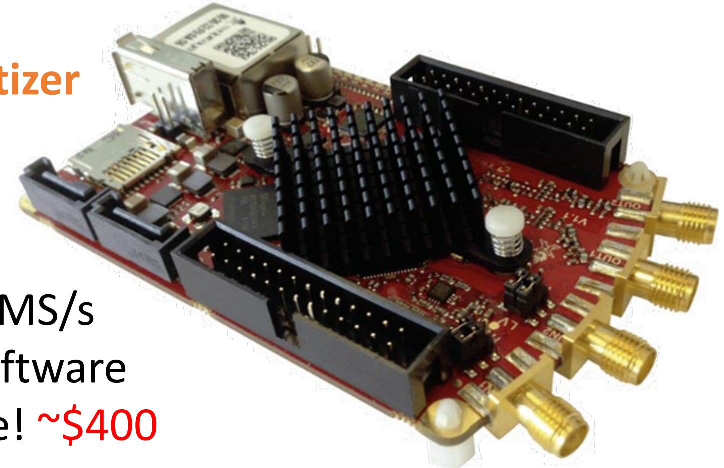
Nab detector

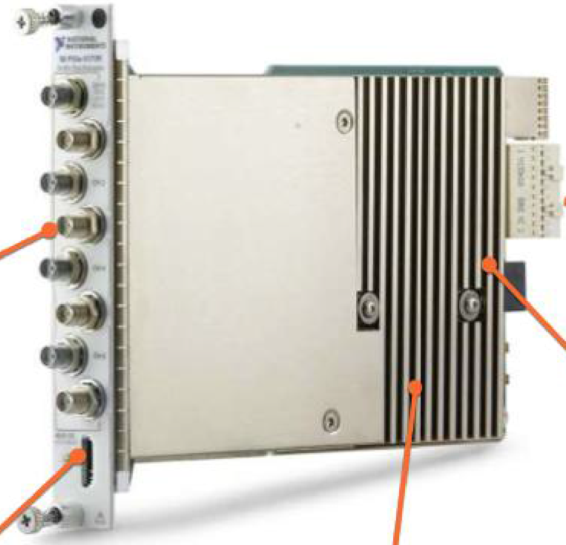


Data Acquisition System



Red Pitaya digitizer
Zynq 7010 SoC
4 GB RAM
2 ADC + 2 DAC
@ 14 bit, 125 MS/s
open-source software
AND firmware! ~$400





13.6 ENOB*

8 channels
- 250 MS/s, 14-Bit ADC
- 300 MHz bandwidth
- 0.2 to 5 Vpp input ranges
- Calibrated analog Inputs

User programmable IO
- 8 lines, up to 50 MHz
- For triggers, DUT control,
- Serial interfacing (I2C, SPI, JTAG, others)

