Incorporating prior knowledge into Gaussian Process regression: with applications to aerospace engineering

Ankit Chiplunkar

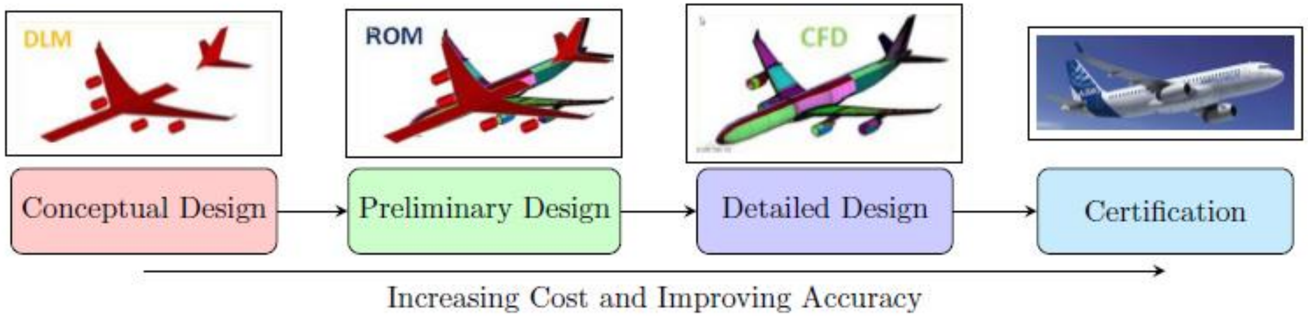
Directors: Joseph MORLIER, Emmanuel RACHELSON
Industrial Supervisor: Michele COLOMBO

Rapporteurs: Nicolas GAYTON et Alexander FORRESTER

Examinateurs: Rodolphe Le RICHE et Nathalie BARTOLI

Motivation and Objectives

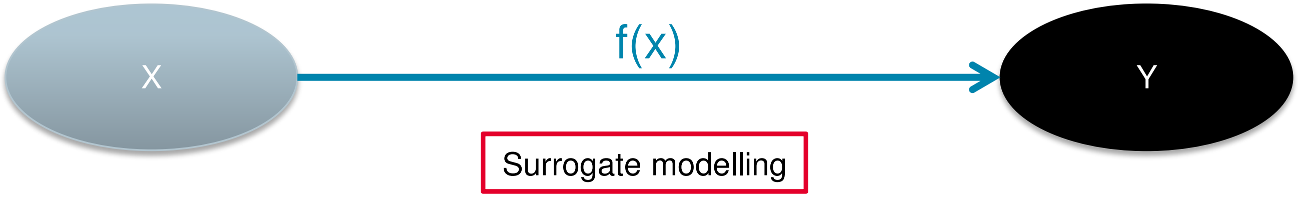
Motivation



Time	Few seconds	Few minutes	Few hours	Few days
Cost	N/A	N/A	10k euros	Millions euros

Find a quicker, cheaper and accurate representation

Objective of the PhD



- Engineering design
    - Bottleneck → Data cost
    - Advantage → Prior information

<u>Incorporate prior information into surrogate models</u>

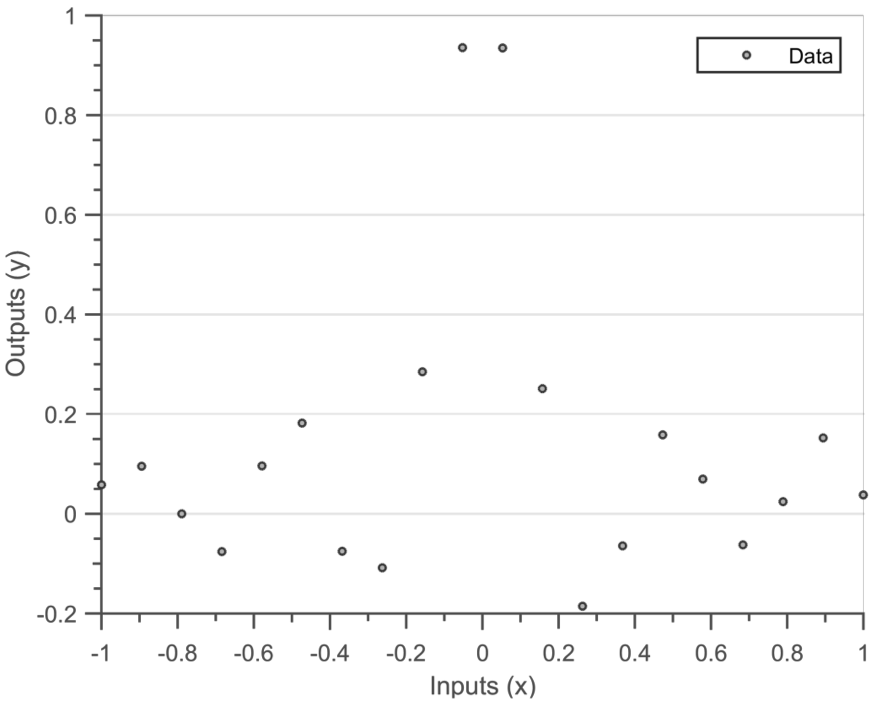
Contents

Gaussian Process Regression

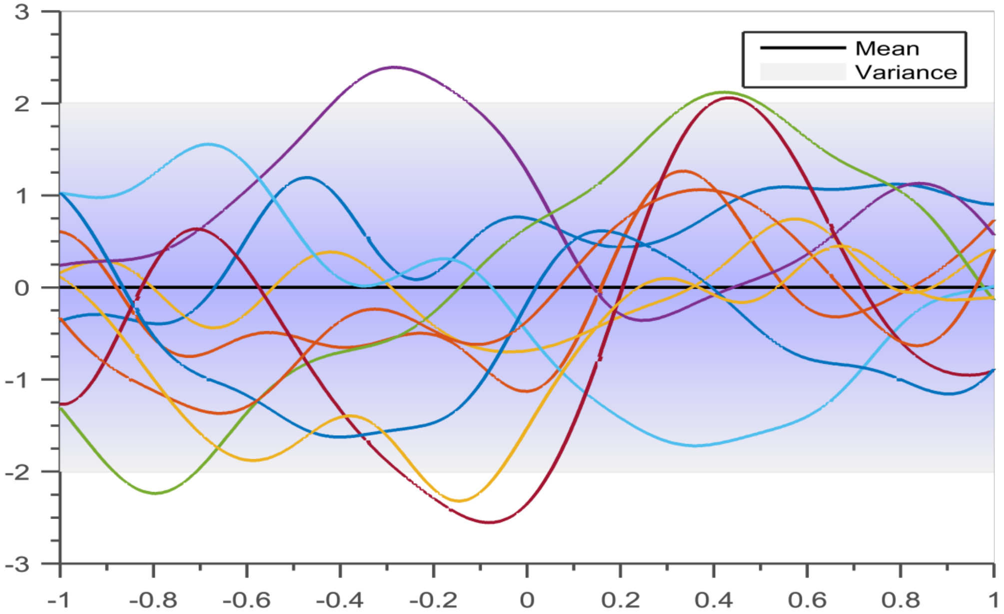
Regression

- Objective:

  Find → f: $R^{D_{inputs}} \rightarrow R^{D_{outputs}}$



Gaussian Process



Prior

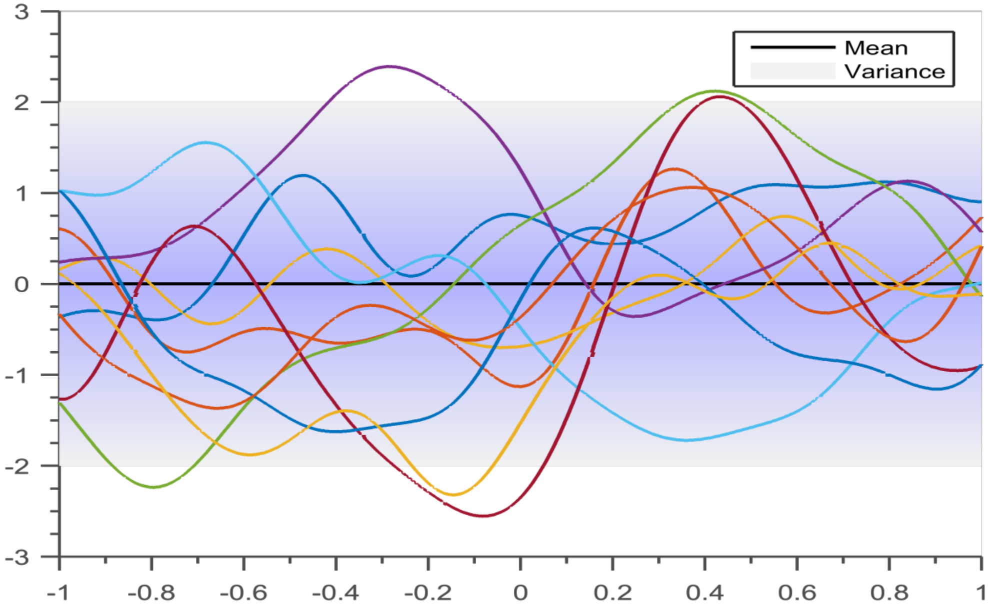
$$p(f|M) = GP(0, K(\theta))$$

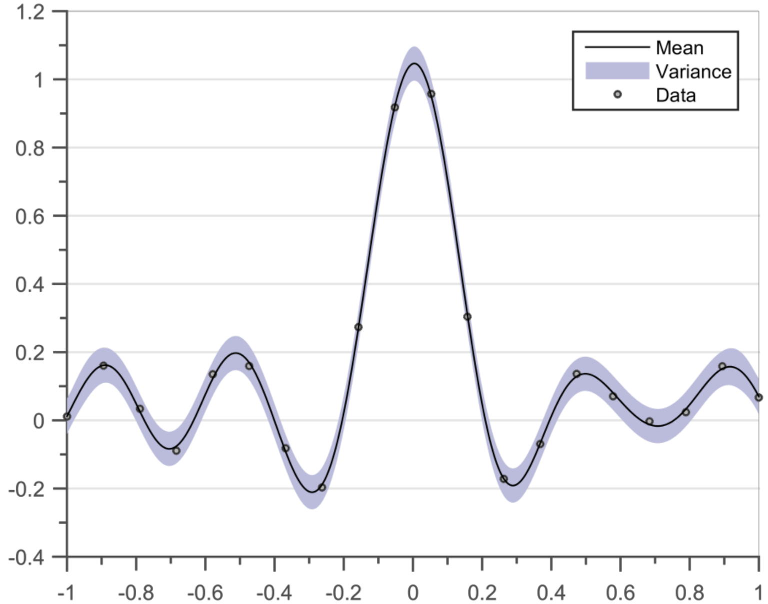
$$k(x, x', \theta) = {\theta_1}^2 \exp\left(-\frac{(x - x')^2}{2{\theta_2}^2}\right)$$

References:

- **[G. Matheron. 1962] → Kriging**
- **[C. Rasmussen et. al. 2005] → Gaussian Process**

Gaussian Process Regression (GPR)





Prior

$$p(f|M) = GP(0, K(\theta))$$

$$k(x, x', \theta) = {\theta_1}^2 \exp(-\frac{(x-x')^2}{2{\theta_2}^2})$$

Posterior

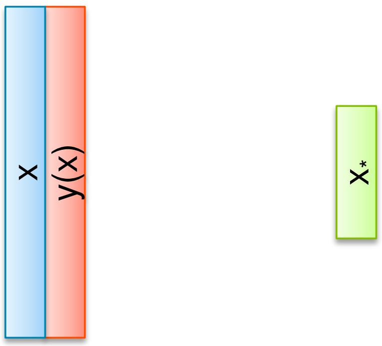
$$m(x_*) = K_*[K + {\sigma_n}^2 I]^{-1} y$$

$$var(x_*, x_*') = K_{**} + \sigma_n^2 I - K_*^T [K + {\sigma_n}^2 I]^{-1} K_*$$

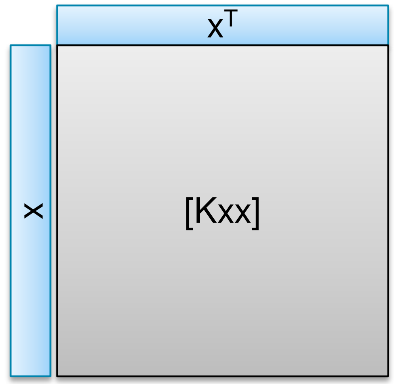
References:

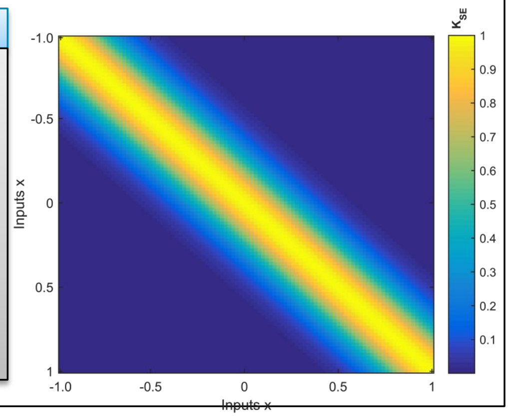
- [G. Matheron. 1962] → Kriging
- [C. Rasmussen et. al. 2005] → Gaussian Process

Matrix view of Gaussian Process

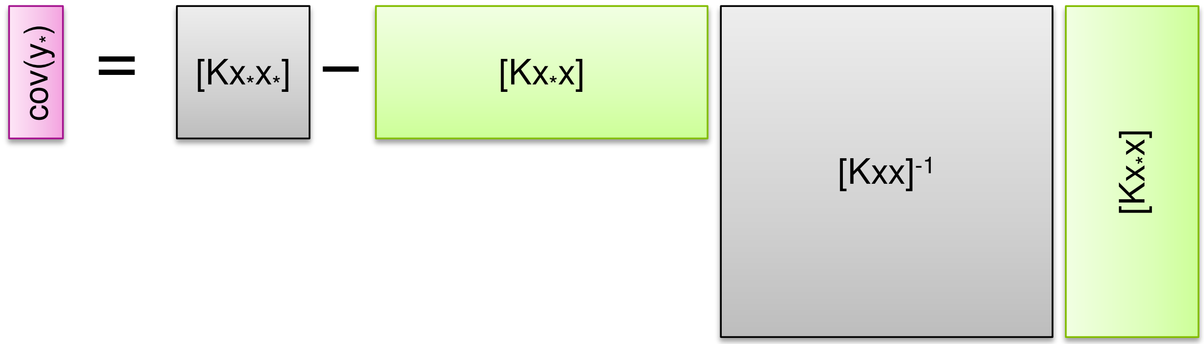


$$k(x,x') = \theta_1{}^2 \exp\left(-\frac{(x-x')^2}{2\theta_2{}^2}\right) =$$





$$m(x_*) = K_*[Kxx]^{-1}y$$

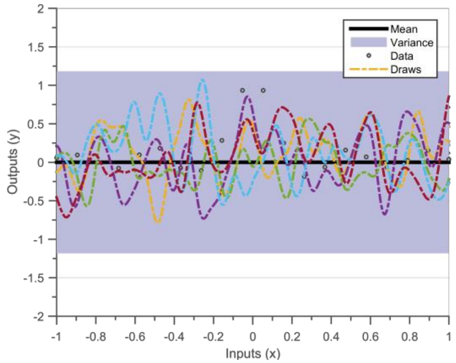


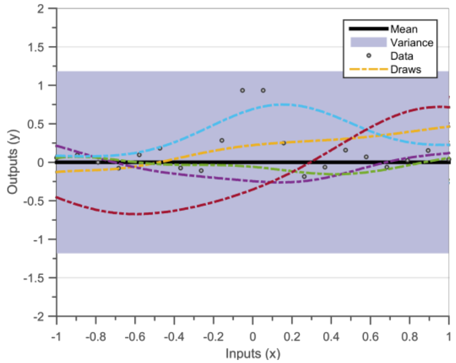
$$var(x_*, x_*') = K_{**} - K_*^T[Kxx]^{-1}K_*$$

