PHYSICS-AWARE SOFT-SENSORS FOR EMBEDDED DIGITAL TWINS

Erik Chinellato[1], Paolo Martin[2], Simone Pierobon[1], Laura Rinaldi[1], Giulio Giusteri[1], Fabio Marcuzzi[1]

[1]Dipartimento di Matematica "Tullio Levi Civita",
Universitá di Padova
marcuzzi@math.unipd.it

[2]SimNumerica s.r.l.
paolo.martin@simnumerica.com

BUILD-IT 2023
October 20th 2023

Università degli Studi di Padova

DIPARTIMENTO MATEMATICA
DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

the innovation will play forever
SIMNUMERICA

Physics-aware soft-sensors

Embedded Digital Twins[1], that is the virtual representations of physical systems that run in embedded systems, are deployed on the edge within the embedded software stack to realize e.g. virtual sensors to enrich available information about physical variables and parameters that cannot be provided by direct physical measurements. These lacking measurements are estimated, at least roughly, by an algorithm that processes the available data, usually called a **soft-sensor**.

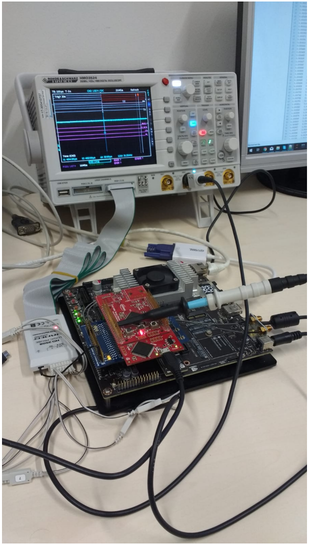
Let us call a ***Physics-aware soft-sensor*** the numerical algorithm that performs an indirect measurement by exploiting a **physico-mathematical model** plus a possible **data-driven extension**, used within an **estimation algorithm**.

In this talk we will present some paradigmatic examples of physics-aware soft sensors for embedded digital twins, formulated with a **scientific machine learning** approach, i.e. where scientific computing methods are blended with machine learning and, in particular, neural networks.
Also, we will point out that complexity of physics-aware soft-sensors depends on:

- ◇ model complexity,
- ◇ interactions with the environment,
- ◇ centrality of measured variables in the virtual measurement process.

[1] Herman Van der Auweraer and Dirk Hartmann. The executable digital twin: merging the digital and the physics worlds, Proceedings ISMA2022, International Conference on Noise and Vibration Engineering, Leuven (B), Sept. 12-14, 2022

Embedded computing issues for physics-aware soft-sensors



- ◇ **Numerical accuracy**: from 8-bit microcontrollers, to 32-bit ARM cores (single precision HW f.p.)
- ◇ **Computing power**: clocks of embedded systems is much lower than PCs.
- ◇ limited **software libraries for computing**, e.g. numerical linear algebra.
- ◇ Modelling for physics-aware soft-sensors with differential equations or sophisticated linear algebra implies to **debug** nontrivial numerical models on a real-time system[1].

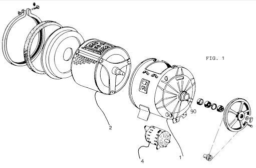
→ for this reason we developed a tool for automatic translation from LaTeX to Python and C, to debug the same code on a PC emulation.

[1] Dessole M., Marcuzzi F., Fully iterative ILU preconditioning of the unsteady Navier–Stokes equations for GPGPU (2019) Computers and Mathematics with Applications, 77 (4), pp. 907 - 927

Physics-aware soft-sensor I: load estimation inside a washing machine through a spinning process

In this example[1], the microprocessor that controls the machine also run a digital twin of the machine mechanical operation, to estimate the weight of the load inserted by the user. The embedded digital twin of each item has been tuned at the end of the production-line (60 pcs/hour), with a least-squares estimation of physical parameters.

Washing Machine



→

at Production plant:
model tuning of the specific Instance model in the Embedded Digital Twin

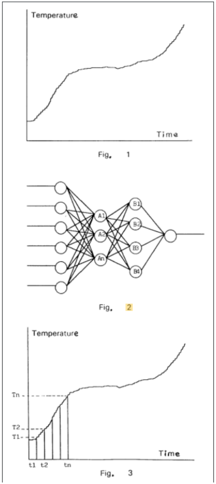
→

at User site: Load Estimation performed by the Embedded Digital Twin

- ⋄ no model reduction (physics-based lumped model);
- ⋄ online least-squares estimation;
- ⋄ 8-bit microcontroller, tenth seconds for computing.