PCIe Gen2 x8
- >3.2 GByte/s
- P2P streaming

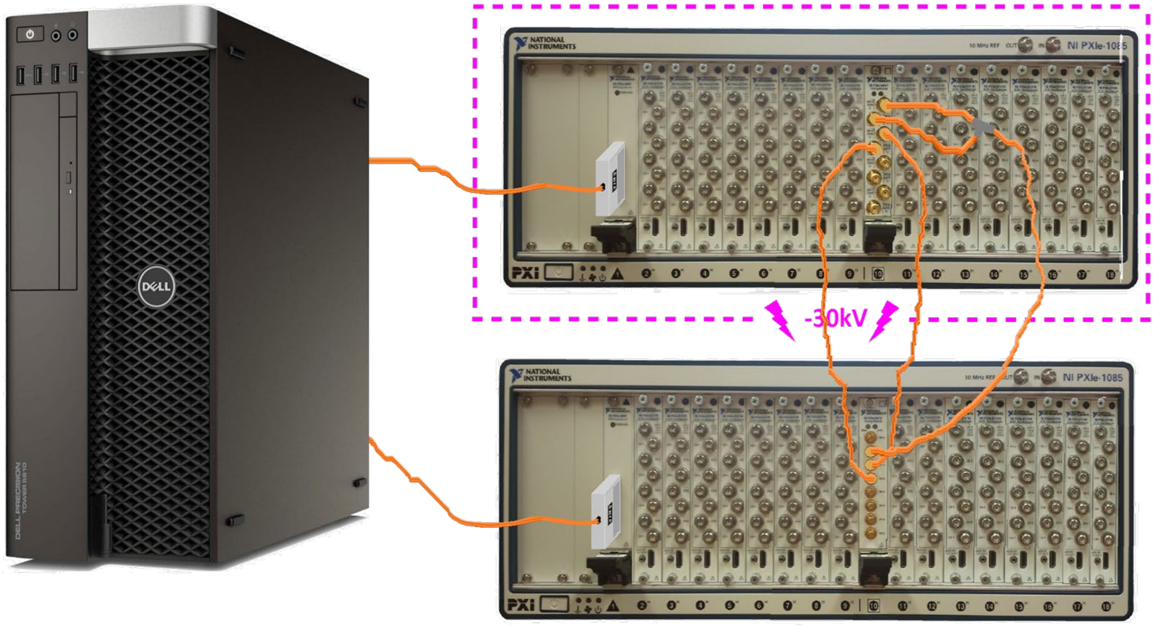
12 GBit RAM

0.375 s / ch

Kintex 7 FPGA
- processing,
- decision making

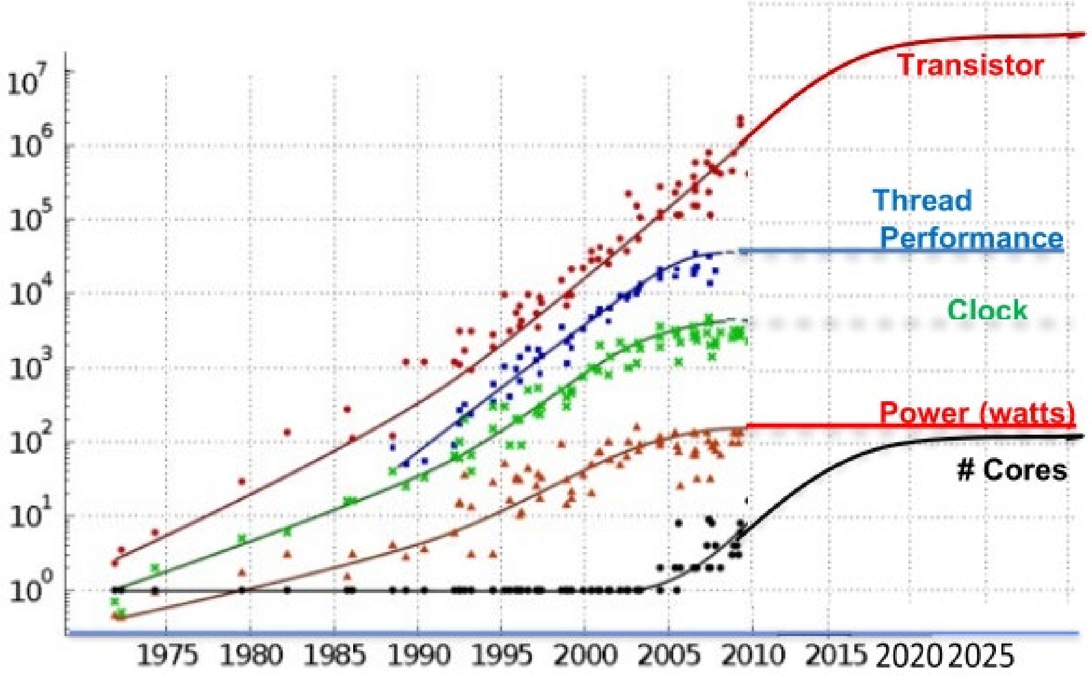
1540 DSP48 slices

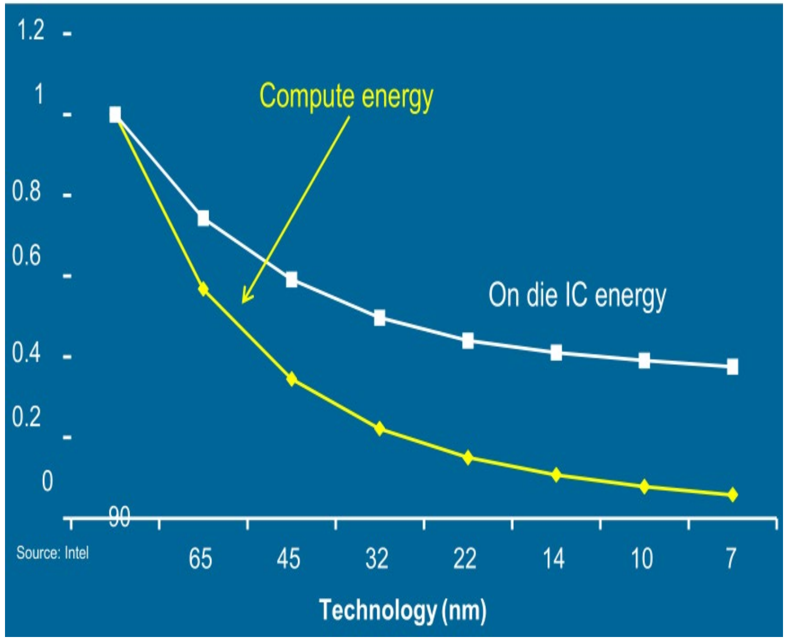
NATIONAL INSTRUMENTS



~$900k

Extreme Heterogeneity 2018

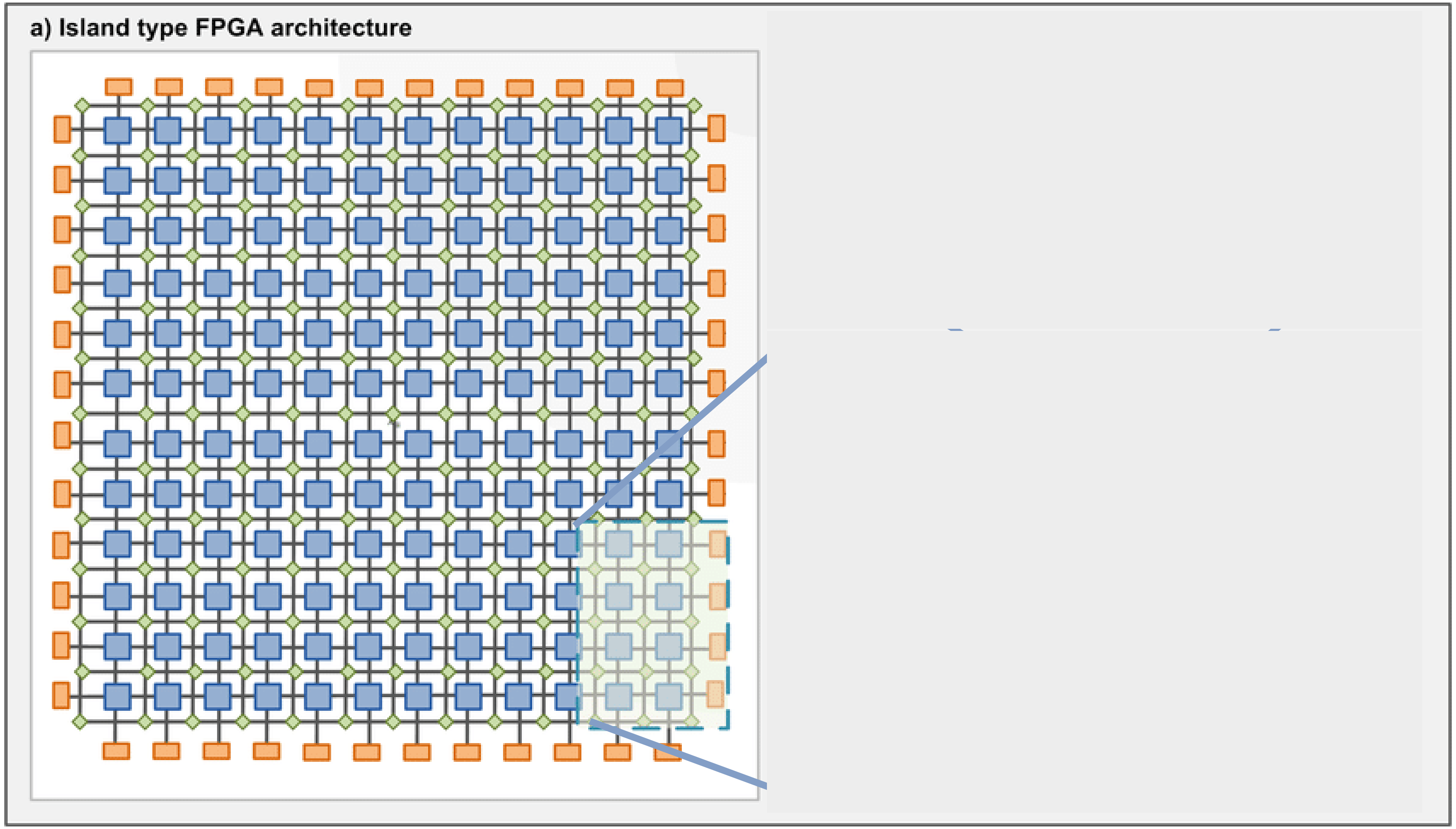




1. Dennard scaling – exponential clock speed till 2004
   - *exponentially increasing parallelism*
2. Moore scaling – exp. incr. transistor density till 2020's
   - *heterogeneous hardware acceleration*
3. Data transport vs processing speed (ex. GPUs)
   - *new architectures for in-place computation*