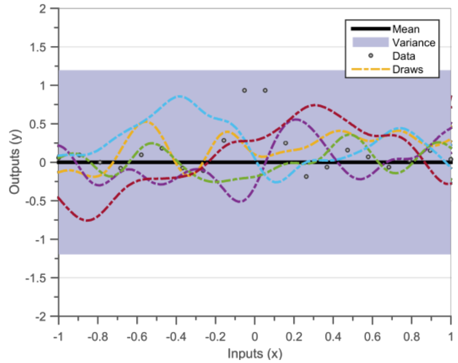
Optimizing marginal likelihood

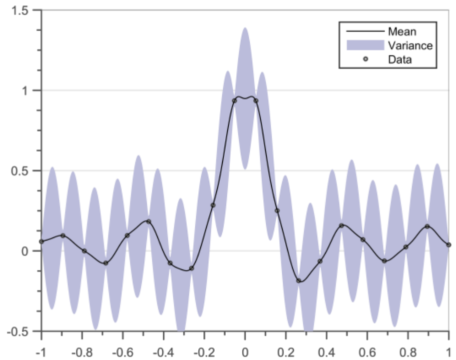
$$\text{ML} = log\big(p(y|X,\theta)\big) = -\frac{1}{2}\boldsymbol{y}^{T}\boldsymbol{K}(\boldsymbol{\theta})^{-1}\boldsymbol{y} - \frac{1}{2}\boldsymbol{log}|\boldsymbol{K}(\boldsymbol{\theta})| - \frac{n}{2}\boldsymbol{log}(2\boldsymbol{\pi})$$

- It is a combination of **data-fit term**, a **complexity penalty** term and a **normalization term**

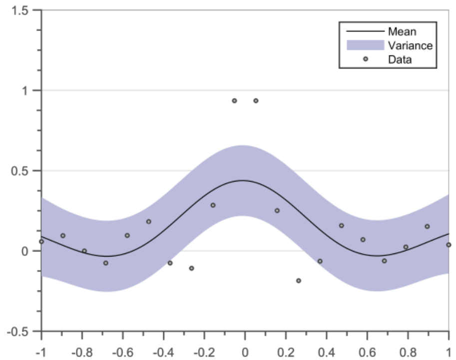




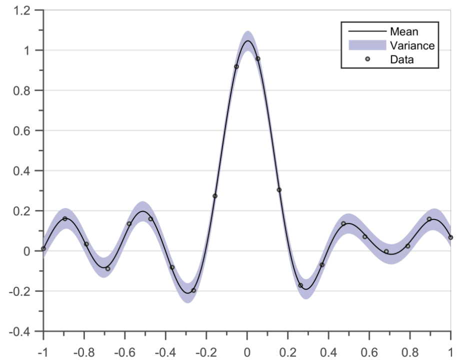




ML = -8.2



ML = -35.3

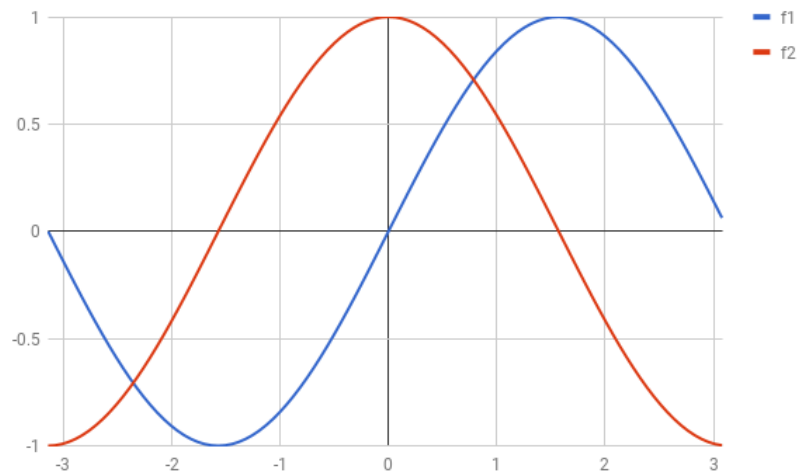


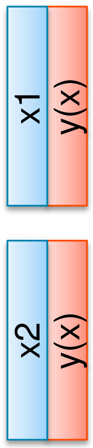
ML = 6,04

Adding information of multiple outputs

Multi-task Gaussian Process

- Case 1: When the outputs are non-correlated

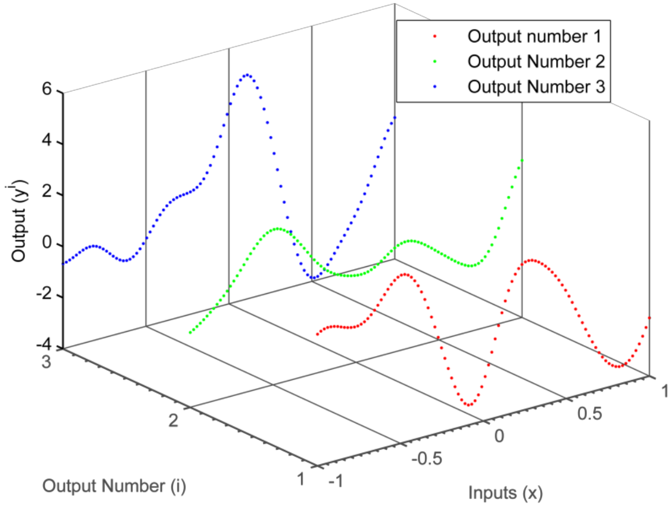


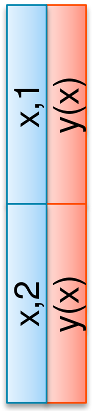


- Define prior independently → $p(f_1) = GP(0, K_1)$ and $p(f_2) = GP(0, K_2)$
- Maximize ML
- Calculate mean and variance

Multi-Task Gaussian Process

Case 2: When the outputs are related but the relation is unknown





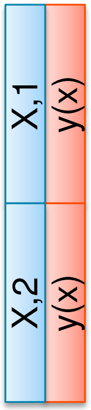
- Joint prior:

$$PR\binom{f_1}{f_2} = GP\left(\begin{matrix}0\\0\end{matrix}\middle|\begin{matrix}K_{11} & K_{12}\\K_{21} & K_{22}\end{matrix}\right)$$

- Maximize marginal likelihood
- Calculate mean and variance

References:
- **[Michael Osborne. Thesis, 2010]**

Matrix view of Multi task Gaussian Process

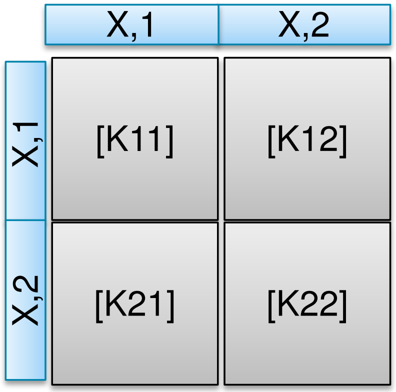


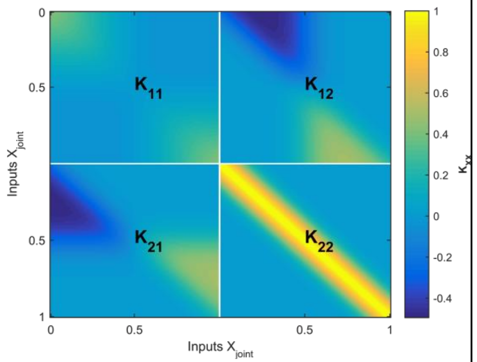
$$k_{input}(x,x') = \theta_1{}^2 exp\left(-\frac{(x-x')^2}{2\theta_2{}^2}\right)$$

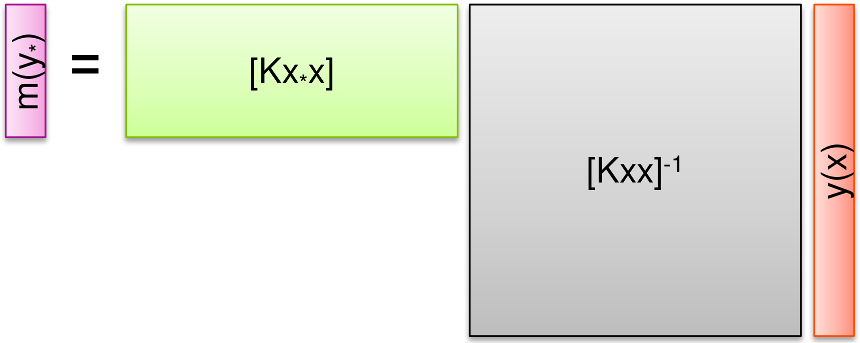
$$K = k_{output} k_{input}$$

$$K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$$

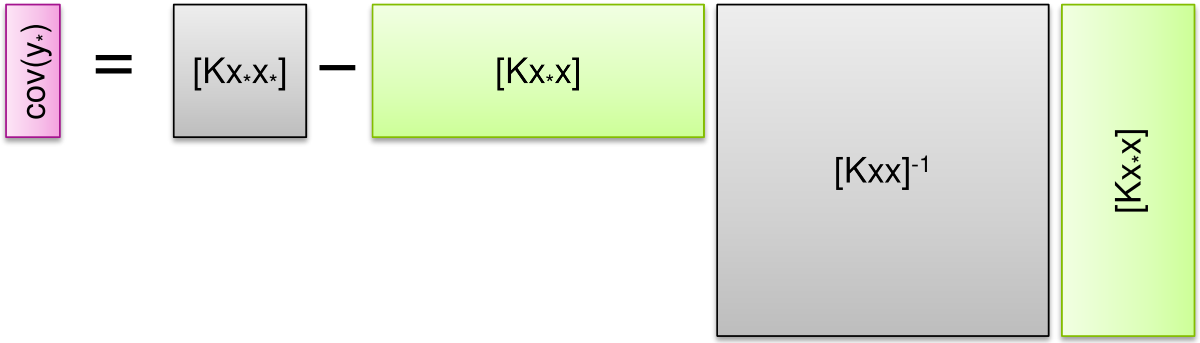
=







$$m(x_*) = K_*[Kxx]^{-1}y$$



$$var(x_*,x_*') = K_{**} - K_*^T[Kxx]^{-1}K_*$$

Multi-task Gaussian Process

Case 3: When the outputs are related by a known relation/function

Given: $f_1 = g(f_2, x)$

Affine property: Iff g is a linear operator such as

$$g(a+b) = g(a) + g(b)$$

Then:

$$p\binom{f_1}{f_2} = GP\left(\begin{matrix}0\\0\end{matrix}\middle|\begin{matrix}g(g(K_{22},x_2),x_1) & g(K_{22},x_2)\\ g(K_{22},x_1) & K_{22}\end{matrix}\right)$$

Example: Gradient enhanced kriging

[AIAA, Chiplunkar 2016]

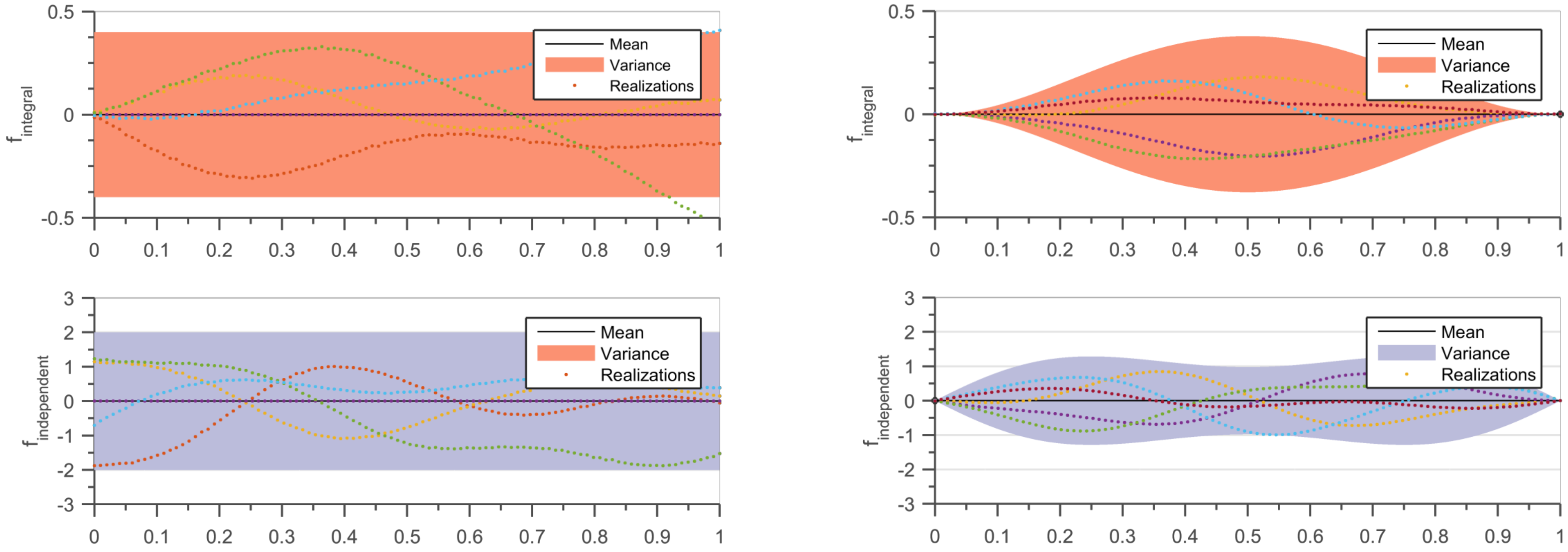
References:
- **[S. Särkkä. et. al ICANN 2011] [D. Ginsbourger, et. al arXiv, 2013] → Linear processes**
- **[Emil M. Constantinescu et. al. SIAM 2013] → Non-linear processes**