[1] European Patent 95112646.5-2314 "Improvement in a washing machine with automatic determination of the weight of the washload": https://patents.google.com/patent/EP0704568A1/es

Physics-aware soft-sensor II: drying-time estimation from temperature measurements



Here an important process variable (air humidity) could not be measured, by the machine embedded controller. Therefore, the physics-based model could not be used and the data-driven model (a neural network) tried to understand the physics from the available data.

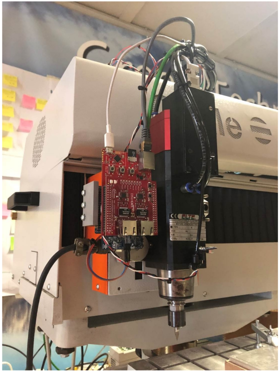
Modeling characteristics of utmost importance:

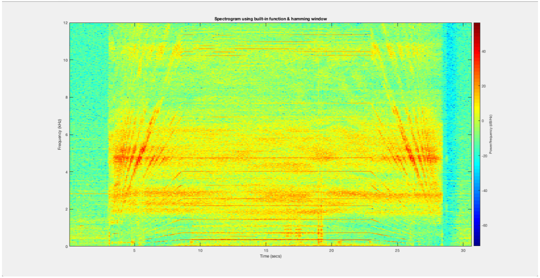
- ◇ no generalizability,
- ◇ trustworthiness hard to reach (implicit nonlinear surface response),
- ◇ high computational efficiency (12-bit arithmetic were sufficient),
- ◇ no self-adaptation,
- ◇ 8-bit microcontroller, almost **one minute to forward run the neural network**.

[1] European Patent Application EP95114450A "Improvement in the arrangement used in a clothes drying apparatus to determine the drying time": https://patents.google.com/patent/EP0707107A1/pt

Physics-aware soft-sensor III: hits detection from audio source-separation

In experimental vibration analysis, using a **microphone** instead of the usual accelerometer, **imposes to separate the acoustic created by the process to be monitored, from that generated by the environment**.





The Deep NMF[1]

Let us consider the standard NMF problem in the form

$$
\begin{aligned}
&\text{minimize} && D_\beta(X, WH) \\
&\text{subject to} && W \in \mathcal{M}_{m\times r}(\mathbb{R}^+) \quad , \quad H \in \mathcal{M}_{r\times n}(\mathbb{R}^+)
\end{aligned}
\tag{0.1}
$$

By **interpreting the iterative update scheme as a neural network**, where $H^{(k+1)}$ is the output of the k-th layer given the input $H^{(k)}$, Deep-NMF tries to address the convergence issue by *untying* the bases across layers.

$$
\begin{tikzcd}
H^{(0)} \arrow[r, "W^{(0)}"] & H^{(1)} \arrow[r, "W^{(1)}"] & \cdots \arrow[r, "W^{(K-2)}"] & H^{(K-1)} \arrow[r, "W^{(K-1)}"] & H^{(K)} \\
X \arrow[r] & {} \arrow[u] \arrow[r] & \cdots \arrow[r] & {} \arrow[u] \arrow[r] & {} \arrow[u]
\end{tikzcd}
$$

Denoting S the clean source spectrogram we want to separate and reconstruct, $\hat{W}_S$ the subdictionary allocated to describe clean frequencies during the training phase and $H_S^{(K)}$ the corresponding rows in the coefficient's matrix, a possible choice for the loss-function $\mathcal{E}$ is

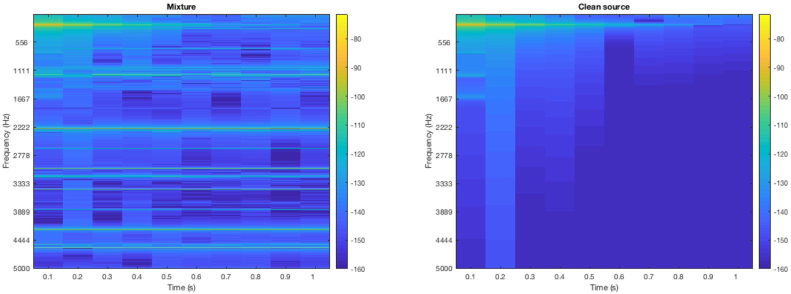
$$
\mathcal{E} = \|S - F \circ X\|_2^2 \tag{0.2}
$$

where $F = \frac{\hat{W}_S H_S^{(K)}}{\hat{W} H^{(K)}}$ is the Wiener filter (the division is elementwise).