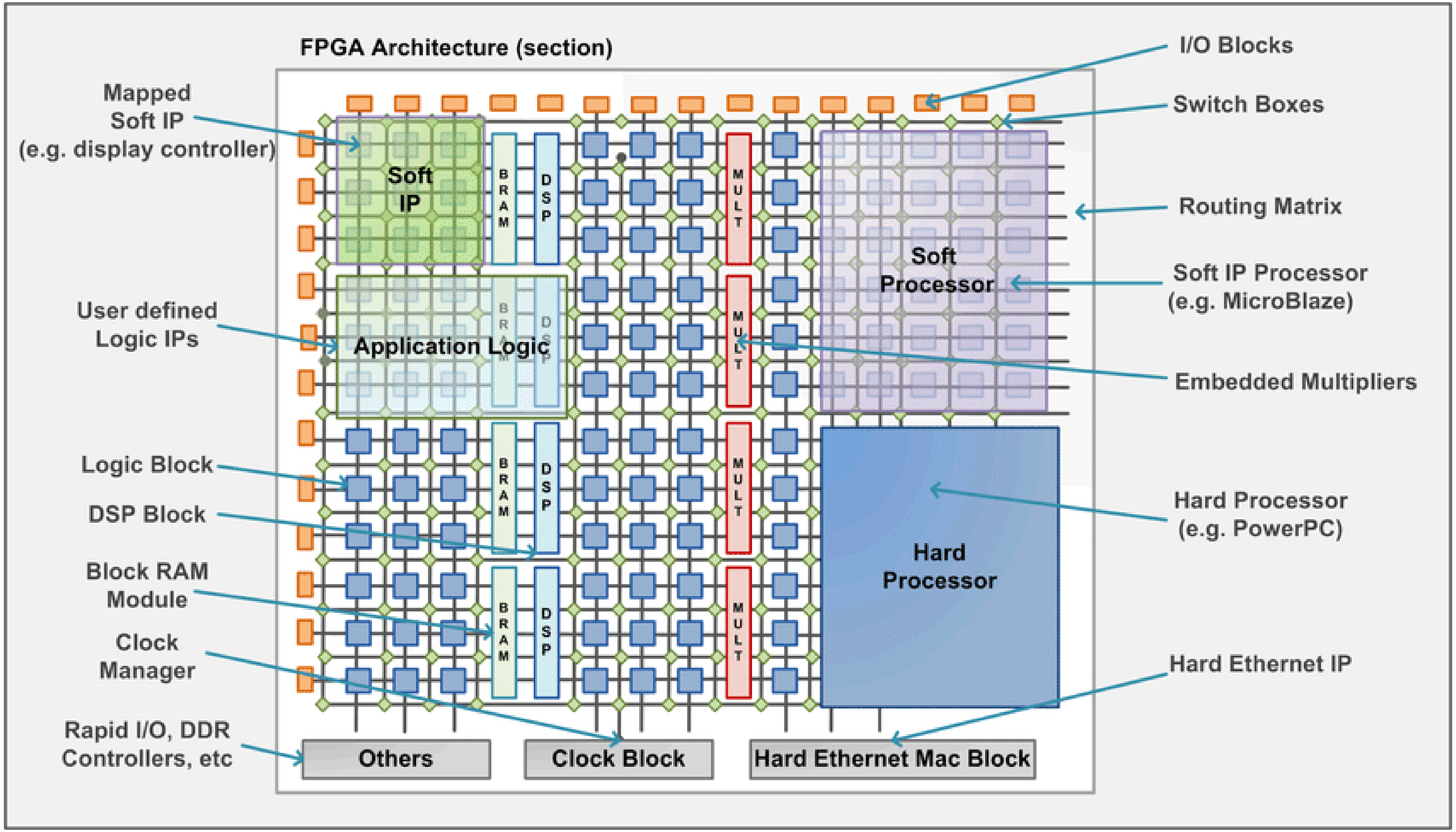
FPGA Components

I/O, Clock, Switch Matrix, Configurable Logic Block, DSP



https://vhdl.es/fpga

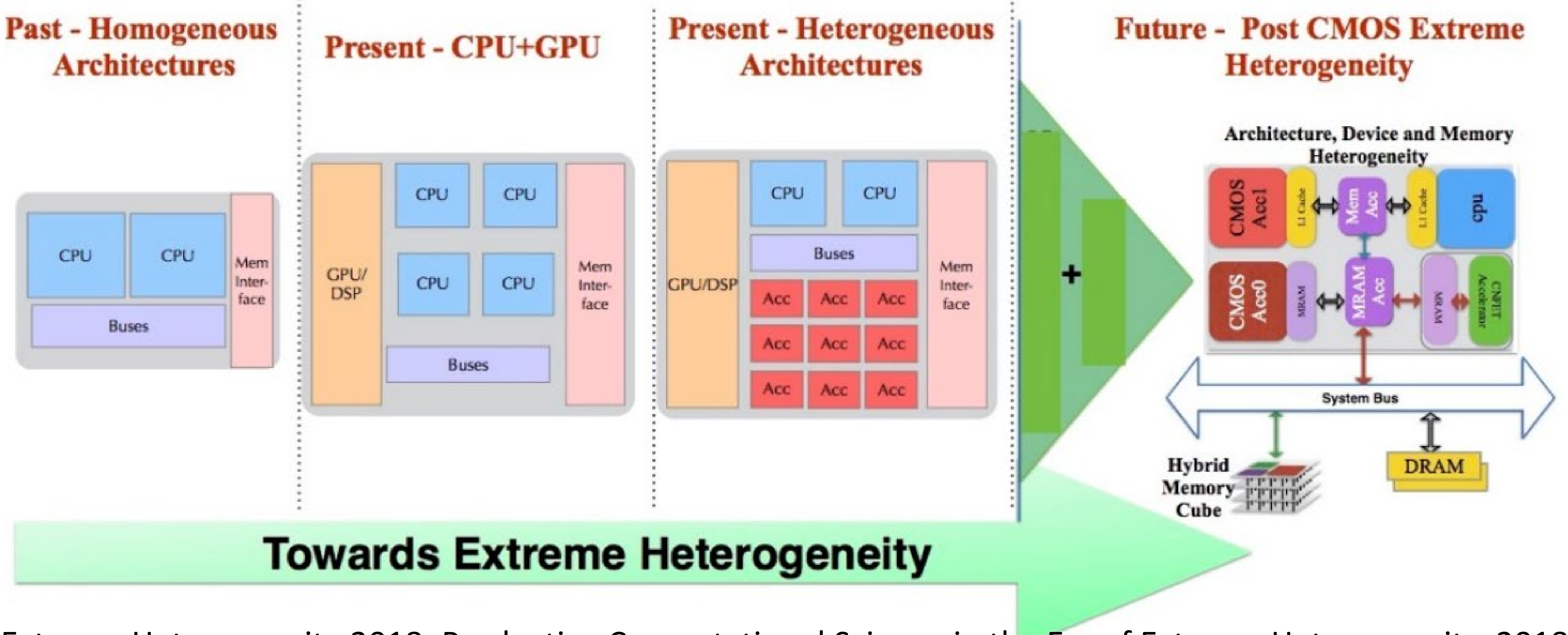
FPGA Components



Thesis, A. Ochoa-Ruiz Gilberto, 2013

New algorithms for extreme heterogeneity

- post-Moore's Law—shift toward dedicated hardware accelerators
- FPGA compute modules suitable for highly repetitive tasks



from: Extreme Heterogeneity 2018: Productive Computational Science in the Era of Extreme Heterogeneity, 2018-01-23

Digital FIR filters

the Convolution – atomic force microscopy (AFM)

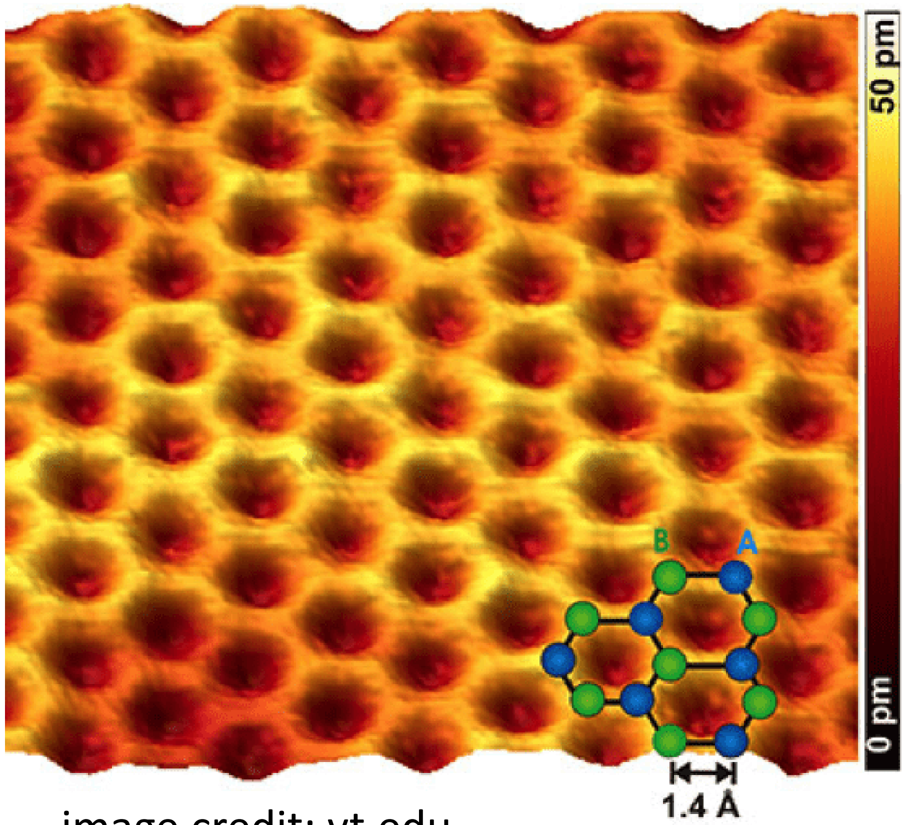
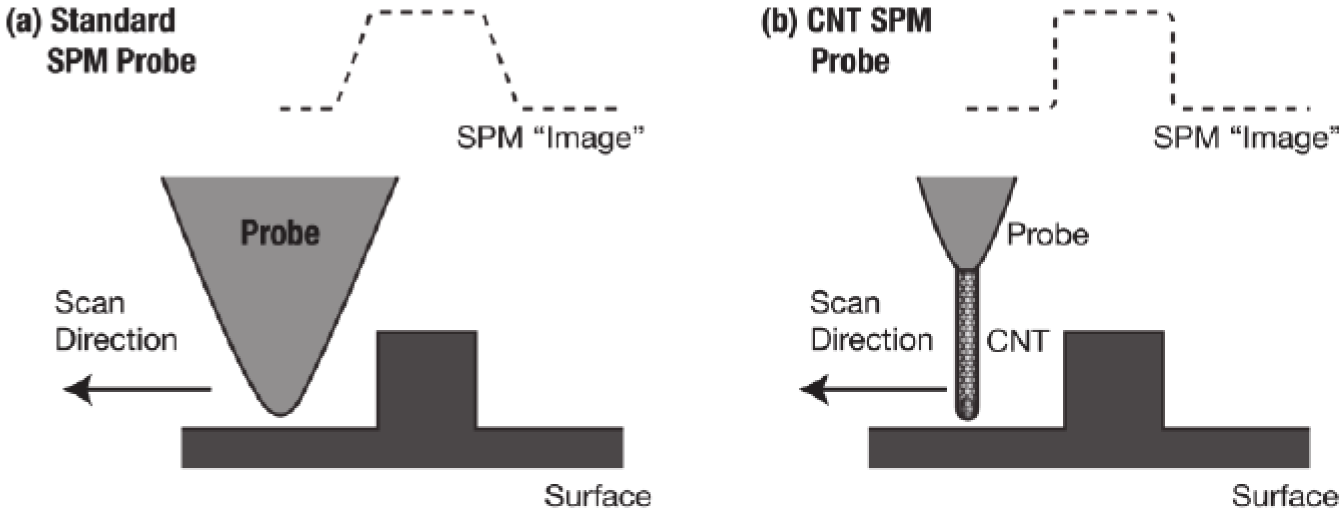