Integral relation

$$f_{integral} = \int_0^x f_{independent}$$

$$f_{integral}\Big|_{x=0} = 0$$

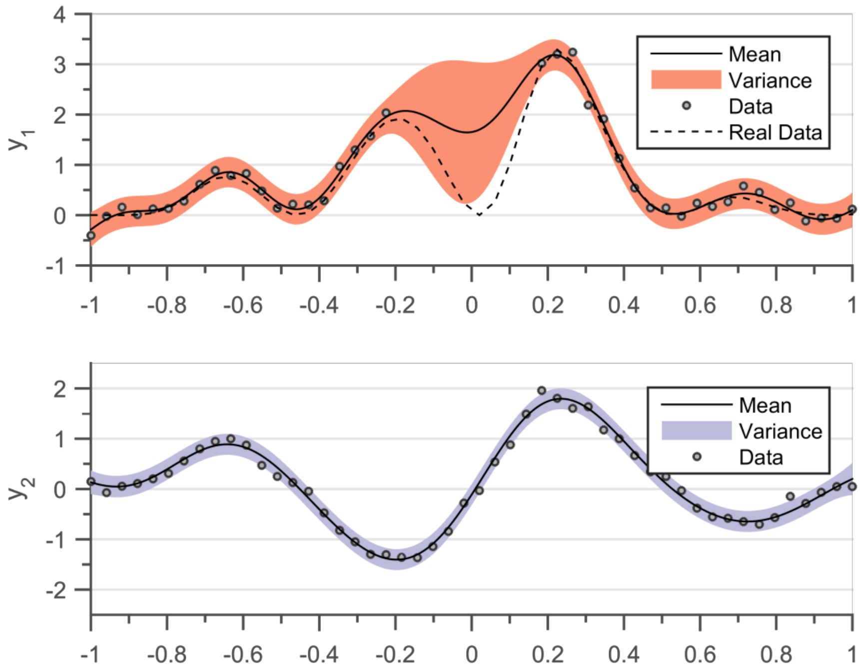
$$f_{integral}\Big|_{x=1} = 0$$



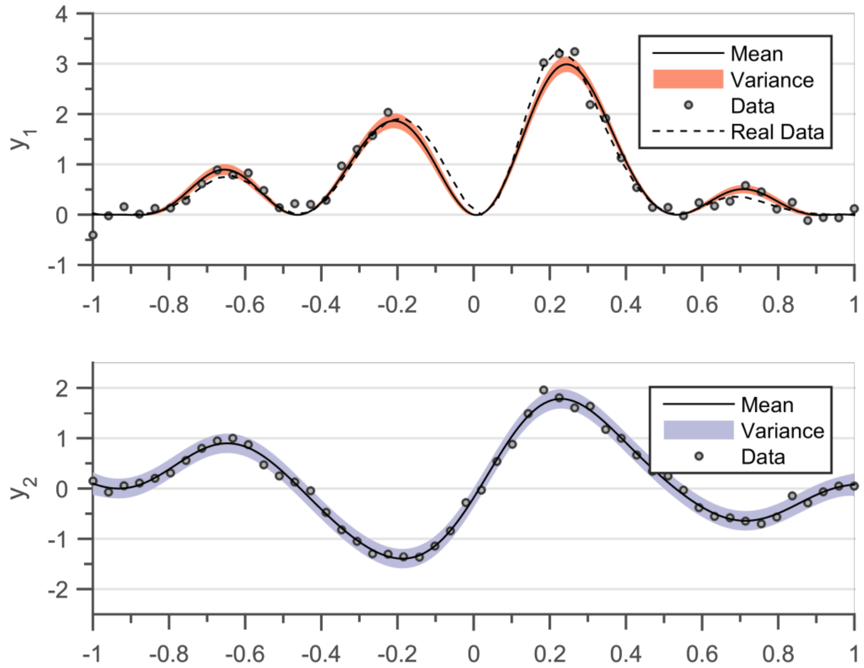
Simulated Data

Sensor malfunction

$$y_1 = (y_2)^2$$



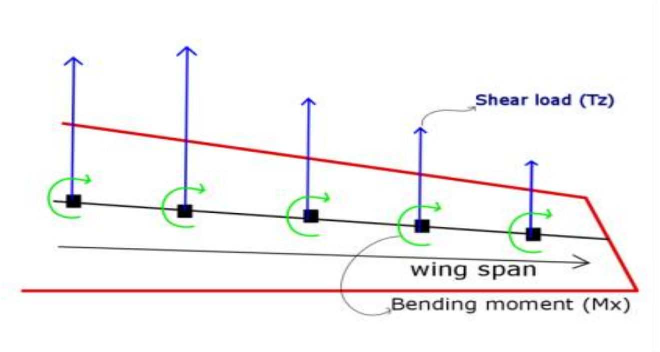
Independent GPs



Related GPs

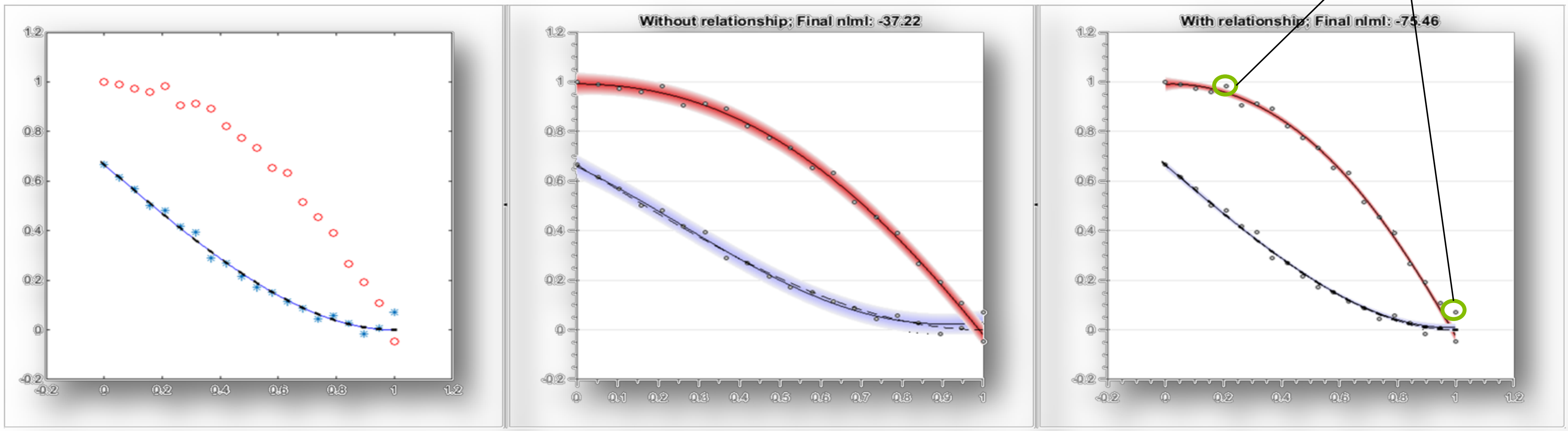
Simulated Data

Detecting faulty sensors



$$Mx = \int_{\eta}^{\eta_{edge}} (x - \eta)\, Tz\, dx$$

Wrong data point

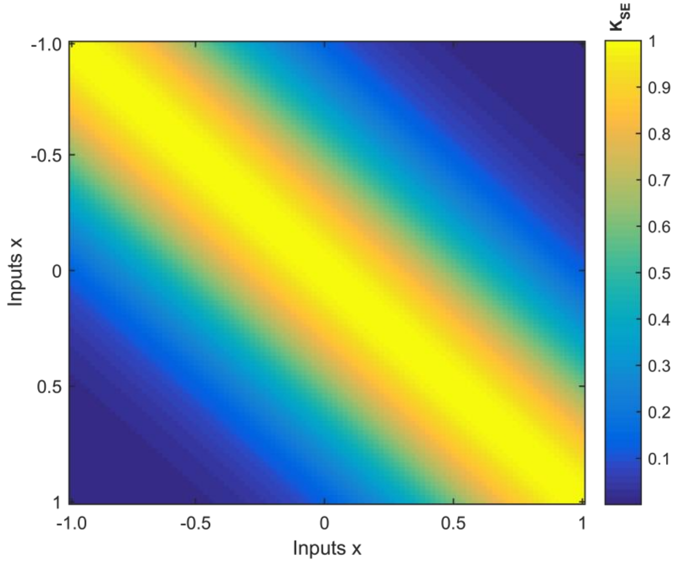


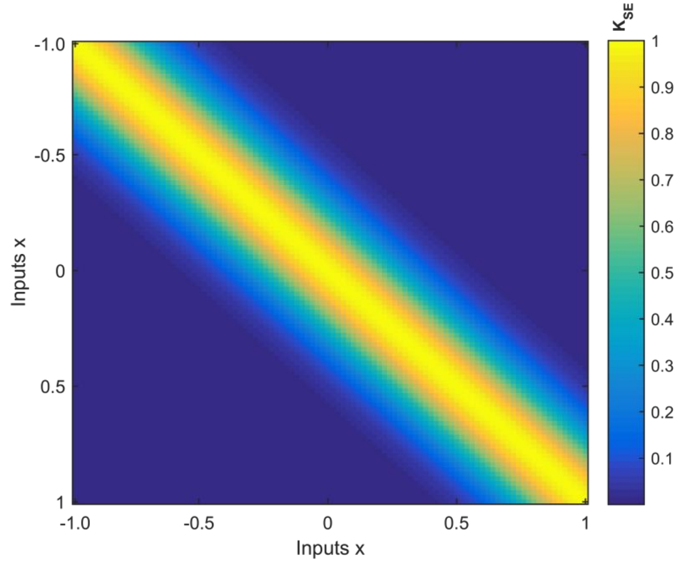
η η η

Simulated Data

Scaling solutions

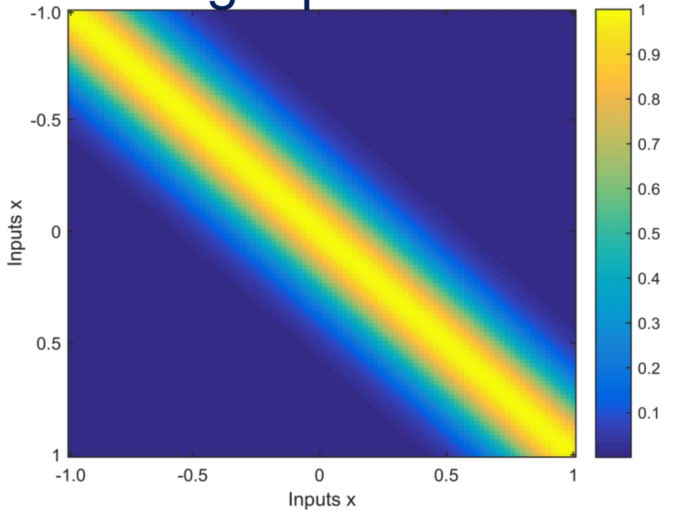
Learning	Storage	Mean	Variance
$O(n^3)$	$O(n^2)$	$O(n)$	$O(n^2)$



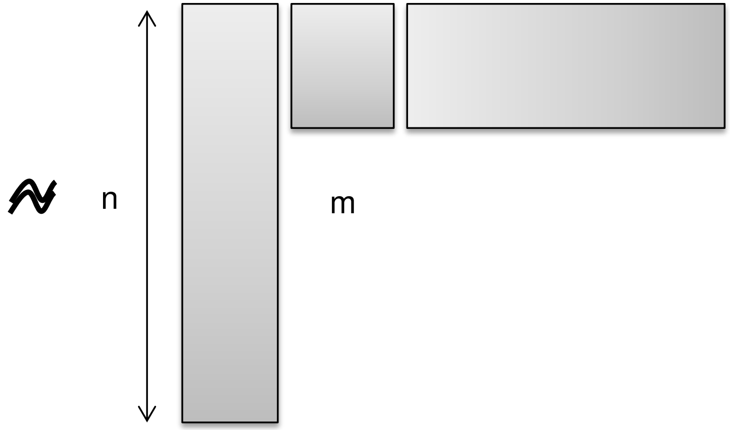


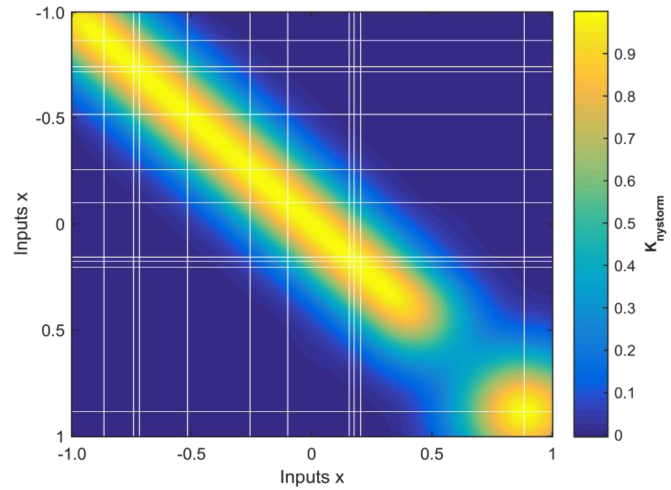
- Bottleneck → Inversion and Storage of matrix
- Infrastructure → HPC cluster

Inducing inputs



K_{xx}





$Kxx_{approximate}$

$$\boxed{m(x_*) = K_{*approximate}\left[Kxx_{approximate}\right]^{-1} y}$$

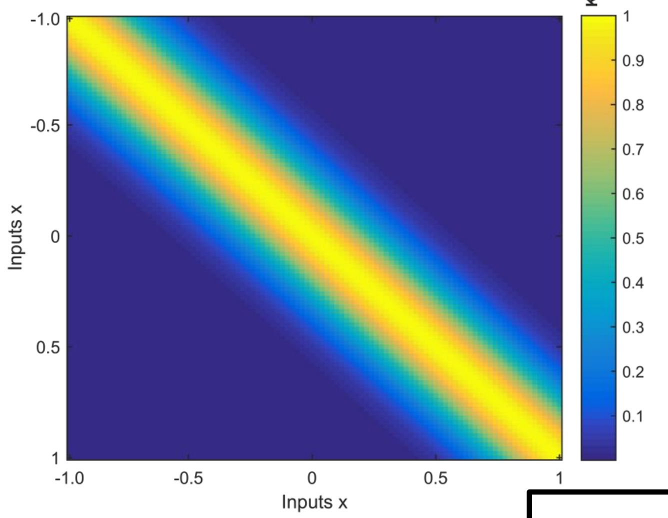
$$\boxed{var(x_*, x'_*) = K_{**approximate} - K^T_{*approximate}\left[Kxx_{approximate}\right]^{-1} K_{*approximate}}$$

$$\boxed{log\big(p(y|X,\theta)\big) = -\frac{1}{2} y^T K_{approximate}{}^{-1} - \frac{1}{2} log\left|K_{approximate}\right| - \frac{n}{2}\log(2\pi)}$$