[1] Hershey J. R., Le Roux J., Weninger F. - *Deep Unfolding: Model-Based Inspiration of Novel Deep Architectures*, (2014)

Physics-aware soft-sensor III: hits detection from audio source-separation

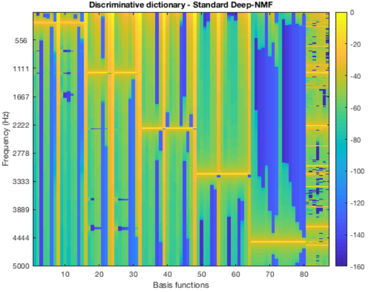
We have imposed a discriminative optimal dictionary and an Hankel structure to the $W^{(i)}$ to recognize physically-characterized hit responses[1]:

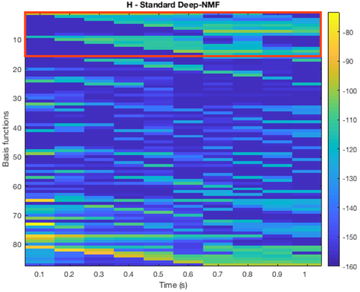


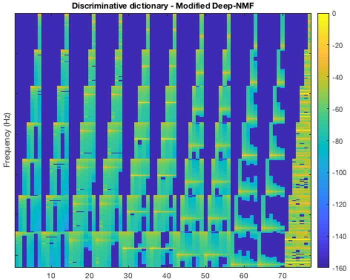
In order to process 1 second of mixture signal and produce the Deep-NMF output matrix H, for the above example around 5.5 million floating point operations are needed. As a consequence, an ARM4 microcontroller (144 MHz, with a HW single-precision floating-point unit) with its $\sim$ 8MFLOPS, can run the complete algorithm in less than 0.7 seconds.

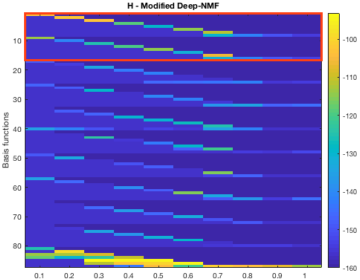
[1] Erik Chinellato and Fabio Marcuzzi. Hits detection in audio mixtures by means of a modified Deep-NMF, submitted, 2023.

Physics-aware soft-sensor III: hits detection from audio source-separation

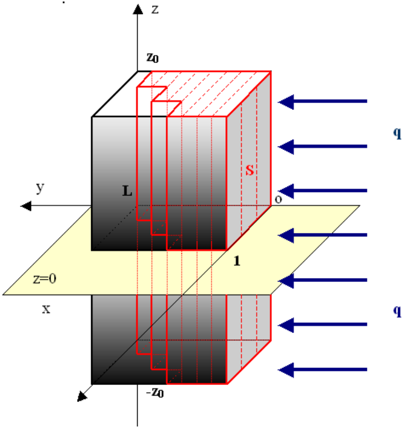








Physics-aware soft-sensor IV: inner voids/material-changes estimation from thermographic inspection



Aim: to estimate the inner material change without solving a nonlinear shape estimation problem.

We demonstrated[1] that there exists a fictitious heat source $f_\vartheta(t,x,y)$

such that for the solution $T_{(f_\vartheta)}$ of

$$
\begin{cases}
\rho C\,\partial_t T_{(f_\vartheta)} = \kappa\,\Delta T_{(f_\vartheta)} + f_\vartheta, & in\ D_c^{(0)} \times [0,t_f] \\
\kappa\,\nabla T_{(f_\vartheta)} \cdot \mathbf{n}_S = q(t), & on\ S \times [0,t_f] \\
\kappa\,\nabla T_{(f_\vartheta)} \cdot \mathbf{n} = 0, & on\ \delta D_c^{(0)}/S \times [0,t_f] \\
T_{(f_\vartheta)}(0,\cdot) = T_0(\cdot), & in\ D_c^{(0)}.
\end{cases}
\tag{0.3}
$$

it holds that $T_{(f_\vartheta)}(t,x,y) = T^{(\vartheta)}(t,x,y)$ for all $(x,y)\in S$ and for all $t\in[0,t_f]$ and $\mathbf{f}_\theta$ **has compact support within the region corresponding to the void**.
Then, our inverse problem becomes to estimate the fictitious heat source $f_\vartheta(t,x,y)$ in (0.3).

[1] Giusteri GG, Marcuzzi F, Rinaldi L. Replacing voids and localized parameter changes with fictitious forcing terms in boundary-value problems, Results in Applied Mathematics, **20**, 2023, 10.1016/j.rinam.2023.100402

Physics-aware soft-sensor IV: inner voids/material-changes estimation from thermographic inspection

In [1] the source term estimator f^k_{mean} be derived from the PDE equation as

$$f^k_{mean} = \frac{1}{N_t} \sum_t \left(\rho C \partial_t \mathbf{e}_{\theta^k} + \kappa \partial_x^2 \mathbf{e}_{\theta^k} + \kappa \partial_y^2 \mathbf{e}_{\theta^k} \right), \tag{0.4}$$

where derivatives are approximated with a suitable finite-difference scheme using the measured values and we assume to accept the approximation $\partial_y^2 \mathbf{e}_{\theta^k} \approx 0$ because it cannot be computed directly.