image credit: vt.edu



Clark, Ian & Yoshimura, Masamichi. (2011) 10.5772/18377

the Convolution – the 'star' of the show

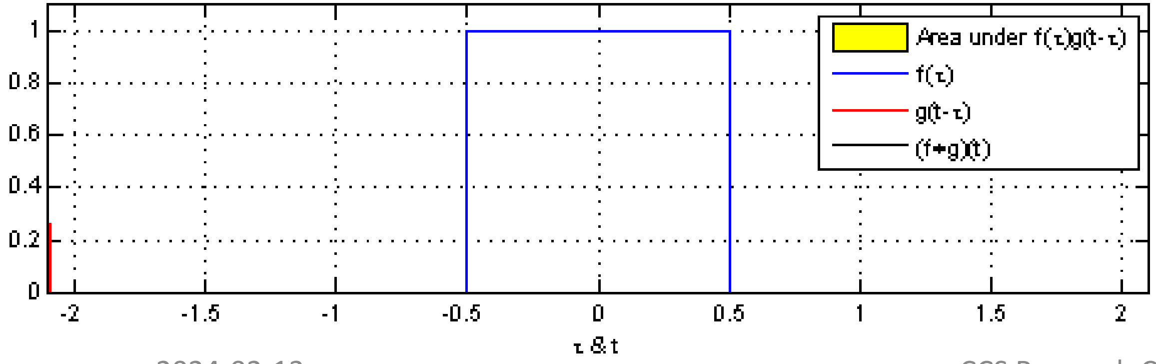
- $(a * b)[n] = \sum_{m=-\infty}^{\infty} a[m]b[n-m]$

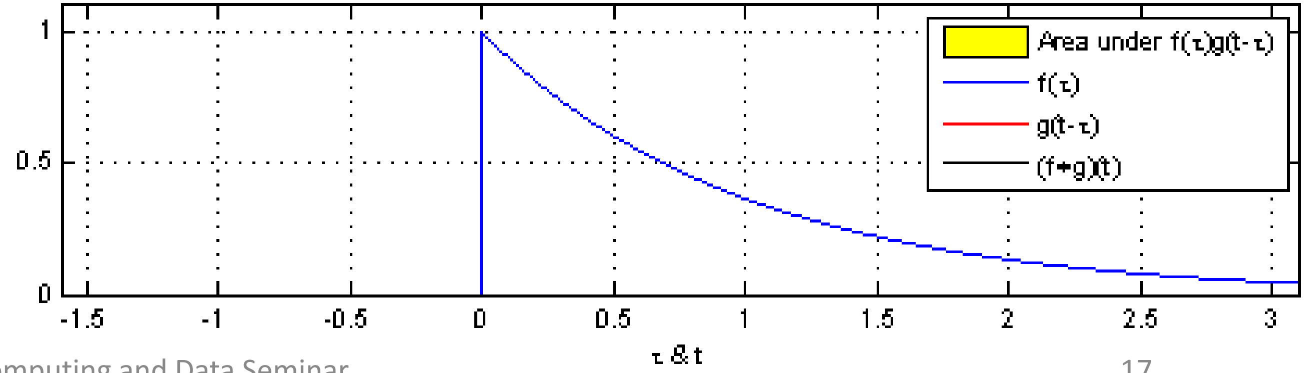
  discrete: "sliding dot product"

- $(f * g)(t) = \int_{-\infty}^{\infty} d\tau\, f(\tau)g(t-\tau)$

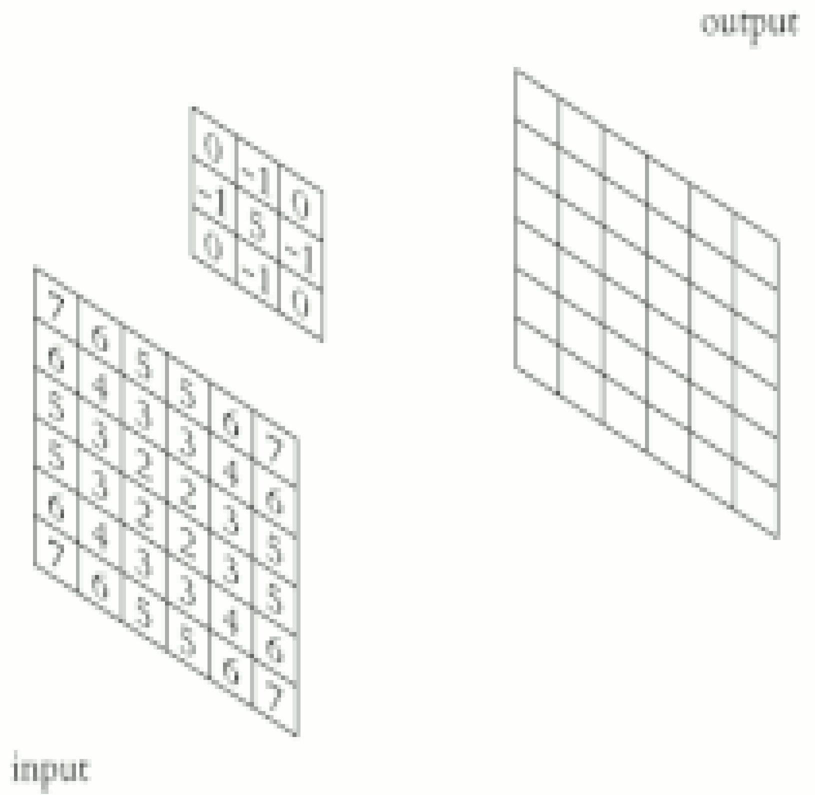
  continuous: "Faltung" German: folding

Wikipedia: convolution





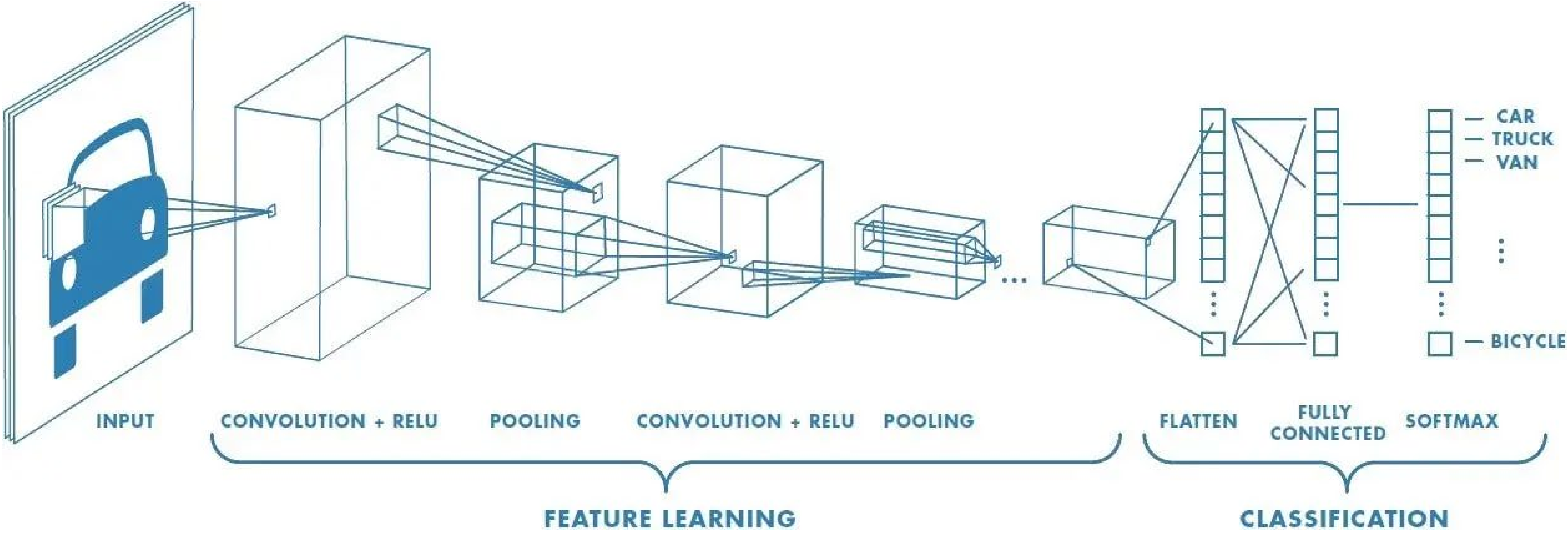
Examples – image processing



- average: filter = [1 1 1]   — blurring
- derivative: filter = [1 0 -1]   — edge detection