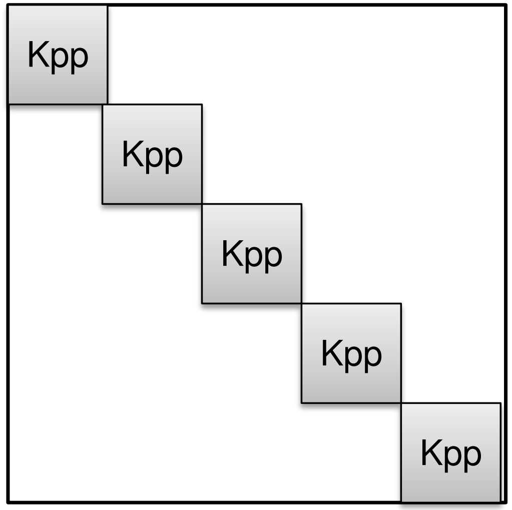
References:

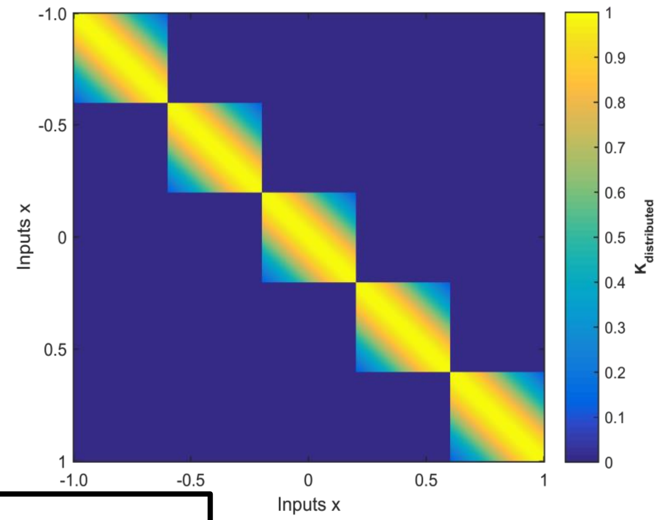
- **[C Williams et al. NIPS 2001] → Inducing inputs**
- **[Michalis K. Titsias. AISTATS 2009] → Variational inference**

Distributed Gaussian Process



$\approx$





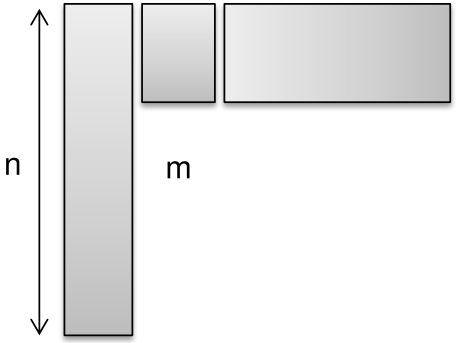
$$m(y_*) = K_{x_*x_*}^{-2} \sum_k \beta_k \, Cov_k(y_*)^{-2} \, m_k(y_*)$$

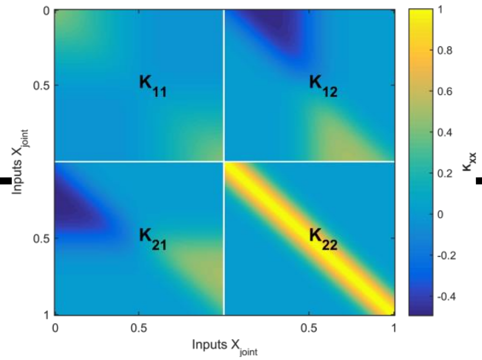
$$\beta_k = \log\left(K_{x_*x_*}{}^{-2}\right) - \log\left(Cov_k(y)^{-2}\right)$$

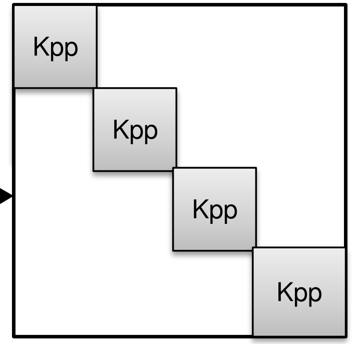
References:

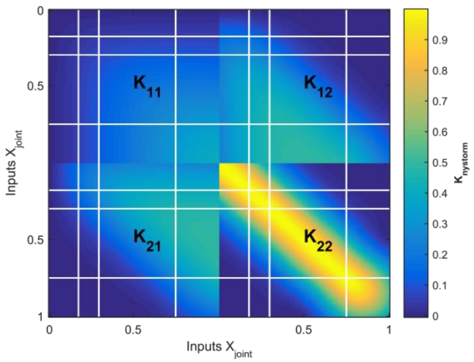
- **[M Deisenroth et al. 2015]**

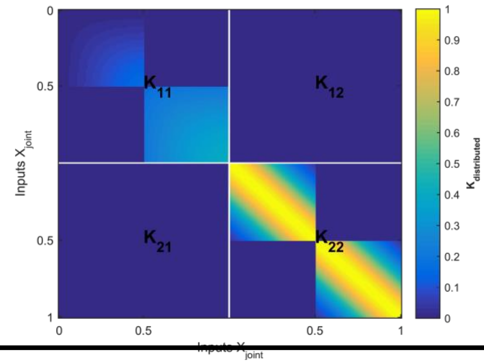
Scaling solutions to Multi-task GP











$$m(x_*) = K_{*approximate}\left[Kxx_{approximate}\right]^{-1} y$$

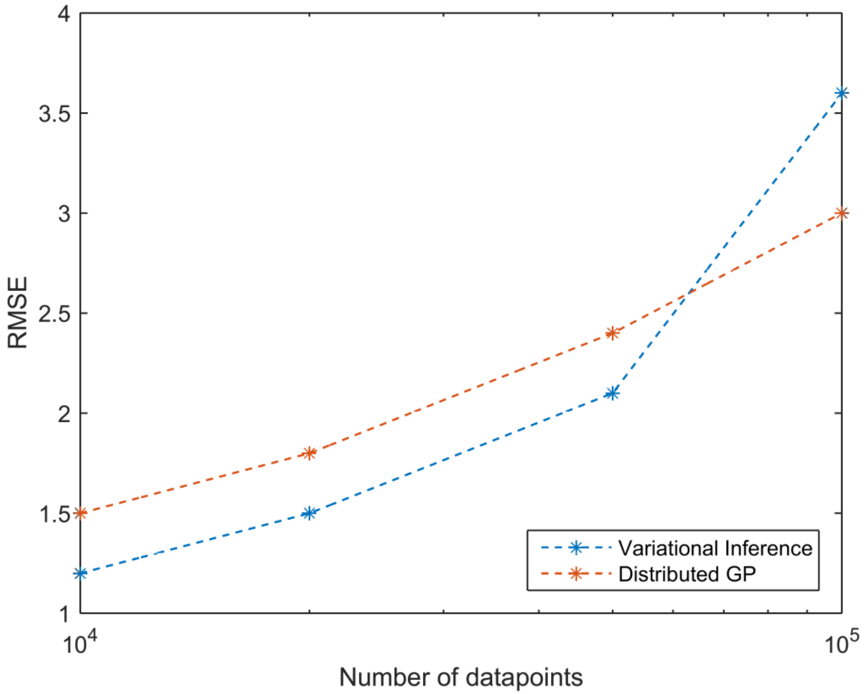
$$var(x_*, x_*') = K_{**approximate} - K_{*approximate}^{T}\left[Kxx_{approximate}\right]^{-1} K_{*approximate}$$

$$log(p(y|X,\theta)) = -\frac{1}{2} y^T K_{approximate}{}^{-1} - \frac{1}{2} log\left|K_{approximate}\right| - \frac{n}{2}\log(2\pi)$$

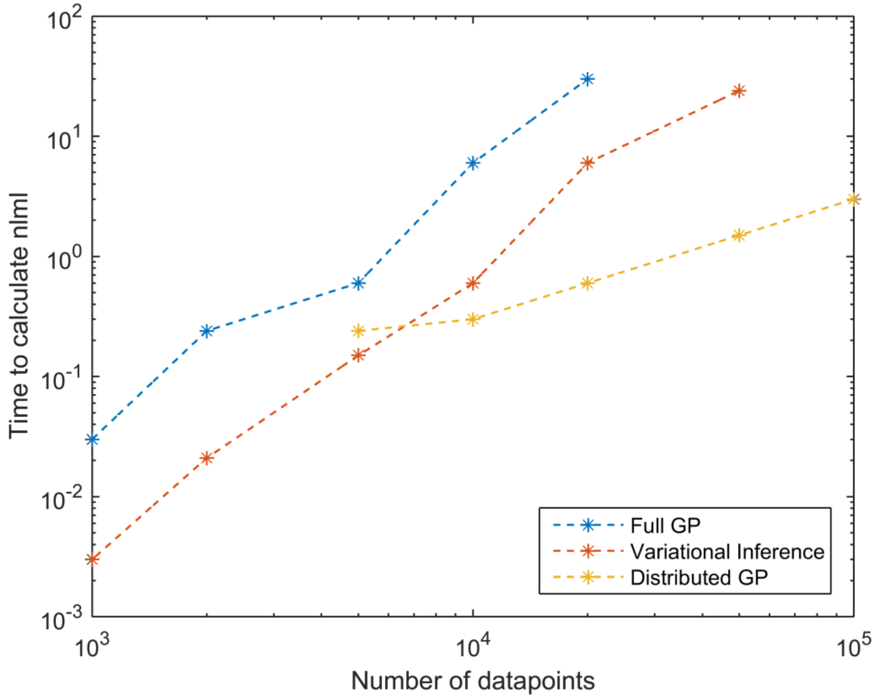
$$m(y_*) = Kx_*x_*^{-2} \sum_k \beta_k \, Cov_k(y_*) \, m_k(y_*)$$

Comparison between two approximation methods

$$f_1 = \frac{df_2}{dx}$$



Error

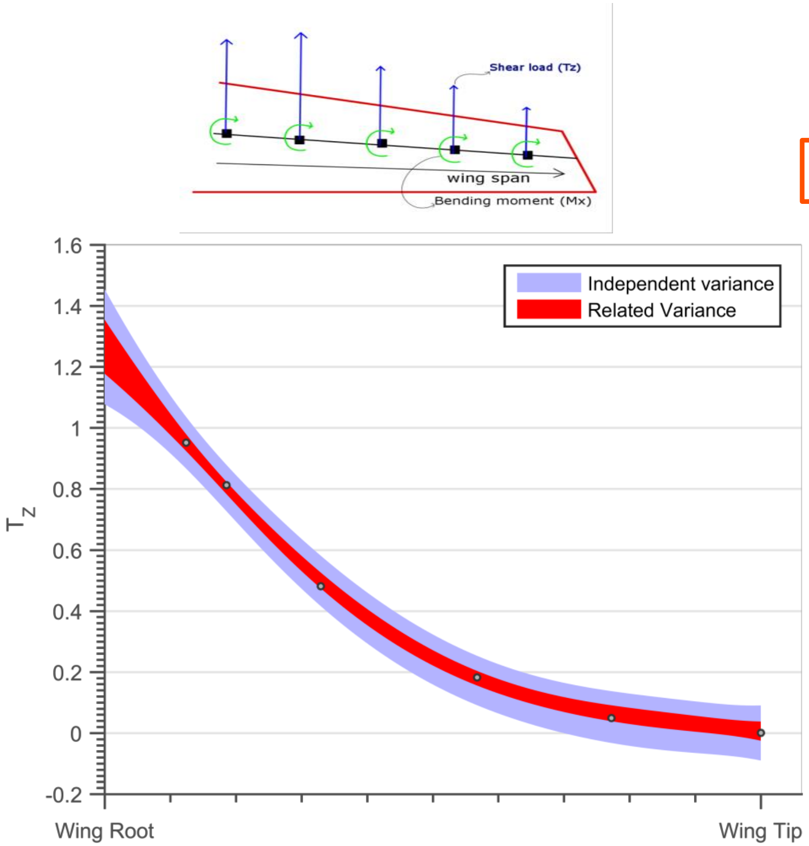


Time

Simulated Data

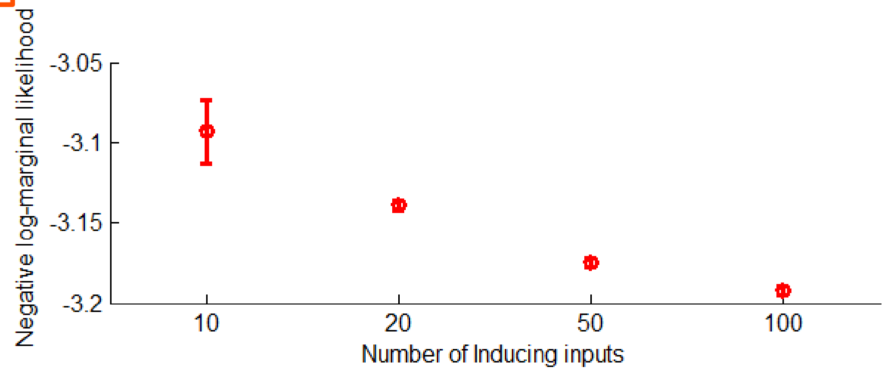
[Chiplunkar et al. ICPRAM, 2016]
[Chiplunkar et al., LNCS, 2017]

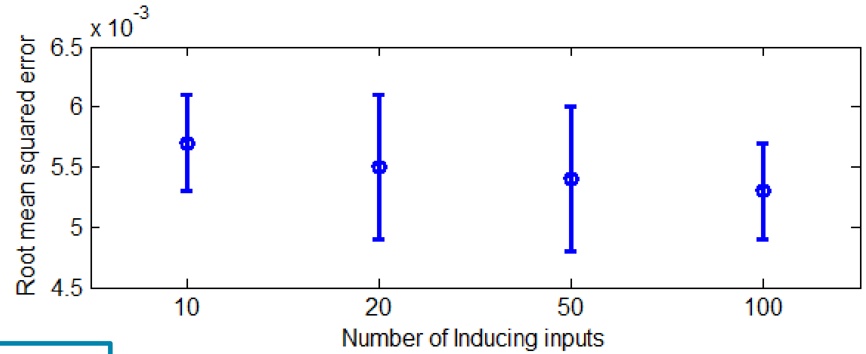
Scaling solutions to Multi-task GP



N = 8800

$$Mx = \int_{\eta}^{\eta_{edge}} (x - \eta)\, Tz\, dx$$





Experimental Data (flight loads)