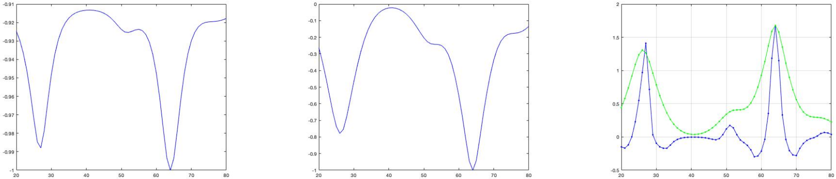


Figure: On the left and on the center, respectively, the prediction error with opposite sign $-\mathbf{e}_{\theta^k}$ at the first iteration of the sparse test problem, whose pattern is shown on the left, with a lower and a bigger value of the conductivity coefficient κ. On the right the same prediction error $\mathbf{e}_{\theta^k}$ with the higher value of κ (light line) together with the external source estimate obtained with (0.4) (dark line).

[1] Dessole M., Marcuzzi F., Accurate detection of hidden material changes as fictitious heat sources, Numerical Heat Transfer, Part B: Fundamentals, 2023, doi 10.1080/10407790.2023.2220905

Physics-aware soft-sensor IV: inner voids/material-changes estimation from thermographic inspection

The fictitious heat source can be estimated from a dynamical system in the so-called *state-space* form:

$$x(k+1) = A \cdot x(k) + B \cdot u(k) + v(k) \quad (0.5)$$
$$y(k) = C \cdot x(k) + D \cdot u(k) + w(k) \quad (0.6)$$

where $\{x(k)\}$ is the state vector, $\{y(k)\}$ is the vector of measured outputs, $\{u(k)\}$ is the vector of inputs (which are supposed known), $\{v(k)\}$ and $\{w(k)\}$ are *model noise* and *measurement noise*, supposed gaussian, with zero mean and covariance matrices Q and R. Then, under certain assumptions, it is possible to apply the Kalman Filter to estimate the state trajectory of the real system. Let us recall here the KF in its one-step version:

$$P(k) = \left[\left(Q(k-1) + A'(k-1)P(k-1)A'(k-1)^T \right)^{-1} + C^T R^{-1} C \right]^{-1} \quad (0.7)$$

$$\delta\hat{x}(k) = -P(k)\ C^T R^{-1} \left[C \left(A'(k-1)\ \hat{x}(k-1) + B\ u(k-1) \right) - \bar{y}(k) \right] \quad (0.8)$$

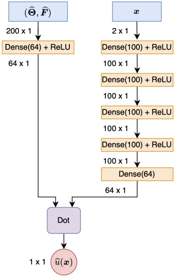
$$\hat{x}(k) = A'(k-1)\ \hat{x}(k-1) + B\ u(k-1) + \delta\hat{x}(k) \quad (0.9)$$

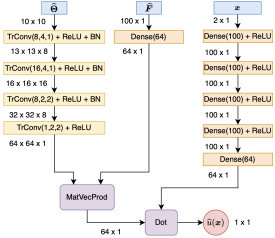
Here the scientific machine learning can learn the covariance matrices and avoid to compute the Kalman gain[1].

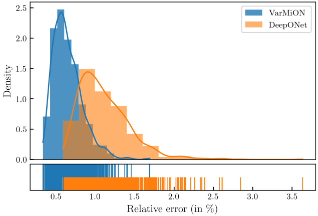
[1] Guy Revach, Nir Shlezinger, Xiaoyong Ni, A.L. Escoriza, Ruud J. G. Van Sloun and Yonina C. Eldar, KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics, IEEE Transactions on Signal Processing, 70, pp.1532-1547, 2022

Physics-aware soft-sensor IV: inner voids/material-changes estimation from thermographic inspection

Surrogate modelling by scientific machine learning: the more interpretable method[1] gets better results in force of an architecture that is more tight to the weak form used in Finite Element models.







An ARM4 microcontroller (144 MHz, with a HW single-precision floating-point unit) with its $\sim$ 8MFLOPS, can compute the predicted temperature in a single-point in space in 0.09 seconds.

[1] Dhruv Patel, Deep Ray, Michael R. A. Abdelmalik, Thomas J. R. Hughes and Assad A. Oberai. Variationally Mimetic Operator Networks, arXiv 2209.12871, 2023.

Thank You for your attention!

marcuzzi@math.unipd.it