Wikipedia: Wiener deconvolution
Wikipedia: convolution

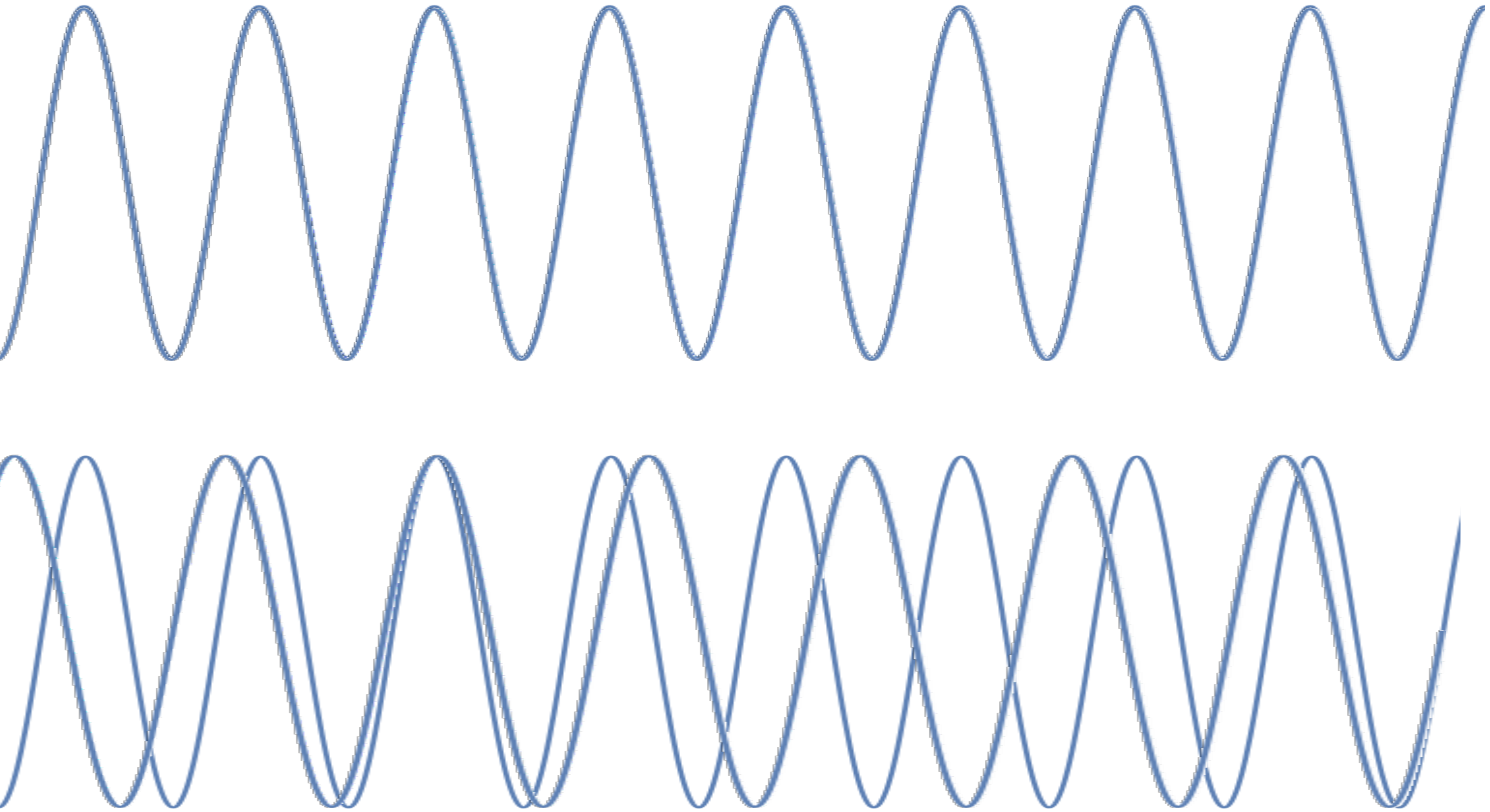
Examples — Convolutional Neural Networks



Properties of the Convolution

- commutative
- associative
- distributive (bilinear)
- derivative/integrals

Sinusoidal Convolutions



Sinusoidal convolutions

Fourier transform

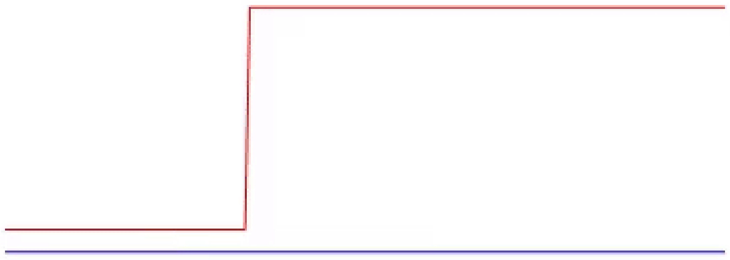
- Convolution theorem

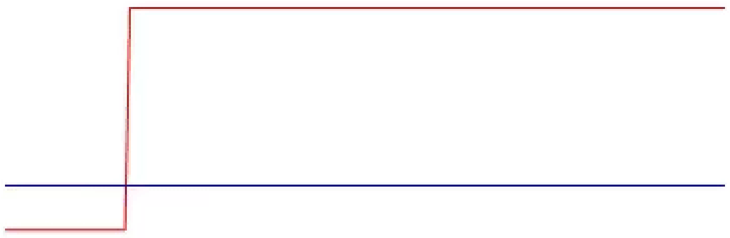
Building blocks

- basic arithmetic
  adder, multiplier -> Multiply-Accumulate in DSP48 slice
- unit impulse
  basis function delay (identity) = result shift register/buffer
- unit step
  const function integration = result accumulator
- properties of convolution
  - commutative (function <-> filter)
  - associative (build up complex filters in stages)

