Multi-fidelity surrogate modelling for time-series output

Baptiste Kerleguer[1,2], Claire Cannamela[1], Josselin Garnier[2]

MASCOT-NUM PhD Day 2022 | June 7, 2022

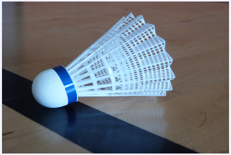
[1] CEA,DAM,DIF, F-91297, Arpajon, France,
[2] Centre de Mathématiques Appliquées, Ecole Polytechnique, 91128 Palaiseau Cedex, France

Outline

Model

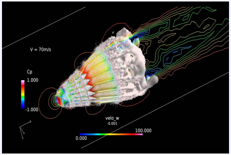
Input parameters x

Physical system



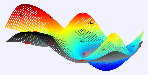
$\approx$

Computer code



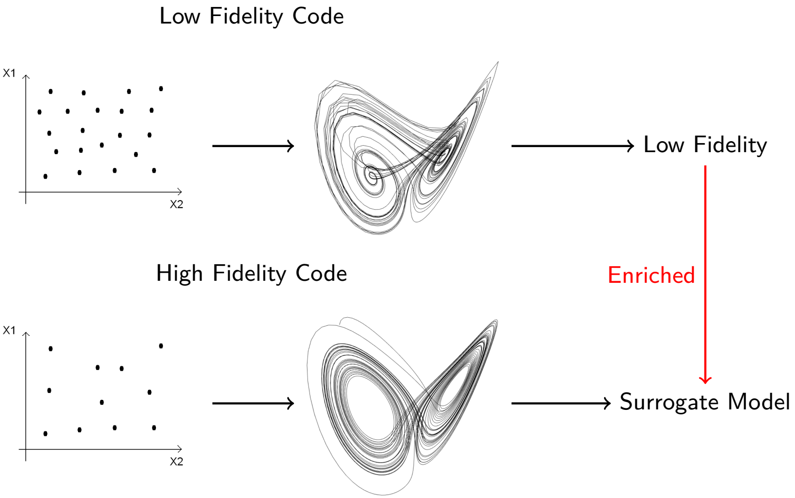
$\approx$

Surrogate model

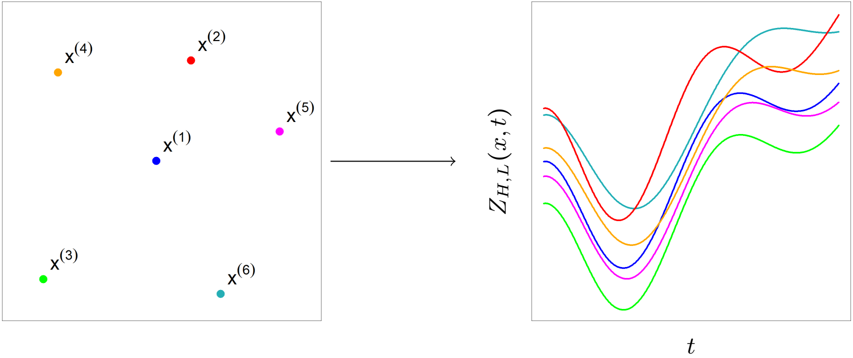


$z_{\text{exp}}(x)$ | $z(x)$ | $Z(x)$

Multi-fidelity surrogate model



Time dependent outputs, definition and use



- The quantity of interest $Z_{H,L}(x,t) : \mathcal{X} \times [0,T] \to \mathbb{R}$ is known at $N_{H,L}$ points of the **designs of experiments** at $\mathcal{D}_{H,L} = \{x^{(1)}, \ldots, x^{(N_{H,L})}\}$ with $N_L > N_H$
- $Z_{H,L}(x,t_u)$ is known for t_u on a regular grid with $u = 1, \cdots, N_t$ and $N_t \gg 1$but only for some $x \in \mathbb{R}^d$.