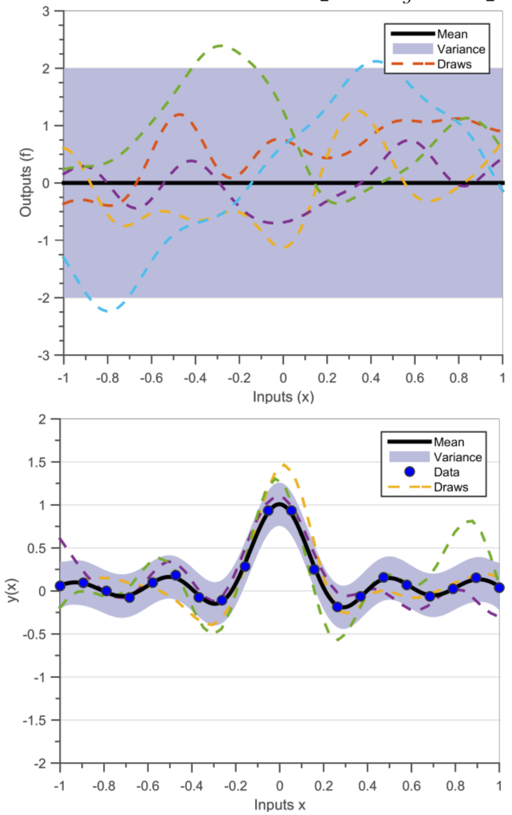
Adding information of shape

- Effect of kernels
- Detecting onset of non-linearity
- Interpolation of shock

Effect of different kernels

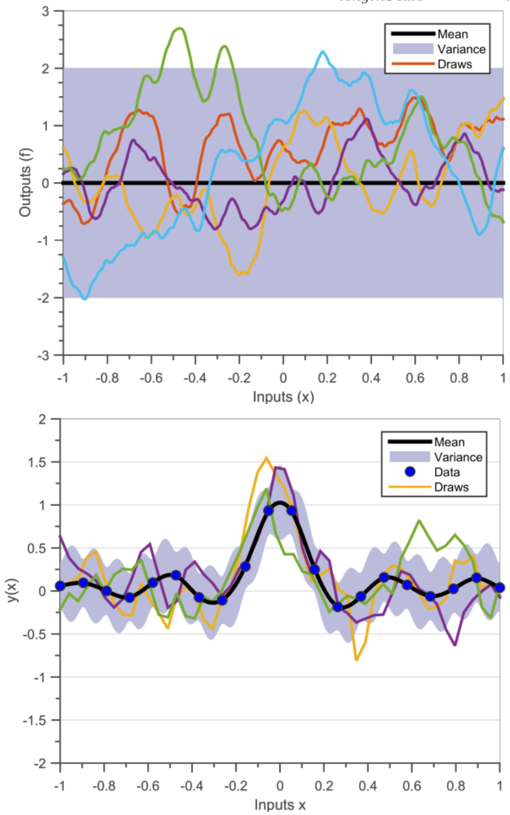
SE kernel (infinitely differentiable)

$$k_{SE}(d,\theta) = \theta^2_{amplitude} exp\left[-\frac{d^2}{2\theta^2_{lengthScale}}\right]$$



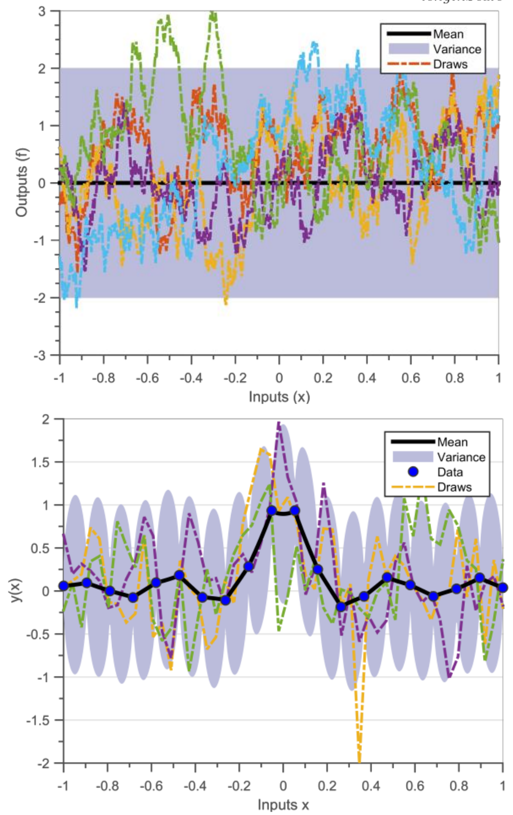
Once differentiable kernel

$$k_{Matern}(\nu = 3/2, d, \theta) = \theta^2_{amplitude}(1+\frac{\sqrt{3}d}{\theta_{lengthScale}})exp[-\frac{\sqrt{3}d}{\theta_{lengthScale}}]$$



Brownian kernel

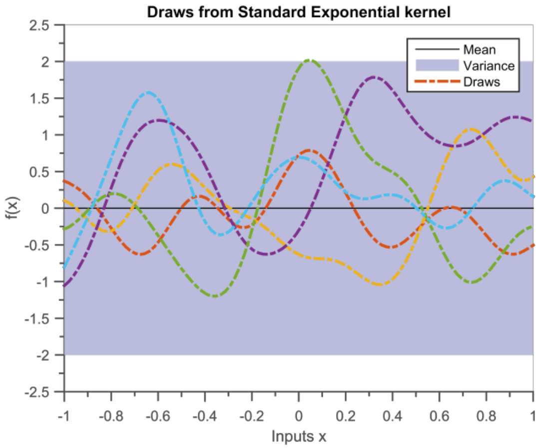
$$k_{Matern}(\nu = 1/2, d, \theta) = \theta^2_{amplitude}exp[-\frac{d}{\theta_{lengthScale}}]$$



Basic Kernel Functions

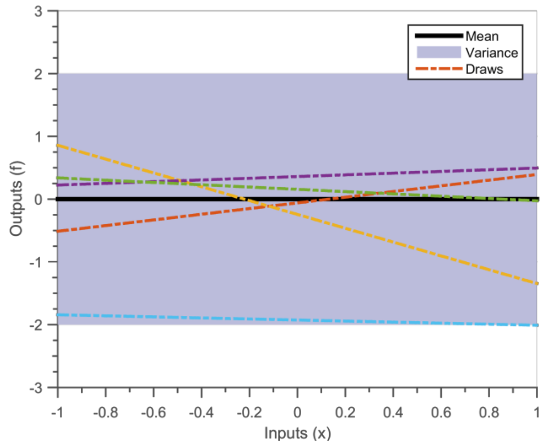
SE kernel (infinitely differentiable)

$$k(x, x') = \sigma^2 \exp(-\frac{(x - x')^2}{l^2})$$



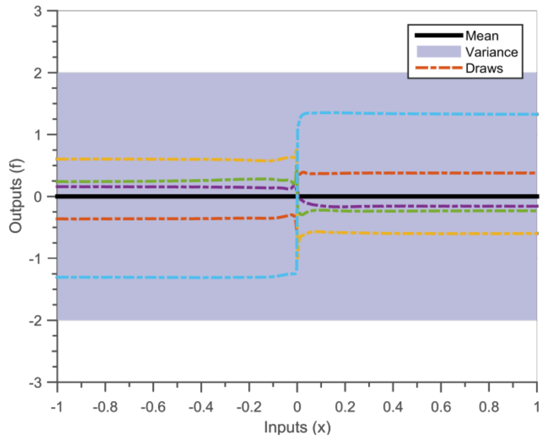
Linear kernel (Linear functions)

$$k(x, x') = {\sigma_b}^2 + {\sigma_v}^2 (x - c)(x' - c)$$



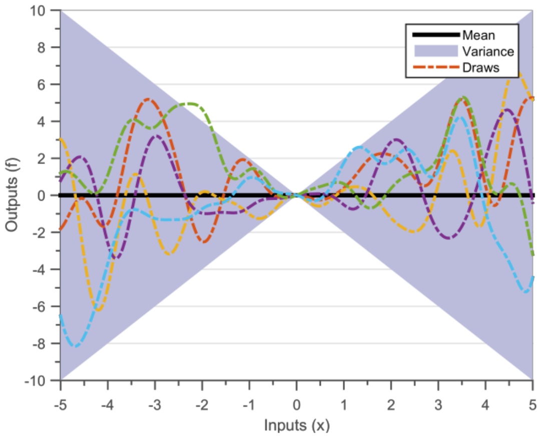
Neural Network kernel (discontinous function)

$$k(x, x') = {\sigma_1}^2 \frac{2}{\pi} \sin^{-1} \left(\frac{2x\theta_2 x'}{\sqrt{(1 + 2x\theta_2 x)(1 + 2x'\theta_2 x')}} \right)$$

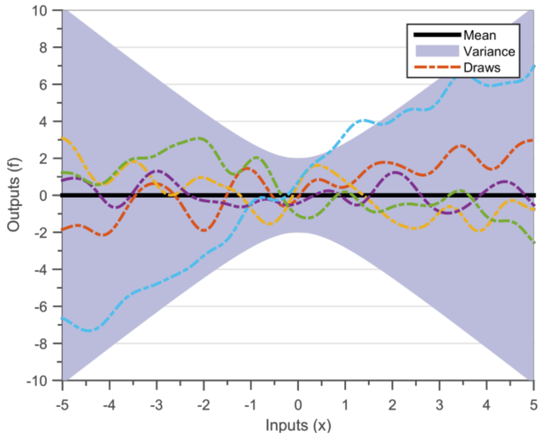


Combining different kernels

Linear (times) SE kernel

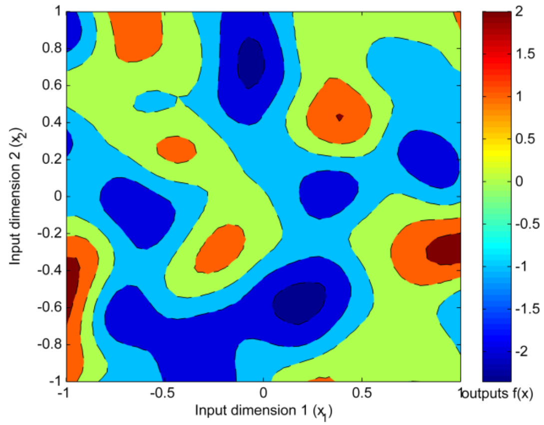


Linear (plus) SE kernel

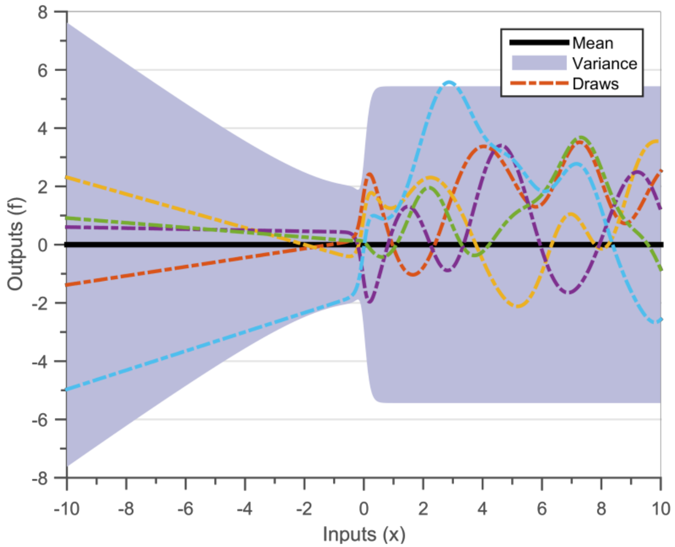


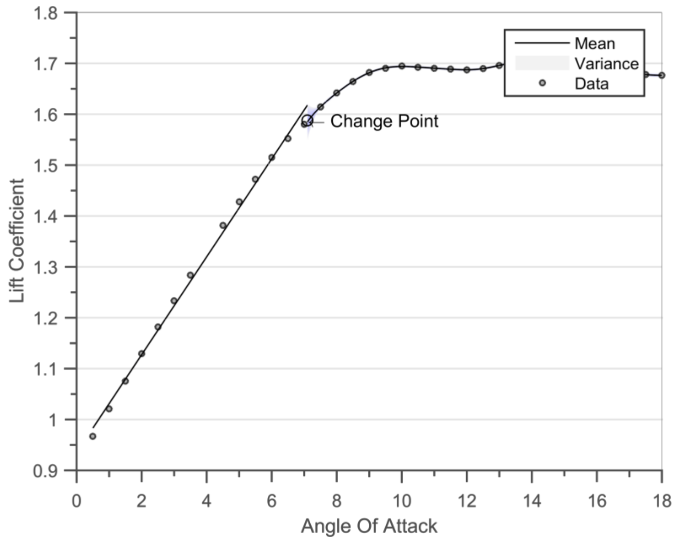
SE kernel (times) SE kernel

Across dimensions



Identifying onset of non-linearity





$$k_{CP}(k_1, k_2, x_1, x_2) = sigm(x_1)k_1 sigm(x_2) + (1 - sigm(x_1))k_2(1 - sigm(x_2))$$

- Linear domain → Non-linear domain
- Maximize ML(θ) to estimate change-point

Experimental Data

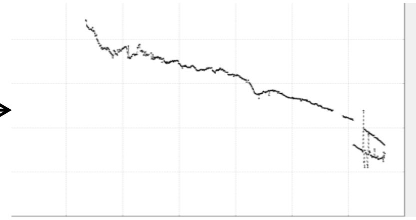
[Chiplunkar et al. SIAM, 2016]