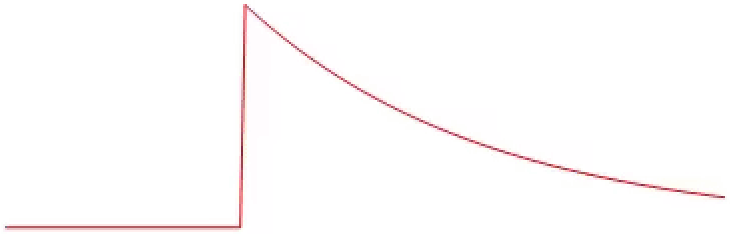
Boxcar filter

Baseline subtract



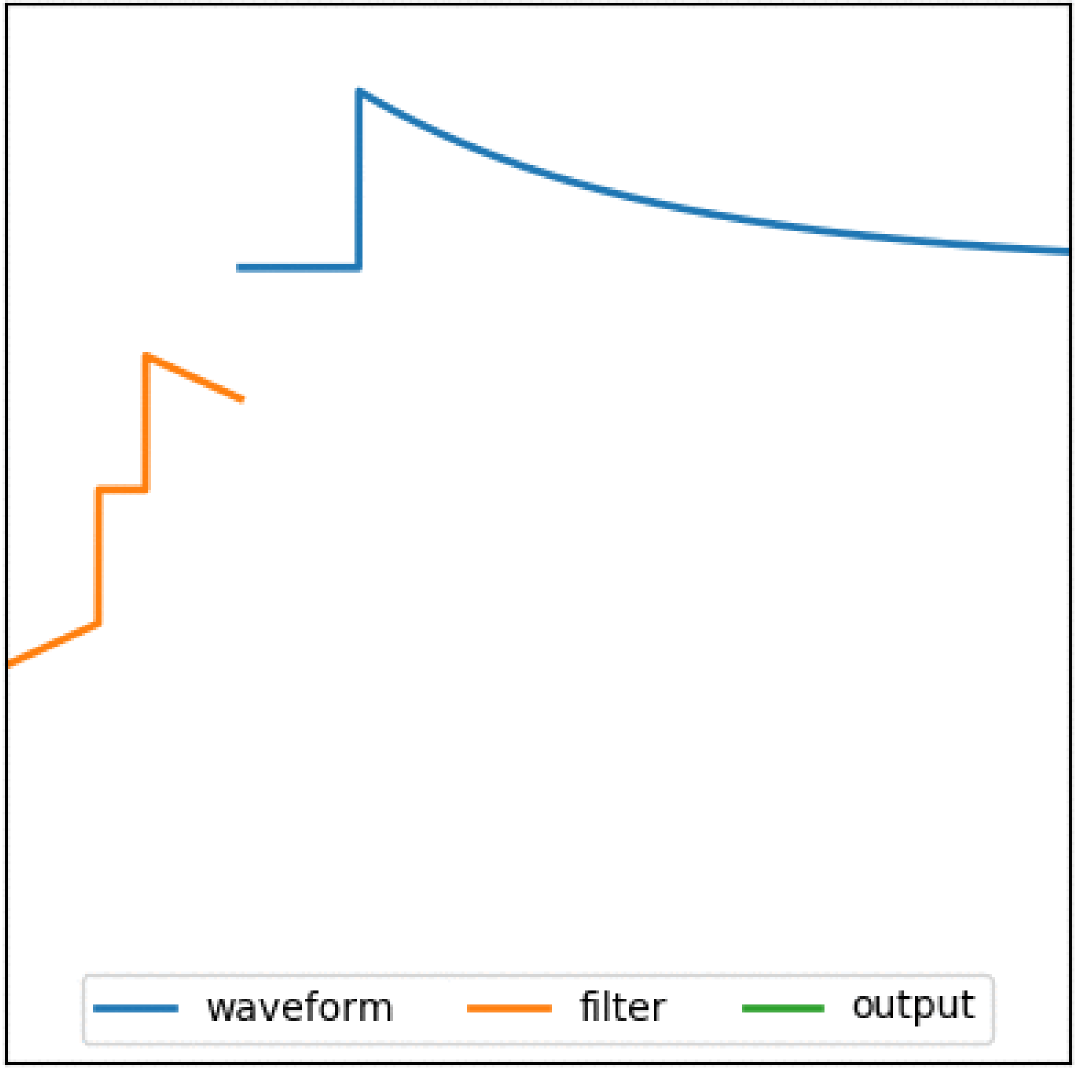


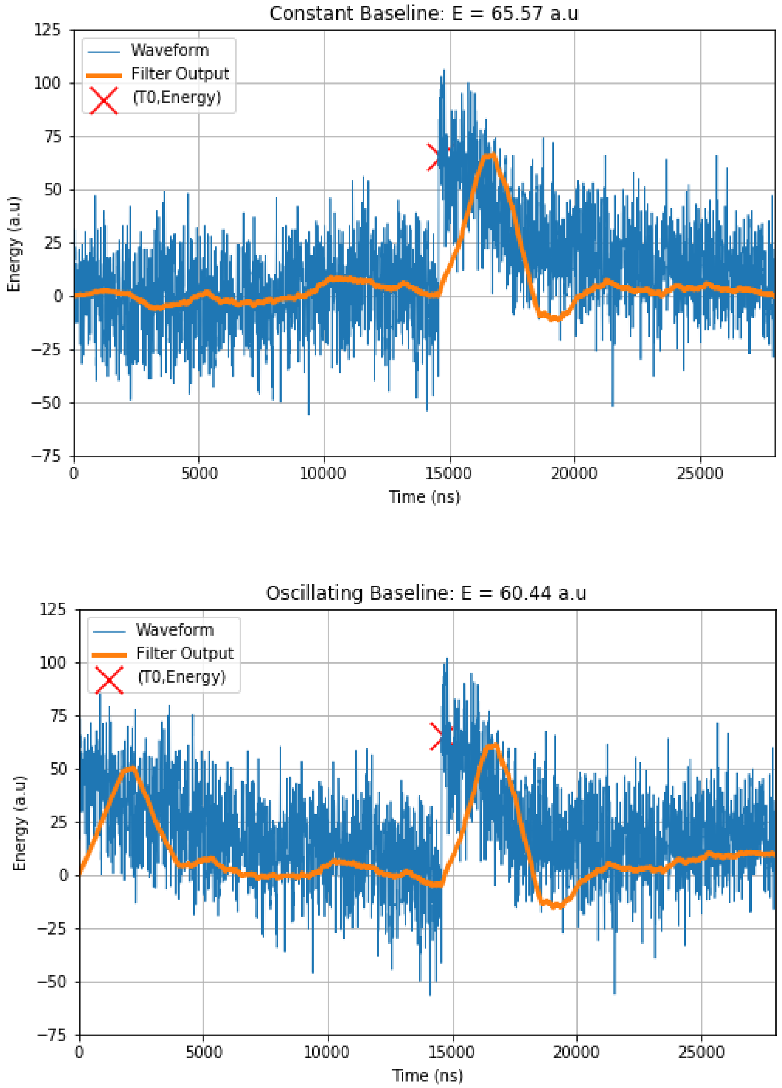
Tail pulse
pole-zero



Trapezoid filter

Full Trapezoid filter

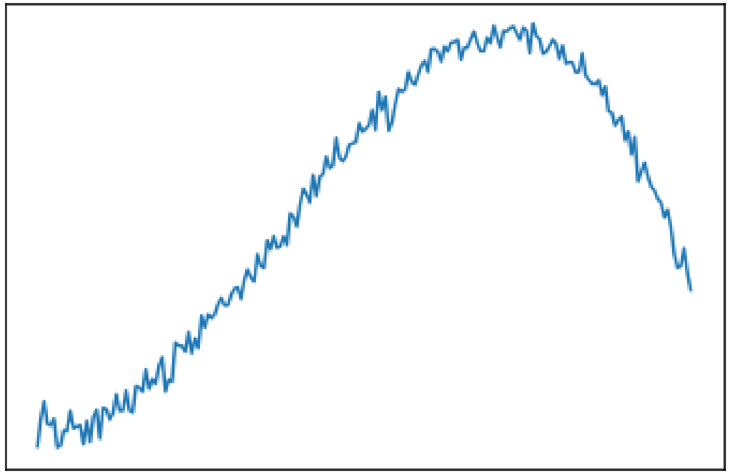




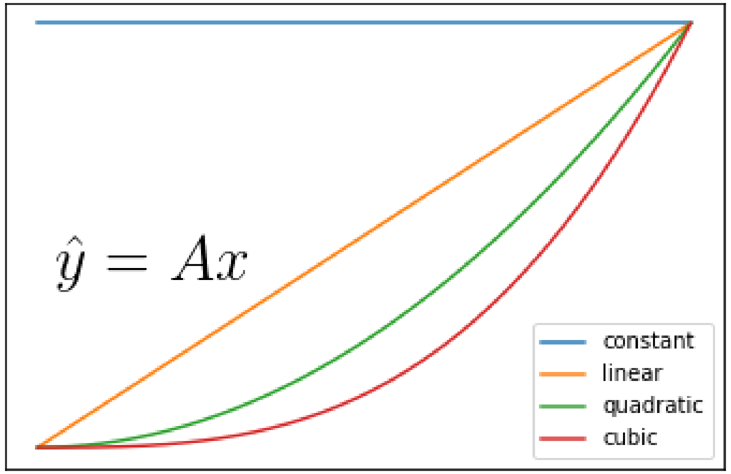
David Mathews, University of Kentucky, APS DNP talk, April 2019