Gaussian process for time-series outputs - Simple fidelity

The emulator of the code is a Gaussian Process (GP) such that :

$$Z(x, t_u) \sim \mathcal{GP}\left(\mu(x, t_u), k(x, t_u)\right).$$

To build the GP, different methods can be used :

- For specific form (tensorized, sparse, ...) of the covariance function the system is well conditioned, and it is possible to optimize the hyperparameters, see Perrin, G. (2020).
- If the dimension of the outputs can be reduced then independent Gaussian processes can be used to estimate the function on the subspace.

  The dimension reduction methods can be : PCA, ICA, Auto-encoders, ...
- In our method the GP's kernel is supposed to be parametric. The **wavelet transform** is used to simplify the conditioning of the GP. The kernel function is tensorized in t_u and x :

$$k(x, t_u) = k_t(t_u) k_x(x).$$

Wavelet Gaussian process (WGP)

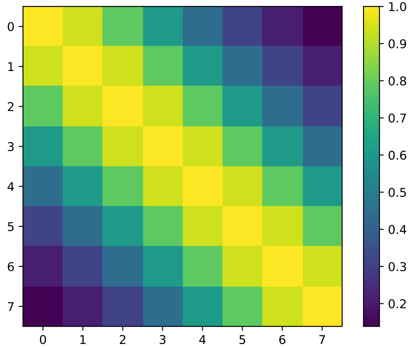
The Gaussian process is modelled by :

$$Z(x,t_u) = \underbrace{A(x,0)\phi(t_u)}_{\text{Scale function}} + \underbrace{\sum_{i=1}^{N_t-1} A(x,i)\psi_i(t_u)}_{\text{Wavelet decomposition}}$$

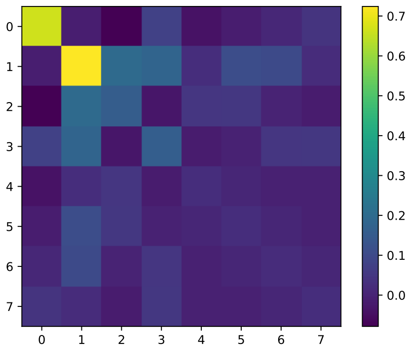
1. The outputs are decomposed on a wavelet basis.
   - A is the GP in the wavelet domain with mean and covariance with an analytical expression obtained form the kernel of Z.
   - Realizations of A are observed from the data.
2. The number of points must be reduced for the Gaussian conditioning.
   - We are not able to work on the $N_x N_t \times N_x N_t$ covariance matrix of the system.
   - Points (x,i) in the experimental are points that maximize : $(\sigma(A(x,i)) - a(x,i))^2 + \sigma(A(x,i))^2$.
3. GP regression in wavelet domain.
4. Return to the time domain, $Z(x,t_u)$ is expressed as a function of A.

Covariance in the time domain and the wavelet domain

Covariance of $Z(x,t)$ (time domain)



Covariance in A (wavelet domain)



The covariance is **more concentrated** in the wavelet case than in time case.

This reduces the number of points with **minimal loss of information**.

WGP conditioning

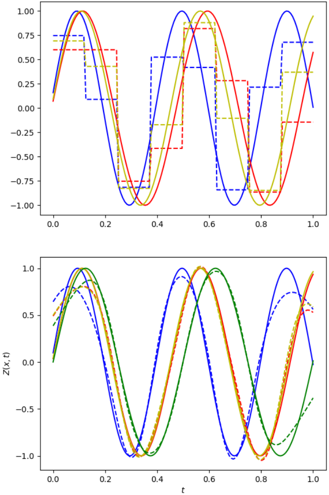
$$Z(x,t) \xrightarrow{\text{Wavelet transform}} A(x,i) \xrightarrow{\text{Truncation}}$$

Covariance function

$$\mathrm{Cov}(A(x,i), A(x',i')) \xrightarrow{\text{WGP}}$$

$N_x \times N_t \approx 2000$ points in the learning set $N_x = 15$, $N_t = 128$.