A350-1000 : Interpolation of shock

- Flight Test Instrumentation

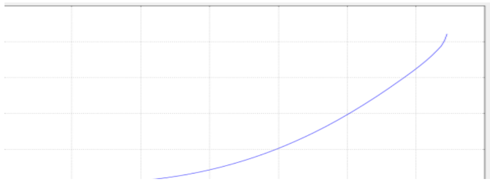
MEMS pads

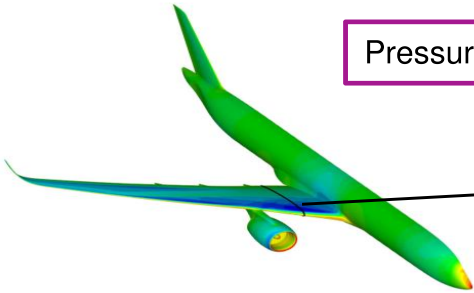
Pressure



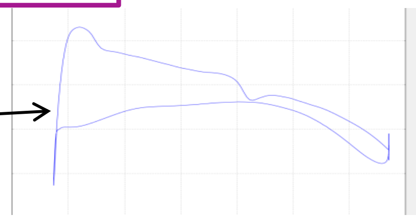
- Engineering simulations

Deformation from CSM

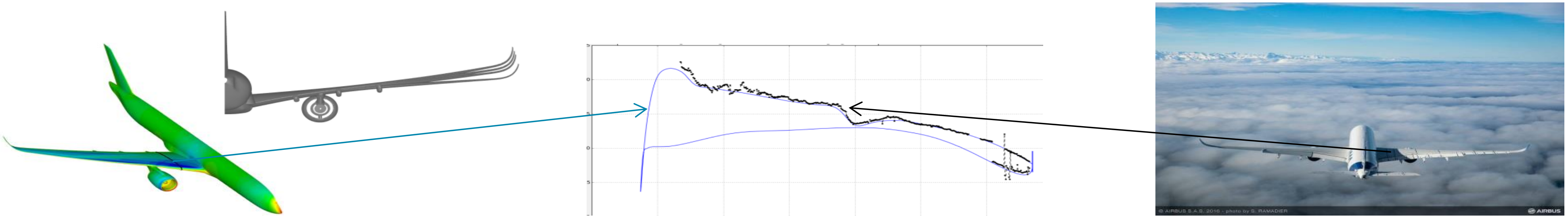




Pressure from CFD



A350-1000 Flight test analysis

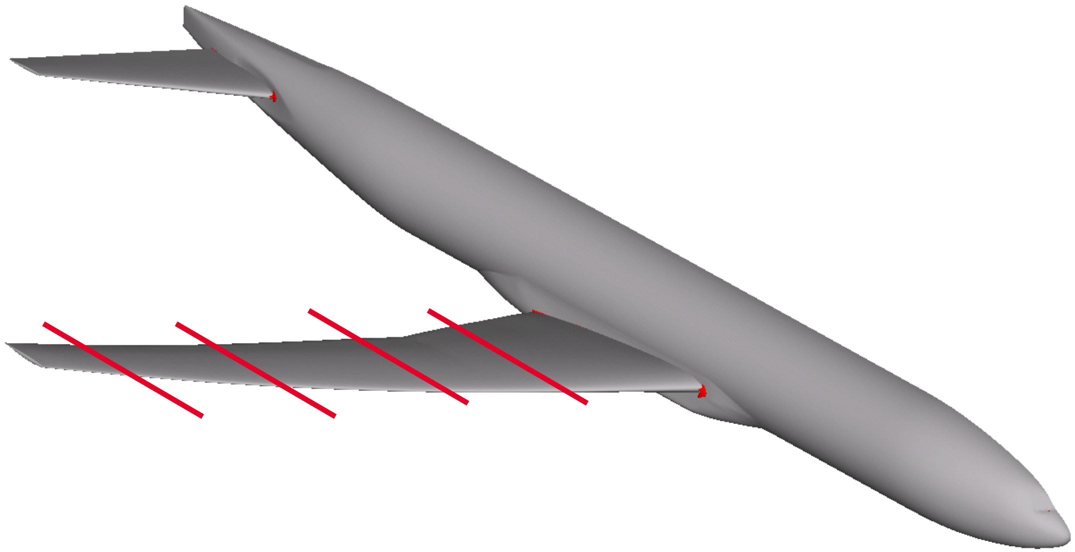


Engineering simulations

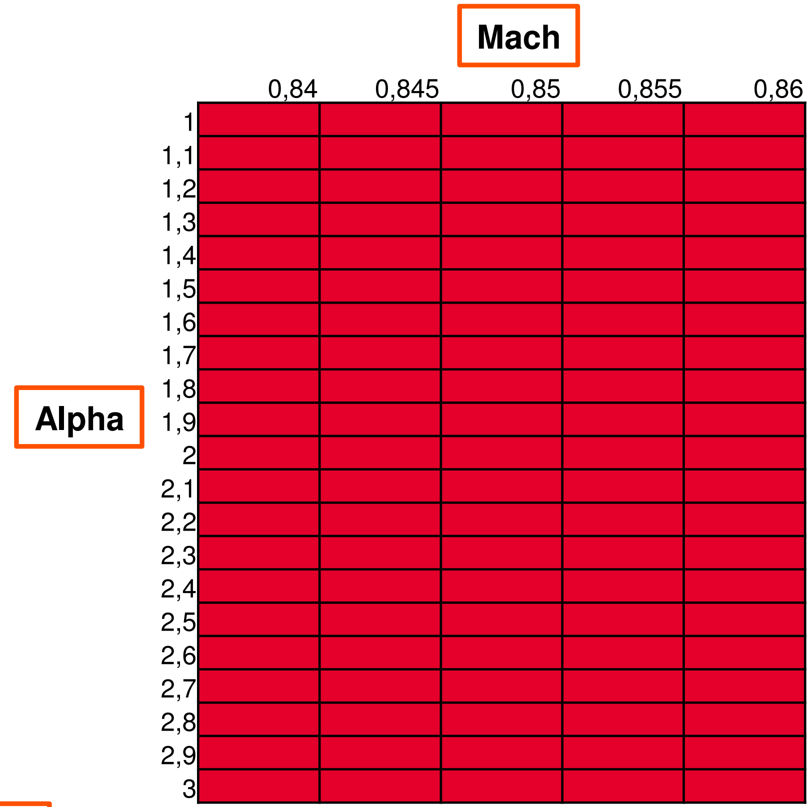
Flight Test Instrumentation

- Verify assumptions
- Extensions
- Computational bottleneck

Experimental dataset – CRM wing



https://commonresearchmodel.larc.nasa.gov/



>50k data-points

References

- **[L Cambier et al. AIAA 2008] → Elsa**
- **[John C Vassberg, et al. Journal of Aircraft, 2014] → CRM**

Different models

POD

Industrialized Process

Pressure dataset $P_i(d_j)$ → Mode shapes ϕ and amplitude α → Mode shapes constant and Interpolating α → Linear summation to predict pressures at new points

Distributed GP

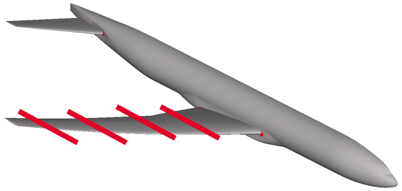
Research proposal

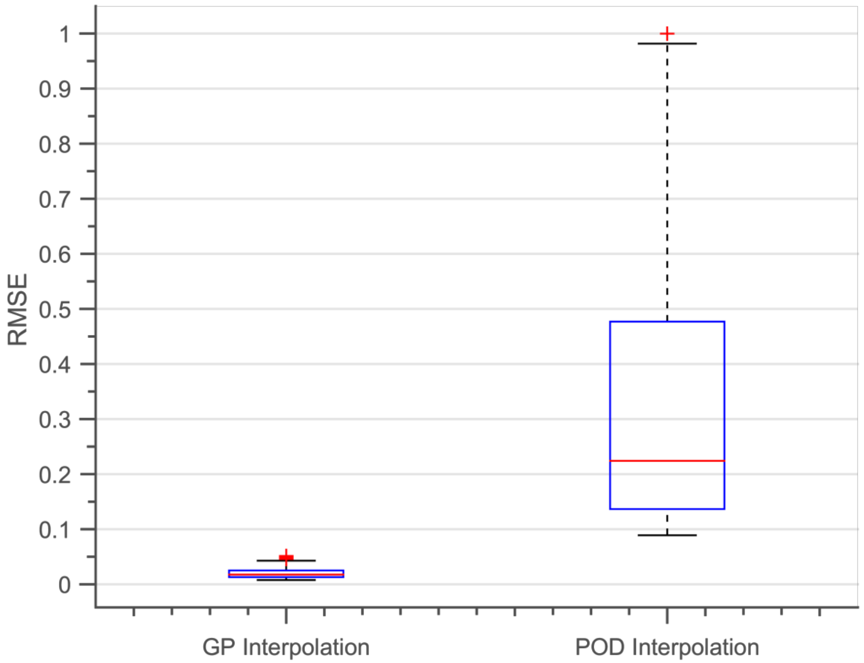
Pressure = GP(0, K) → Use Neural network kernel → Use distributed → Calculate the posterior

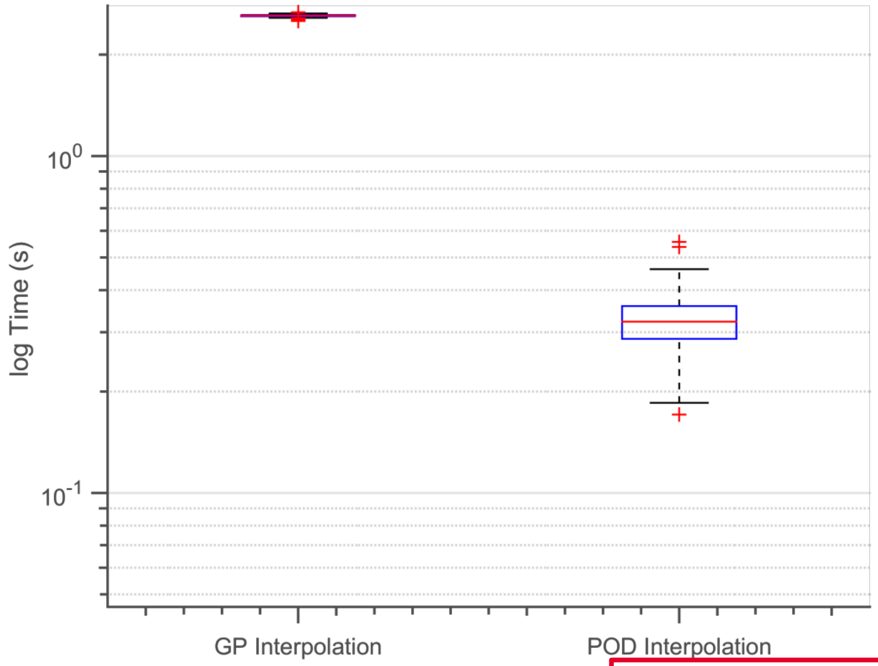
References:

- **[Benjamin Rosenbaum et al SIAM 2013] → POD**
- **[Radford M Neal. Thesis, 2012] → Neural network kernel**

Comparison of results: Cut1 (y/B = 0,105)



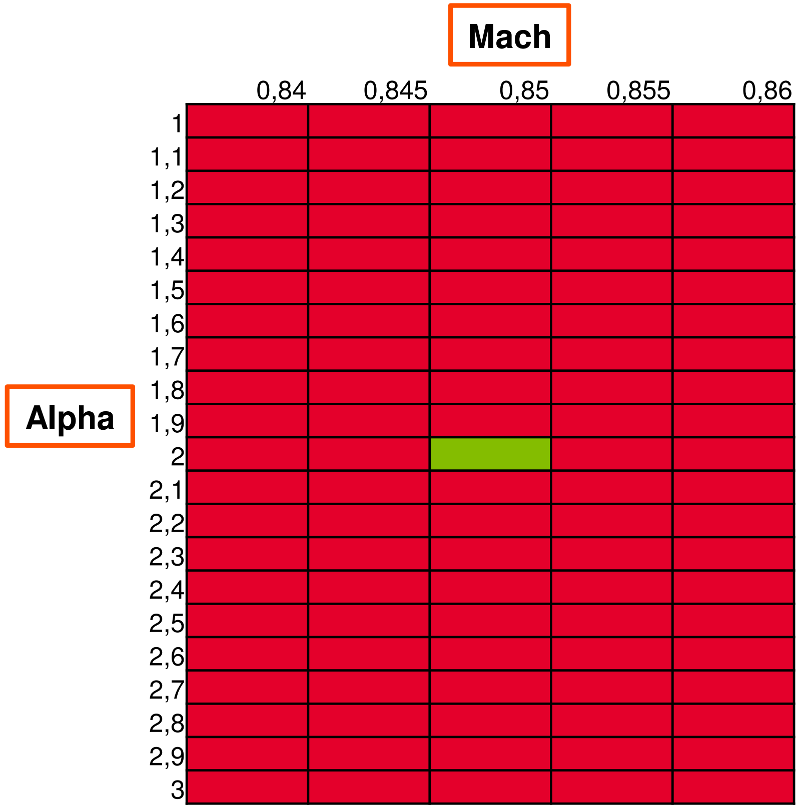


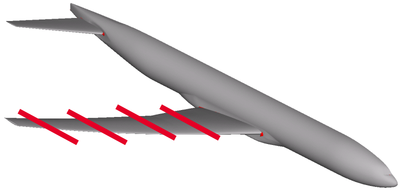


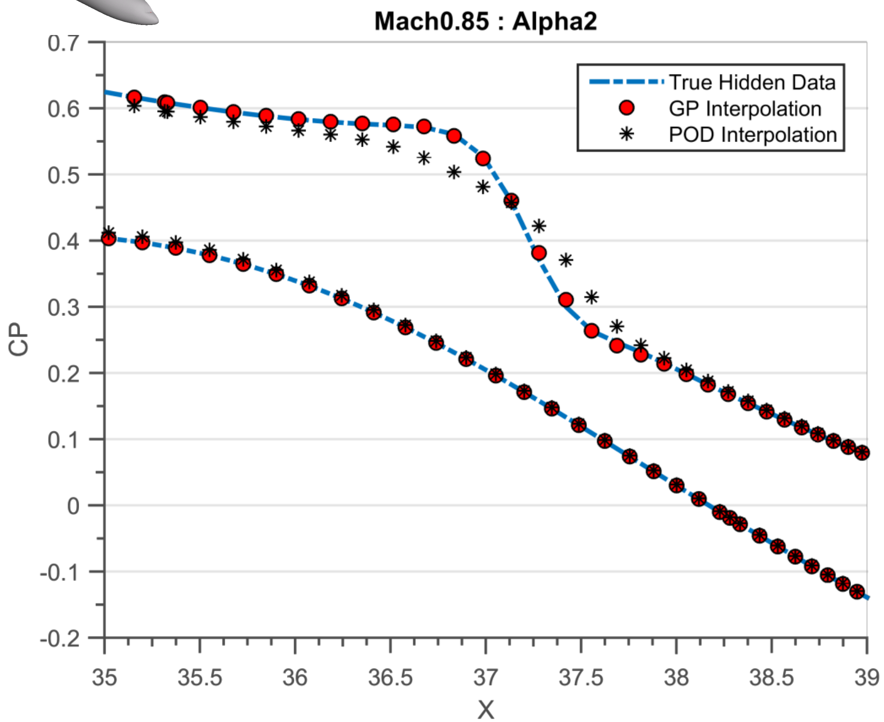
Simulated Data

[WSCMO, Chiplunkar 2017]
[AST, In review]