Least Squares fitter – Piecewise Recursive Polynomial filter

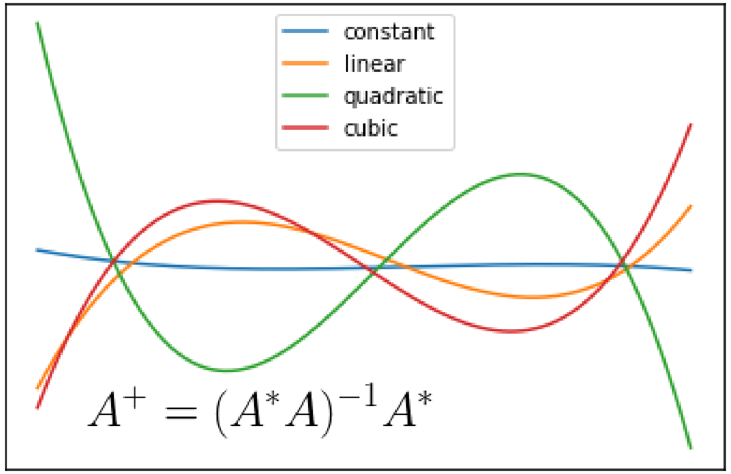
Original data



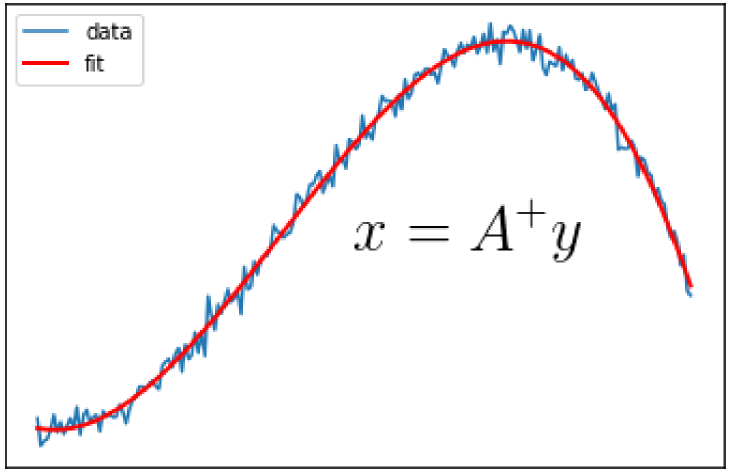
Fit Functions: Design Matrix A



Pseudoinverse of design matrix



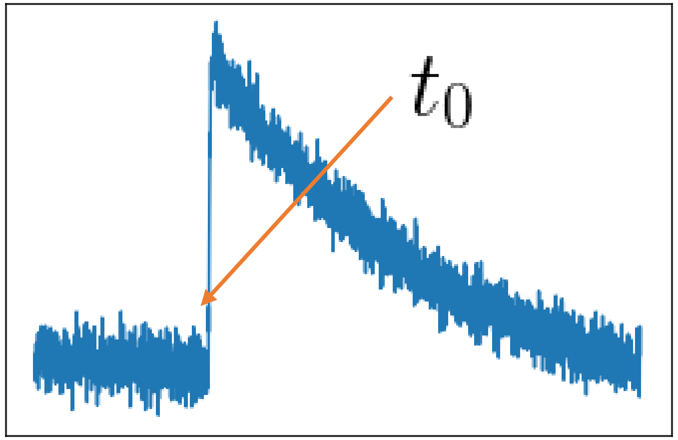
Fit parameters



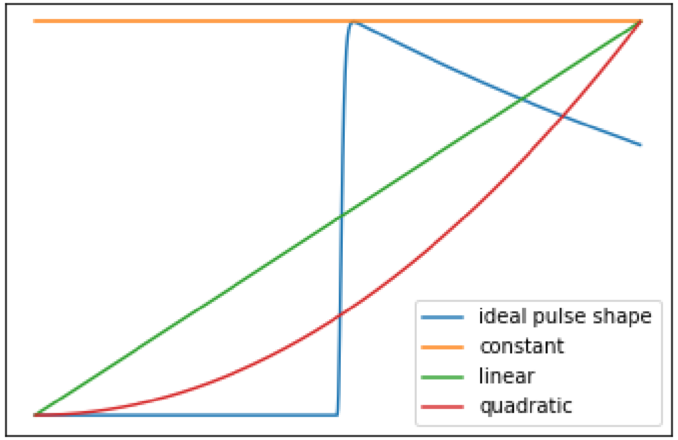
David Mathews, University of Kentucky, APS DNP talk, April 2019

Least Squares fitter – Piecewise Recursive Polynomial filter

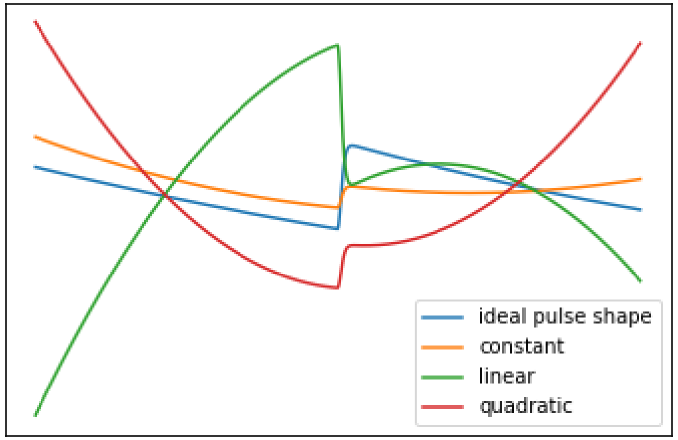
Data: y



Fit Functions: A



Pseudoinverse Fit Functions A^+



Calculate Fit Parameters:

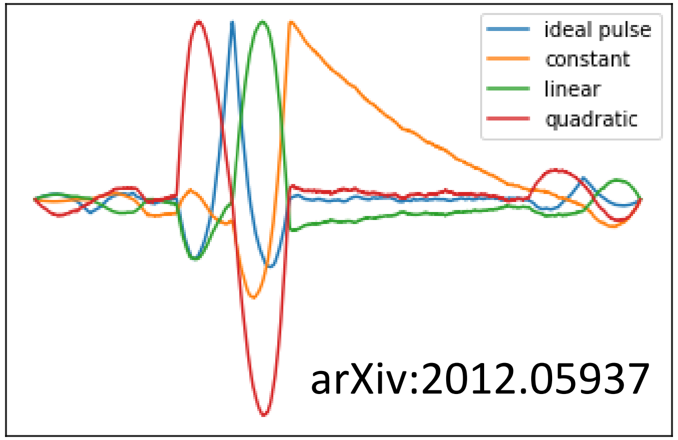
$$x(t_0) = A^+ * y$$

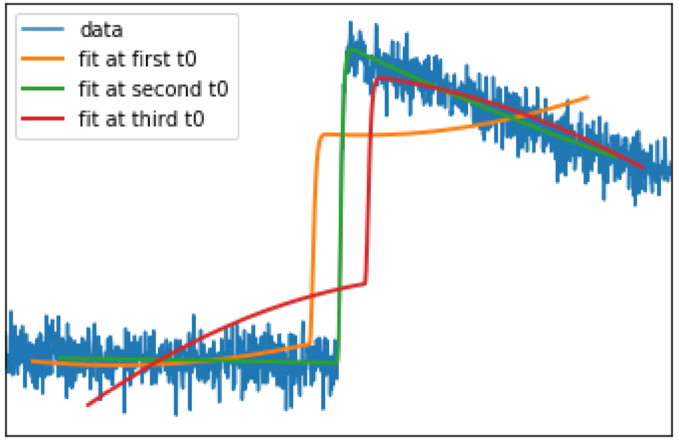
Fit using Parameters

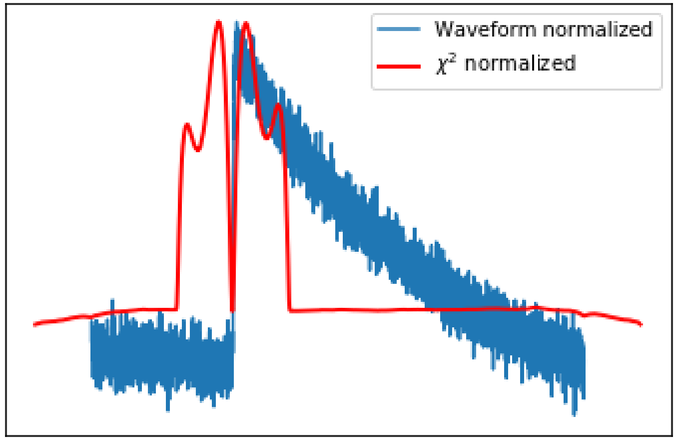
$$\hat{y}(t_0) = Ax(t_0)$$

Find Best Fit with χ^2

$$\chi^2_{t_0} = (y^T y)_{t_0} - x^T_{t_0} A^T A x_{t_0}$$





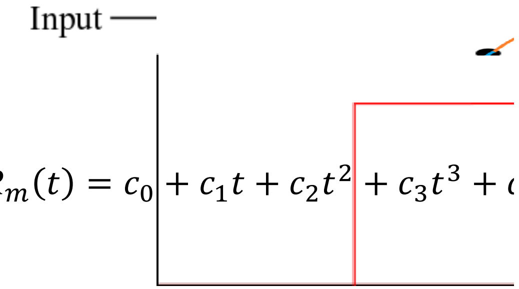


David Mathews, University of Kentucky, APS DNP talk, April 2019

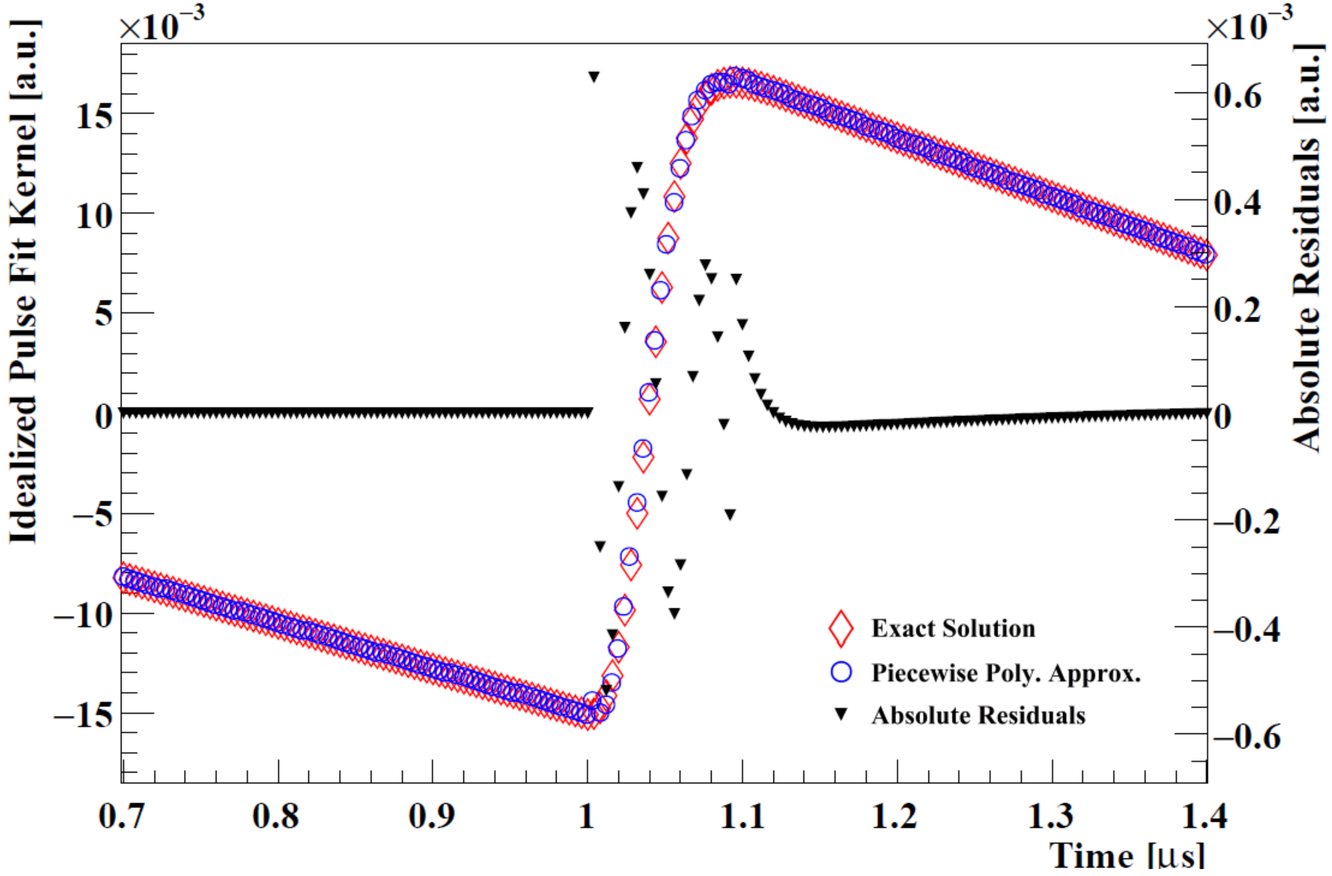
Generic Filter Design

From Jezghani and Knoll:

[illegible]

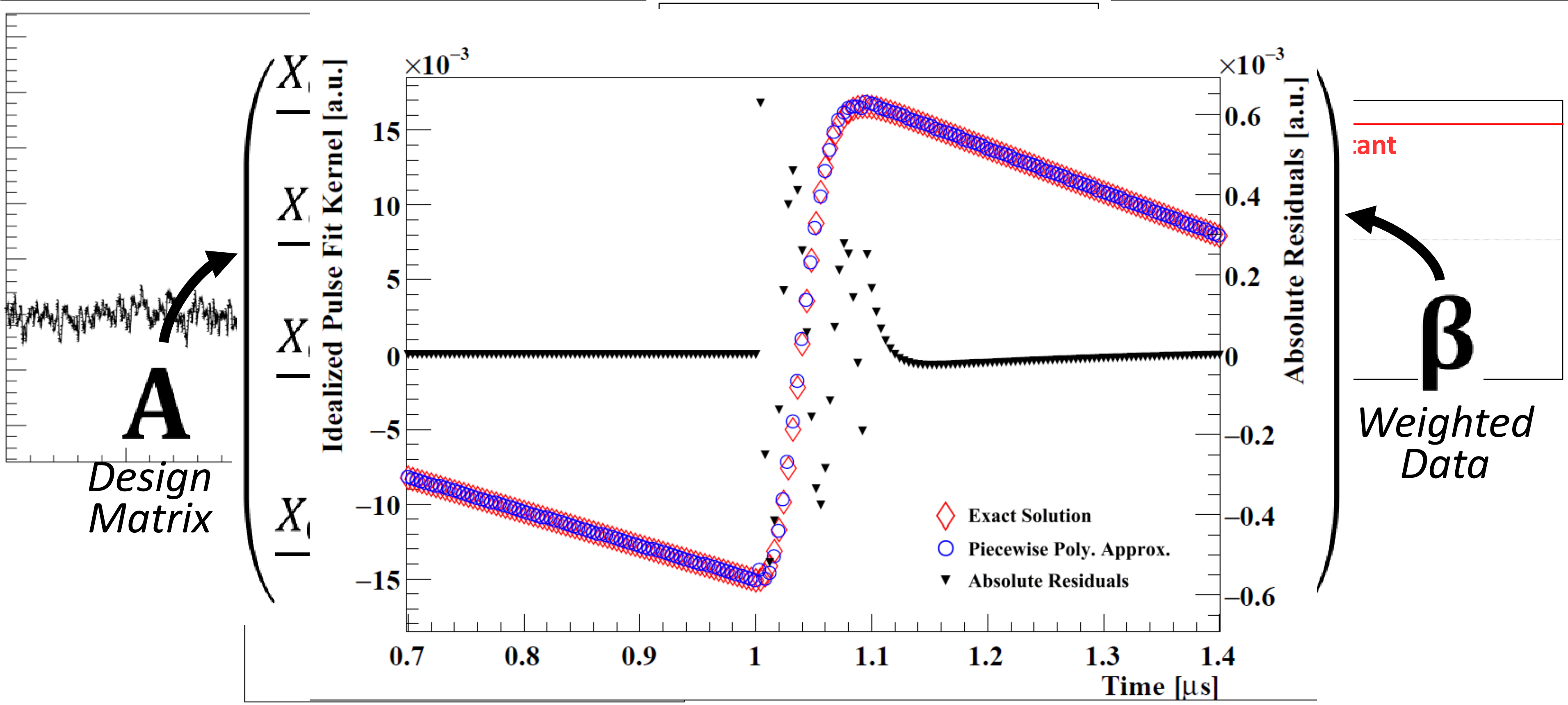


$R_m(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + \cdots$



Aaron Jezghani, University of Kentucky, PhD Thesis

Sliding Least Squares Fitting



Aaron Jezghani, University of Kentucky, PhD Thesis

Generic filter design!