For each method only 100 points are used.



WGP summary

- The covariance is supposed to be tensorized. The analytical formulation of the covariance is :

$$\mathrm{Cov}(A(\cdot,j,k),A(\cdot,j',k')) = \int e^{-i\xi(2^j(k+\frac{1}{2})-2^{j'}(k'+\frac{1}{2}))}\hat{\psi}(2^j\xi)\overline{\hat{\psi}(2^j\xi)}\hat{F}(\xi)d\xi,$$

with $\hat{F}(\xi)$ the Fourier transform of the covariance kernel and $\hat{\psi}(\xi)$ the Fourier transform of the wavelet function. $i = (j,k)$ j is the scale coefficient and k the position. $\hat{F}(\xi)$ is obtained thanks to the tensorized kernel of Z.

  - Analytical expressions for Haar wavelet function and Matérn kernel are available.

- Sparse representation of the covariance in the Wavelet domain allows to have an efficient conditioning of the WGP.

- Same properties as for GP :
  - GP based optimization
  - Active learning
  - ...

Multi-fidelity AR(1) Gaussian process regression - scalar output

- The quantities of interest : $a_H(x) \in \mathbb{R}$, respectively $a_L(x)$ with $x \in \mathbb{R}^d$.
- Hypothesis : The emulator is a **GP** $(A_H(x), A_L(x))$ with realizaitions $(a_H(x), a_L(x))$.
- Autoregressive CoKriging model :

$$A_H(x) = \rho(x) A_L(x) + \delta(x),$$

where $\delta(x)$ GP independent of $A_L(x)$ and $\rho(x)$ adjustment linear form.

- Prediction : when the hyper-parameters of the model are known,

$$[A_H(x)|\text{data}] \sim \mathcal{GP}(m_{A_H}(x), \sigma^2_{A_H}(x)),$$

the quantities $m_{A_H}(x)$ and $\sigma^2_{A_H}(x)$ have **analytical expressions**.

- These expressions come from Le Gratiet, L., & Garnier, J. (2014).

Illustration in 1D of Multi-fidelity

The low-fidelity code (in red) $z_L(x) = 0.5(6x-2)^2\sin(12x-4) + 10(x-0.5) - 5$.

The high-fidelity code (in black) $z_H(x) = 2z_L(x) - 20x + 20$.

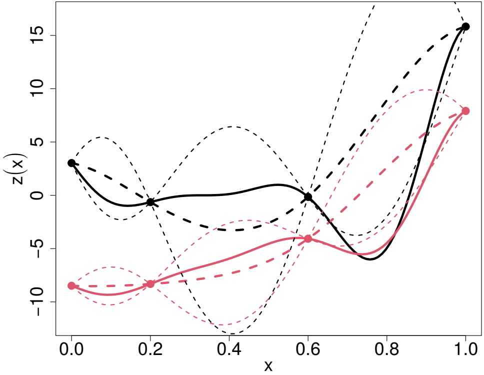


Illustration in 1D of Multi-fidelity

The low-fidelity code (in red) $z_L(x) = 0.5(6x-2)^2\sin(12x-4)+10(x-0.5)-5$.
The high-fidelity code (in black) $z_H(x) = 2z_L(x) - 20x + 20$.

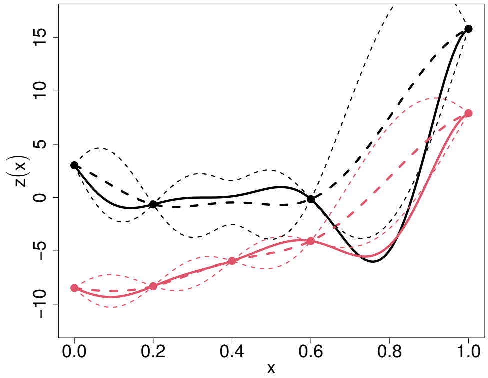


Illustration in 1D of Multi-fidelity

The low-fidelity code (in red) $z_L(x) = 0.5(6x-2)^2\sin(12x-4) + 10(x-0.5) - 5$.