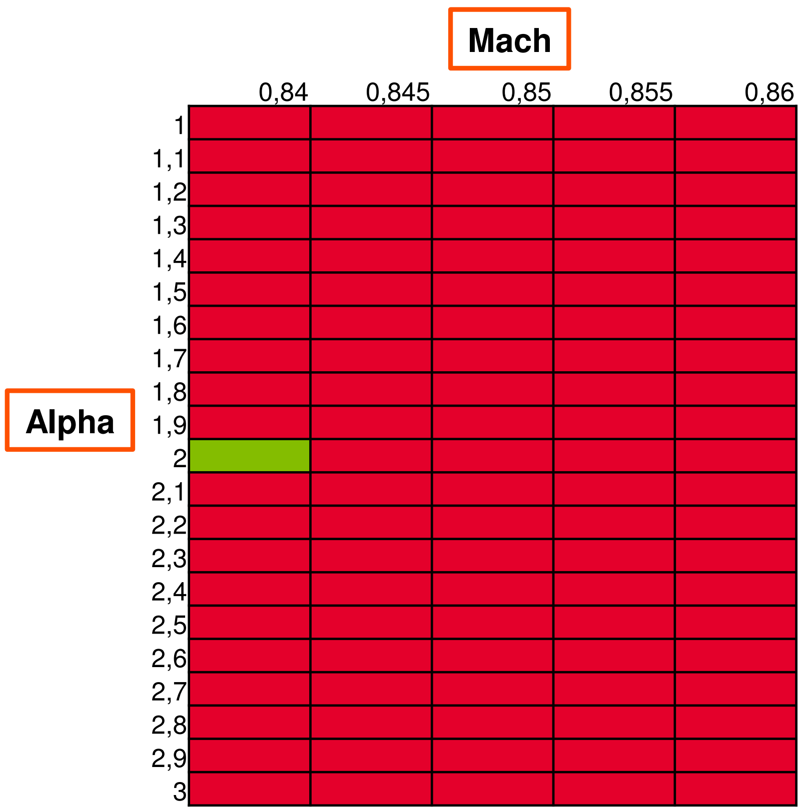
Comparison of results: Cut1 (y/B = 0,105)

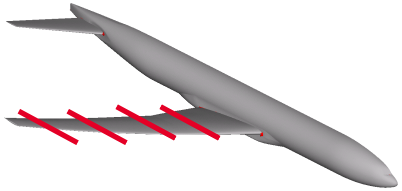


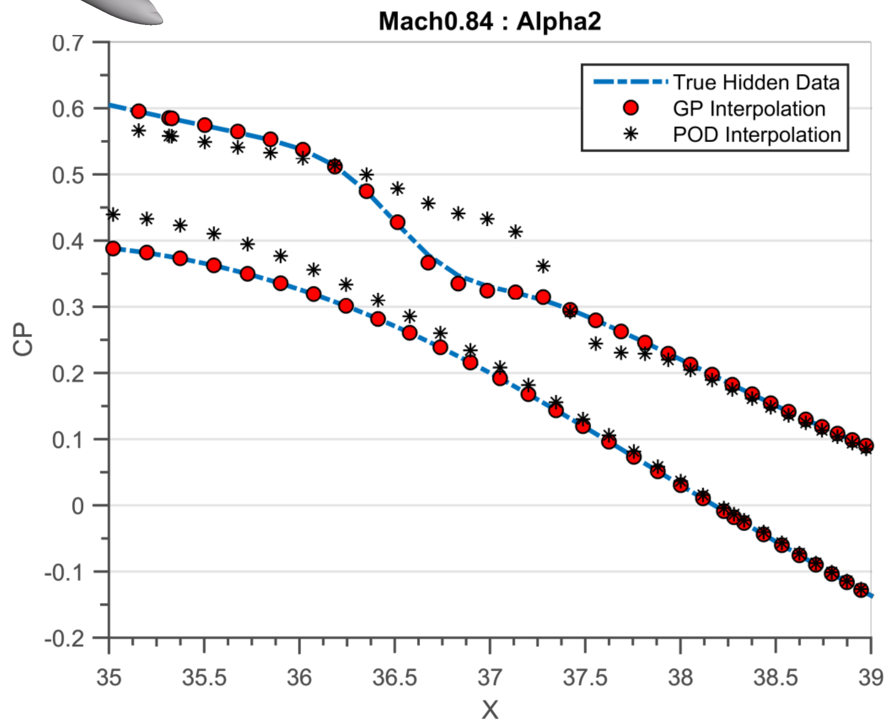




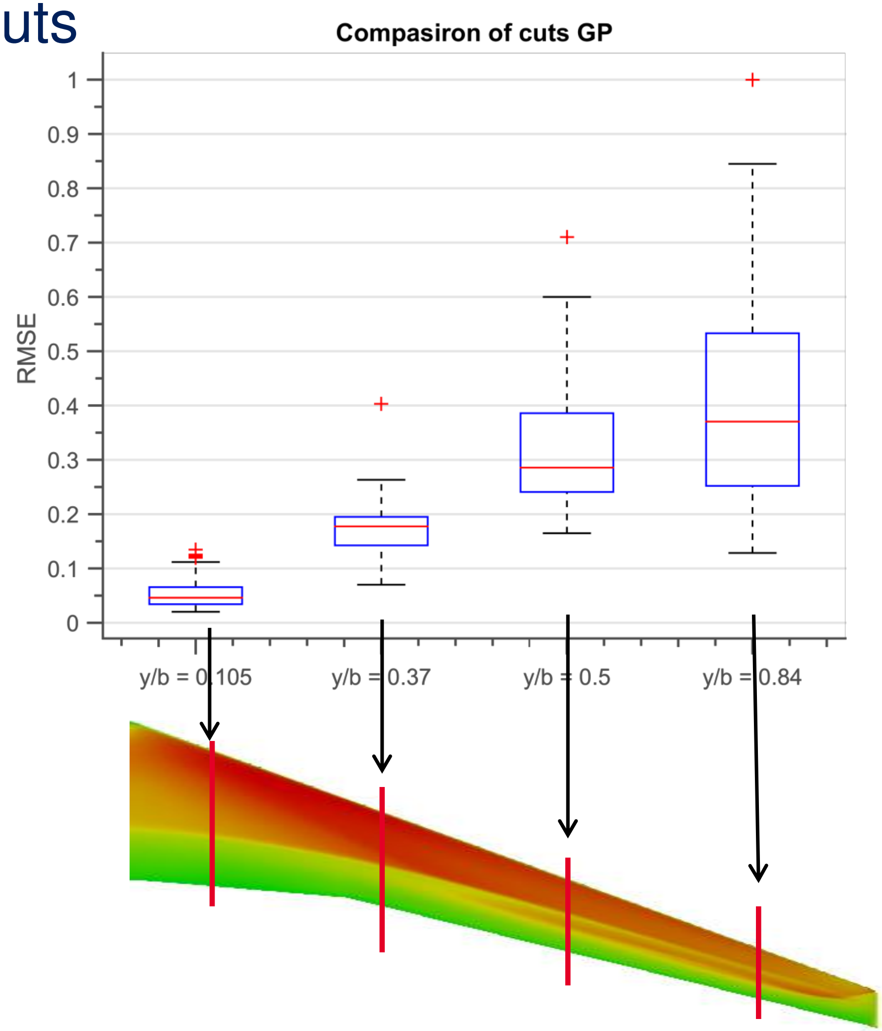
Comparison of results: Cut1 (y/B = 0,105)



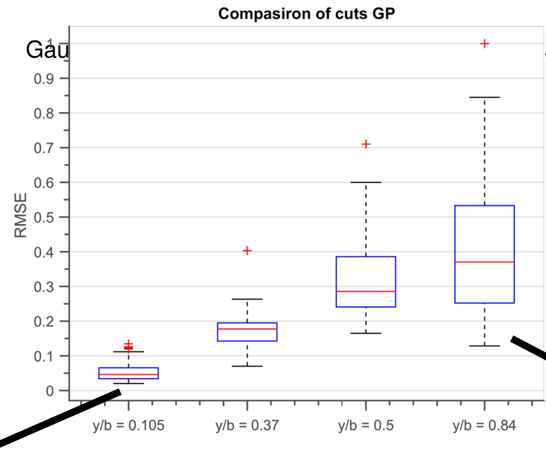


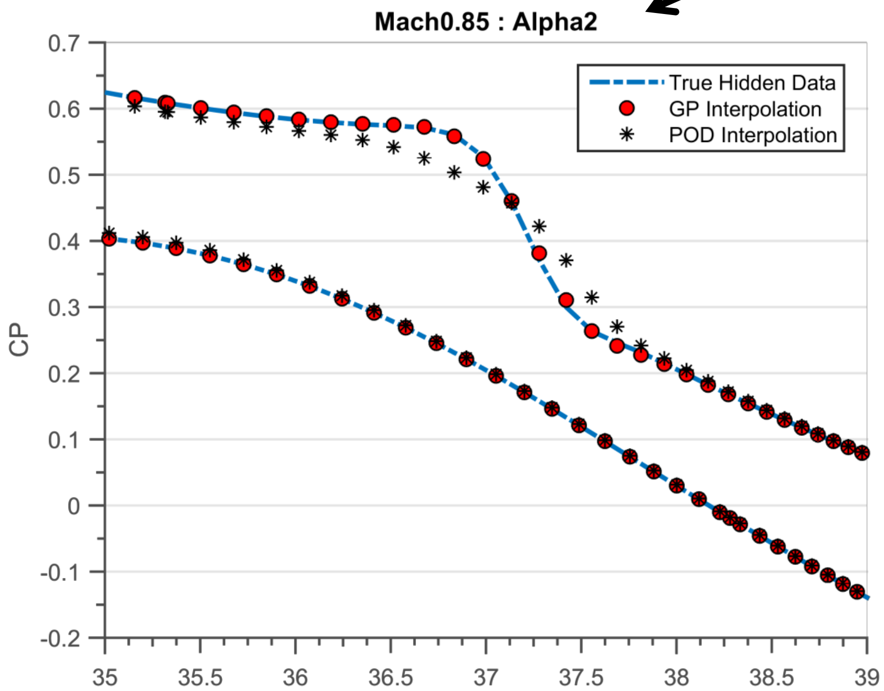


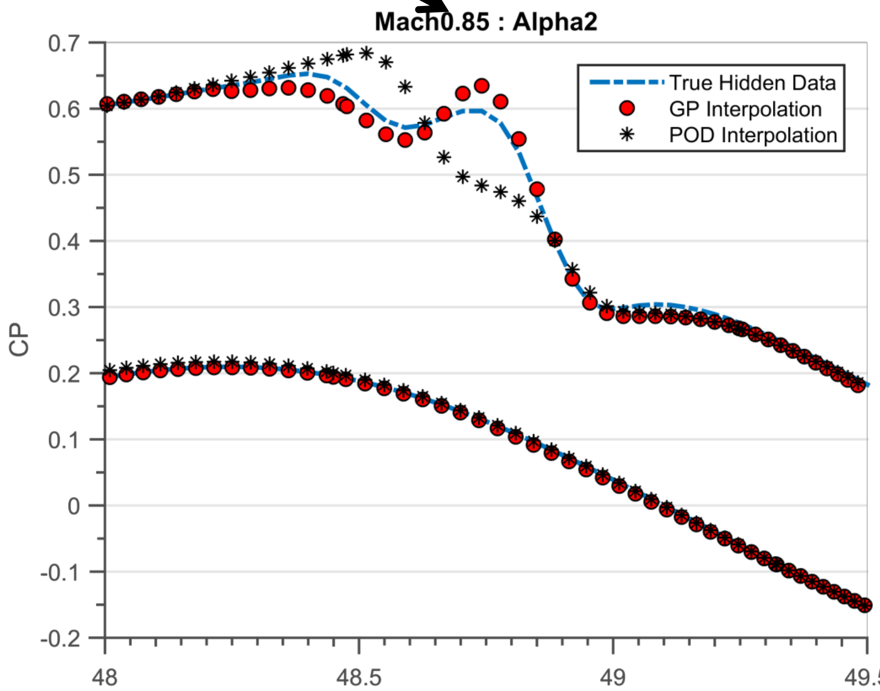
Comparison across Cuts



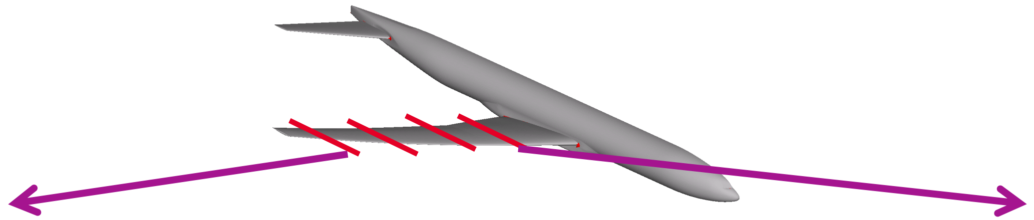
Comparison across Cuts

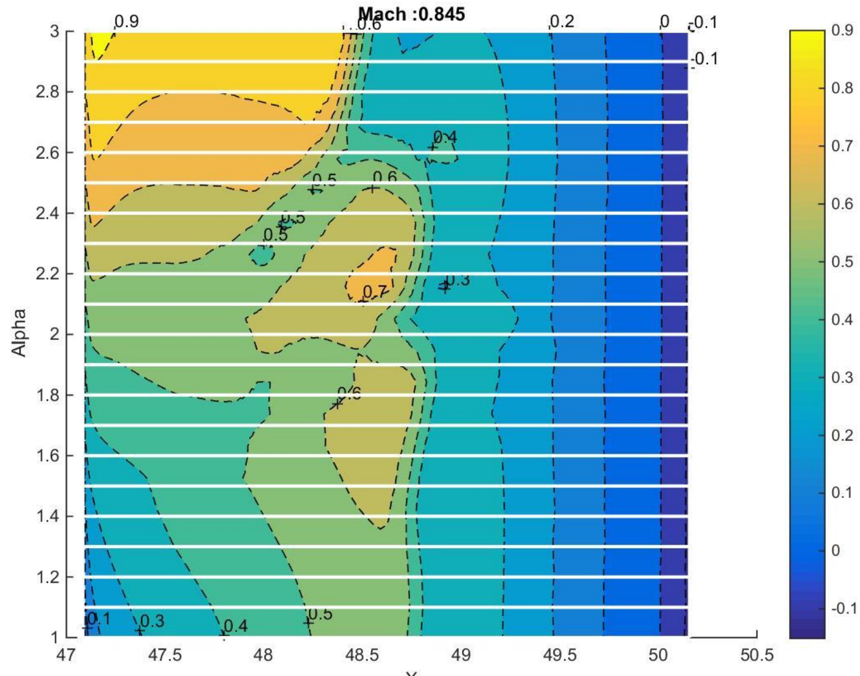




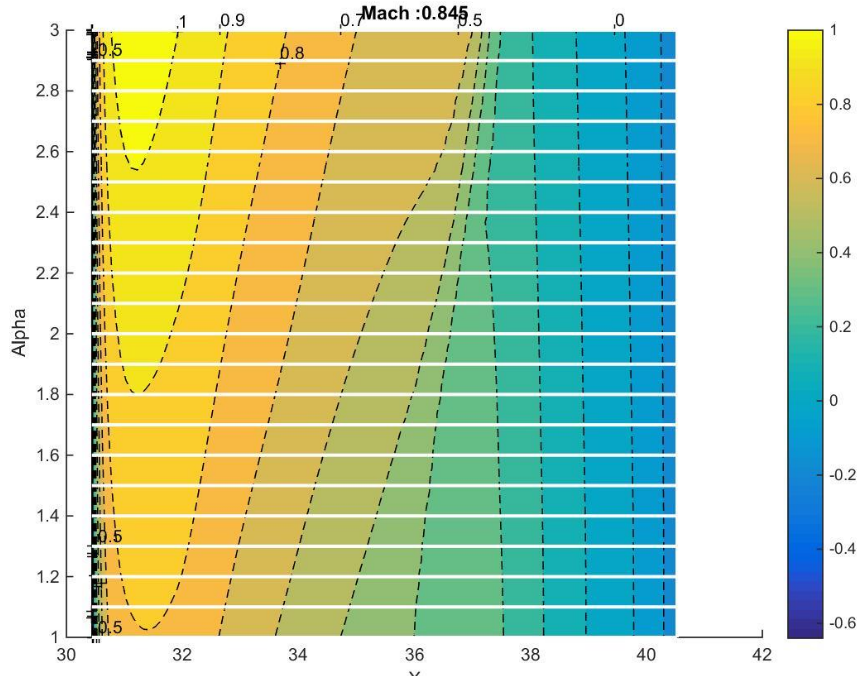


Comparison across Cuts





Cut 4



Cut 1

Conclusions and Future work

Contributions

	Additions to GP regression
Scaling multi-task GP	**ICPRAM 2016** Sparse Physics-Based Gaussian Process for Multiple outputs
	LNCS 2017 Approximate inference in related multi-output Gaussian Process Regression