The high-fidelity code (in black) $z_H(x) = 2z_L(x) - 20x + 20$.

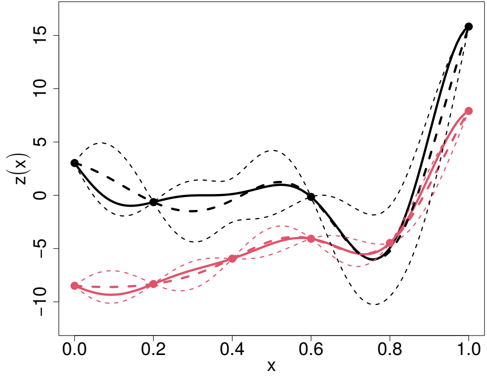
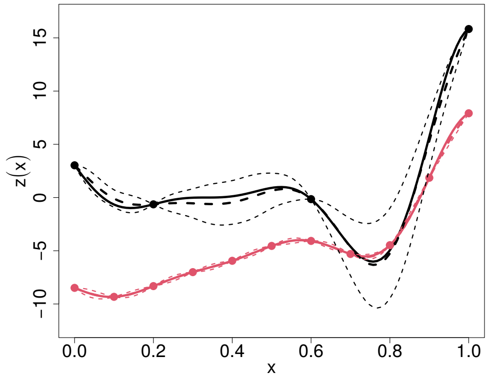


Illustration in 1D of Multi-fidelity

The low-fidelity code (in red) $z_L(x) = 0.5(6x-2)^2\sin(12x-4)+10(x-0.5)-5$.

The high-fidelity code (in black) $z_H(x) = 2z_L(x) - 20x + 20$.



Multi-fidelity Wavelet Gaussian process - time series output

Let assume that (Z_H, Z_L) is a Gaussian process of the form :

$$Z_L(x,t) = A_L(x,0)\phi(t) + \sum_{i=1}^{N_t-1} A_L(x,i)\psi_i(t),$$

$$Z_H(x,t) = A_H(x,0)\phi(t) + \sum_{i=1}^{N_t-1} A_H(x,i)\psi_i(t),$$

then the AR(1) hypothesis is done on (Z_H, Z_L) :

$$Z_H(x,t_u) = \tilde{\rho}(x) Z_L(x,t_u) + \tilde{\delta}(x,t_u).$$

The hypothesis is also true for (A_H, A_L) :

$$A_H(x,i) = \rho(x) A_L(x,i) + \delta(x,i).$$

- We assume that the covariance of (Z_H, Z_L) is of a parametric form. Then the covariance of (A_H, A_L) is of a parametric form.

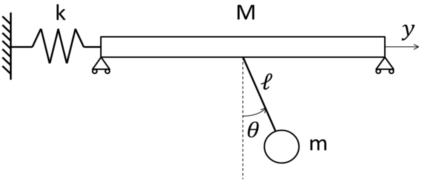
Multi-fidelity WGP regression summary

- The WGP can effortlessly extend to multi-fidelity framework.

1. Wavelet transform for high- and low-fidelity data.
2. Point selection for high-and low-fidelity surrogate model.
3. Regression for low-fidelity using WGP.
4. Regression of high-fidelity using the recursive form
5. Inverse Wavelet transform

An example : The double pendulum

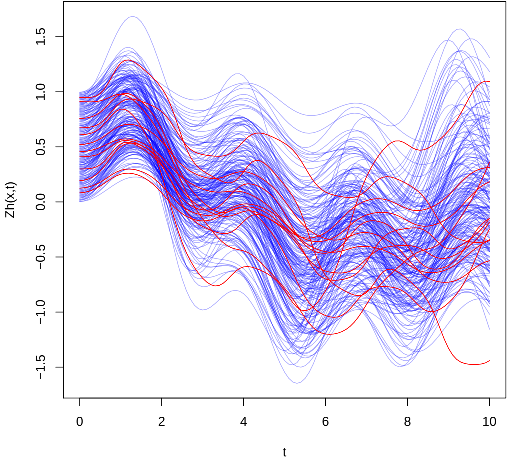
We use the outputs of a calculation code that models the system with two levels of fidelity :



	inputs
k	spring stiffness
M	mobile mass
l	pendulum length
y	mobile position
θ	pendulum angle
m	pendulum mass
	output
z	pendulum position

The difference between high and low fidelity is that the for low fidelity we assume that the angle θ is small.

Examples of Outputs



M	$[3;8]$
k	$[0.1;2.1]$
θ_0	$\left[\frac{\pi}{4};\frac{\pi}{2}\right]$
$\dot{\theta}_0$	$\left[0;\frac{1}{6}\right]$
y_0	$[0;0.2]$
$\dot{y}_0$	0
l	2
m	0.5
g	9.81

TABLE – Range of variation of the different system parameters.

For $N_H = 10$ (red), $N_L = 100$ (blue) and $N_t = 101$ (for wavelet transform $N_t = 64$)

The CVB method

- The emulator of the low- and high-fidelity code output is :

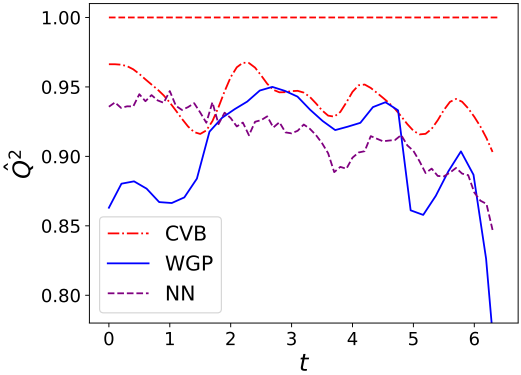
$$Z_L(x,t_u) = \sum_{i=1}^{N} B_{i,L}(x)\Gamma_i(t_u) + Z_L^{\perp}(x,t_u)$$

$$Z_H(x,t_u) = \underbrace{\sum_{i=1}^{N} \overbrace{B_{i,H}(x)}^{\text{coefficients}}\,\overbrace{\Gamma_i(t_u)}^{\text{basis}}}_{\text{dimension reduction}} + \underbrace{Z_H^{\perp}(x,t_u)}_{\text{orthogonal part}}$$

- $(B_{i,H}(x), B_{i,L}(x))$ is a Gaussian process.
- The orthogonal part $Z_H^{\perp}(x,t_u)$ is a Gaussian process with tensorized covariance.
- The basis $\Gamma_i(t_u)$ is obtained by SVD on low-fidelity data.
- We get mean and variance $\mathbb{E}[Z_H(x,t_u)|\text{Observations}]$ and $\mathbb{V}[Z_H(x,t_u)|\text{Observations}]$.
- $N < N_t$ is chosen by cross validation.

This methods is proposed in Kerleguer, B. (2021).

The result in Q^2 :



- NN is a Neural Network method adapted to this example.
- The results are better with CVB because the function is almost stationary.
- WGP is more expensive than NN and CVB method.

Conclusion & Perspectives

- Conclusion :
  - Prediction and uncertainty quantification for the model.
  - The computation cost remains reasonable.
  - Same properties as for GP regression.
- Perspectives :
  - Fast implementation of the Covariance for different kernels.
  - Use Deep GP for multi-fidelity surrogate.
  - Other approaches selecting conditioning points.

References

- Kennedy, M. C., & O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika, 87(1), 1-13.*
- Le Gratiet, L., & Garnier, J. (2014). Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification, 4(5).*
- Perrin, G. (2020). Adaptive calibration of a computer code with time-series output.*Reliability engineering & system safety, 196, 106728.*
- Kerleguer, B. (2021). Multi-Fidelity Surrogate Modeling for Time-Series Outputs. *arXiv preprint arXiv :2109.11374.*

Thank you