	GP + aircraft design
Identifying onset of non-linearity	**SIAM 2016 -** Identification of Physical Parameters Using Change-Point Kernels
Interpolating Shock position	**WCSMO 2017 -** Gaussian Process for Aerodynamic Pressures Prediction in Fast Fluid Structure Interaction Simulations **AST Journal** - In review
Identifying Structural dynamic parameters	**IMAC 2017** Operational Modal Analysis in Frequency Domain using Gaussian Mixture Models
Adding relationships to GP	**AIAA 2016** Adding Flight Mechanics to Flight Loads Surrogate Model

Conclusions

- Manipulate covariance function → Incorporate prior information

- Several applications were proposed
  1. Change point kernel
  2. Operational Modal analysis
  3. Shock detection using adapted kernels
  4. Merging experimental and simulation data
  5. Incorporate functional relationships

- Scalability solutions

Future works

- Real time comparison between simulation and experiment
- Uncertainty quantification
- Adding shape information
- Adding relationships
    - Filtering
    - Identify faulty sensors
    - MDO optimization
- Adding simulation data
    - Model updating
    - Extrapolation

References

1. Pedro Domingos. A few useful things to know about machine learning, 2012
2. Daniel G Krige. A statistical approach to some mine valuation and allied problems on the Witwatersrand. PhD thesis, 2015
3. G. Matheron. Traite de geostatistique appliquee. 1962.
4. Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian Processes for Machine Learning (Adaptive Computation and Machine learning). 2005
5. Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In In Artificial Intelligence and Statistics 2009
6. Christopher KI Williams et Matthias Seeger. Using the Nyström method to speed up kernel machines. In Advances in neural information processing systems, 2001
7. Marc Peter Deisenroth and Jun Wei Ng. Distributed gaussian processes. 2015
8. S. Bochner, M. Tenenbaum et H. Pollard. Lectures on fourier integrals. Annals of mathematics studies. Princeton University Press, 1959
9. Wilson, A. G. and Adams, R. P., Gaussian Process Kernels for Pattern Discovery and Extrapolation Supplementary Material
10. Andrew Gordon Wilson. Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes. PhD thesis, University of Cambridge,2014
11. OH James III et TG Came. The Natural Excitation Technique (NExT) for Modal parameter extraction from operating structures. The International journal of analytical and experimental modal analysis, 1995.
12. Randall J Allemang et DL Brown. A unified matrix polynomial approach to modal identification. Journal of Sound and Vibration, 1998
13. Proper orthogonal decomposition extensions and their applications in steady aerodynamics. Master of Engineering in High Performance Computation for Engineered Systems (HPCES), 2003.
14. Radford M Neal. Bayesian learning for neural networks, volume 118. Springer Science & Business Media, 2012
15. Michael Osborne. Bayesian gaussian processes for sequential prediction, optimisation and quadrature. Oxford University New College, 2010
16. Mauricio A Alvarez, Lorenzo Rosasco et Neil D Lawrence. Kernels for vector-valued functions: A review. arXiv preprint arXiv:1106.6251, 2011
17. Emil M. Constantinescu and Mihai Anitescu. Physics-based covariance models for gaussian processes with multiple outputs. 2013
18. Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction, 2007
19. E Bosco, A Chiplunkar, J Morlier, Estimation of Modal Parameters Confidence Intervals: A Simple Numerical Example, 2013
20. James Robert Lloyd, Representation, learning, description and criticism of probabilistic models with applications to networks, functions and relational data, 2014
21. David Duvenaud. Automatic Model Construction with Gaussian Processes. PhD thesis, 2014

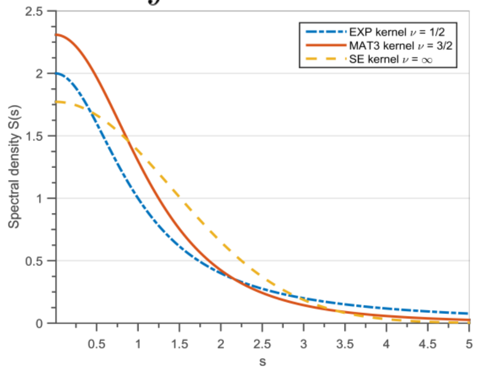
Thank you

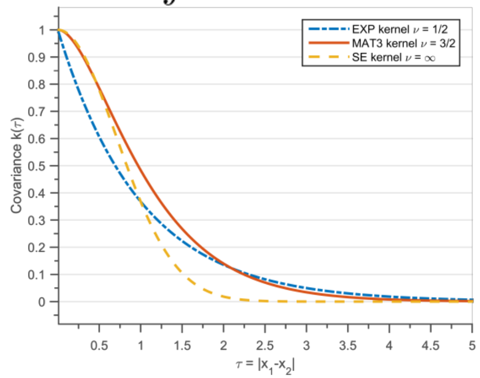
Questions?

Bochners' theorem

Fourier transform of a stationary covariance function exists and is a positive finite measure

$$S(s) = \int k(\tau) e^{-2\pi i s^T \tau} d\tau \qquad k(\tau) = \int S(s) e^{2\pi i s^T \tau} ds$$





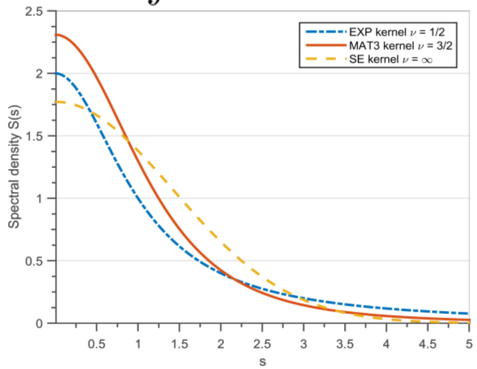
References:

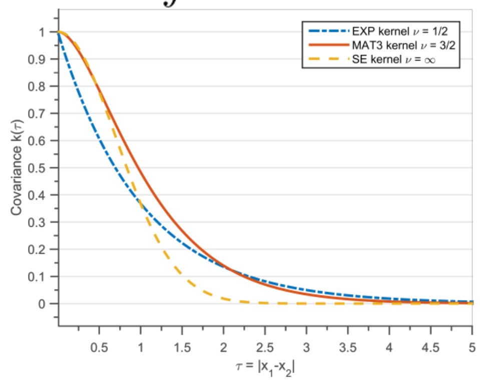
S. Bochner, M. Tenenbaum et H. Pollard. Lectures on fourier integrals. Annals of mathematics studies. Princeton University Press, 1959

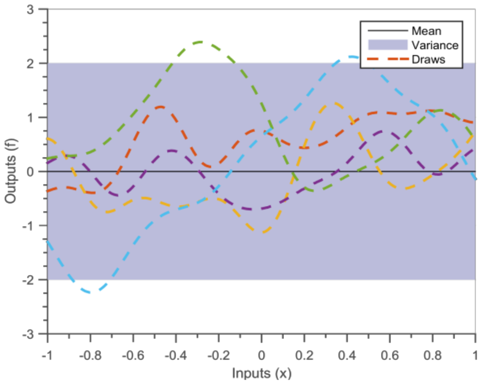
Bochners' theorem

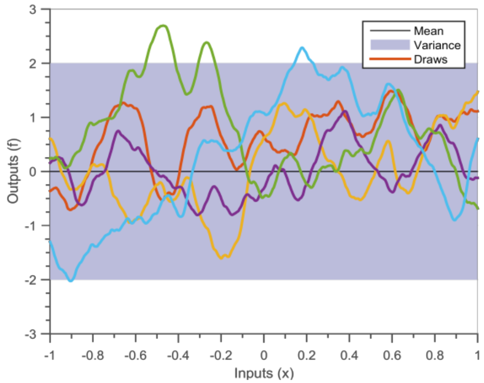
Fourier transform of a stationary covariance function exists and is a positive finite measure

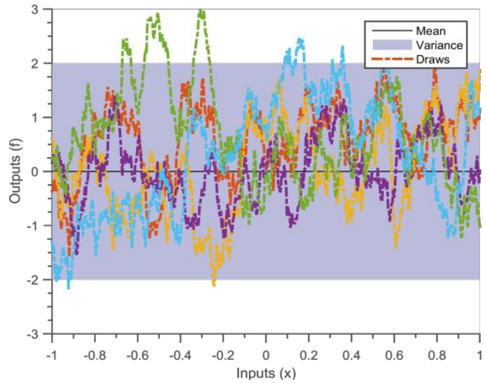
$$S(s) = \int k(\tau) e^{-2\pi i s^T \tau} d\tau \qquad k(\tau) = \int S(s) e^{2\pi i s^T \tau} ds$$



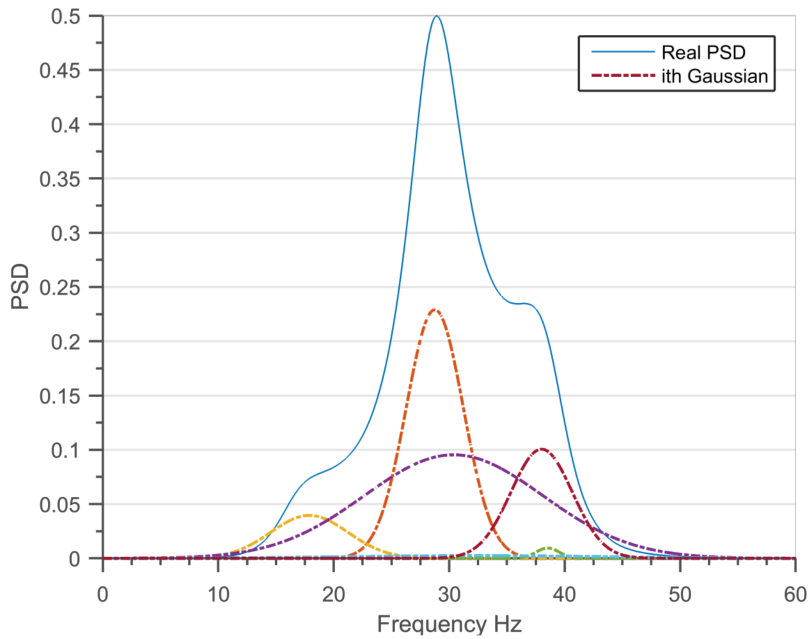








Spectral mixture kernel



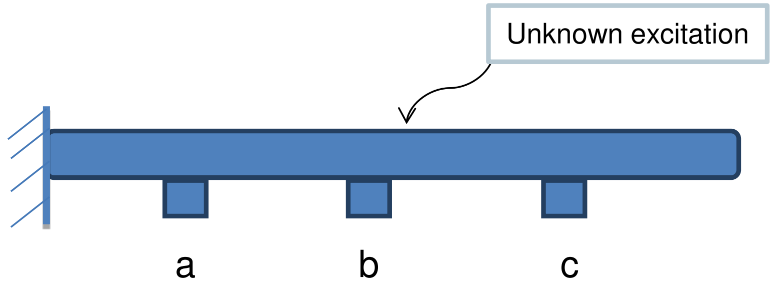
$$S_{SM}(s,\mu,\sigma,w)=\sum_{q=1}^{Q}\frac{w_q}{\sqrt{2\pi\sigma_q^2}}\left(\exp\left[-\frac{(s-\mu_q)^2}{2\sigma_q^2}\right]+\exp\left[-\frac{(-s-\mu_q)^2}{2\sigma_q^2}\right]\right)$$

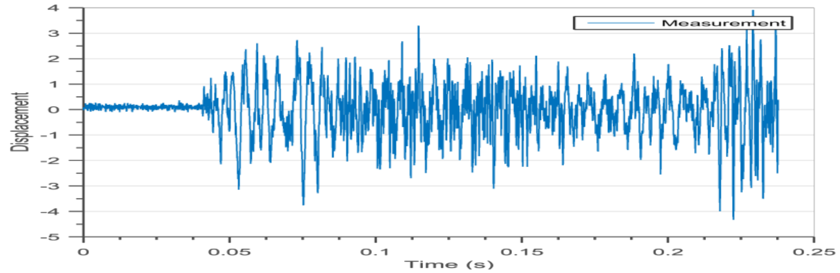
$$k_{SM}(d,\mu,\sigma,w)=\sum_{q=1}^{Q}w_q cos(2\pi\mu_q)exp[-2\pi^2d^2\sigma_q^2]$$

References:

Wilson, A. G. and Adams, R. P., Gaussian Process Kernels for Pattern Discovery and Extrapolation Supplementary Material

Application - Operational Modal analysis





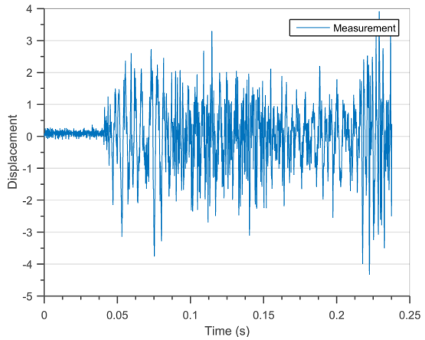
1. Sample is subjected to unknown random excitations
2. Can be used to perform damage-detection or structural health monitoring
3. Peaks in the power spectral density give us the modes of the structure

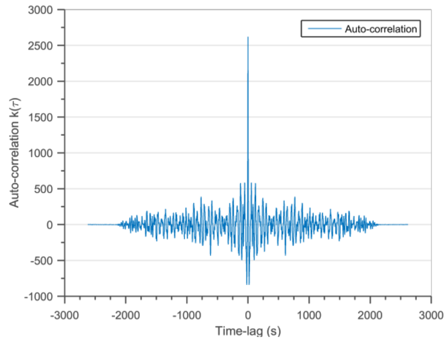
[IMAC, Chiplunkar 2017b]

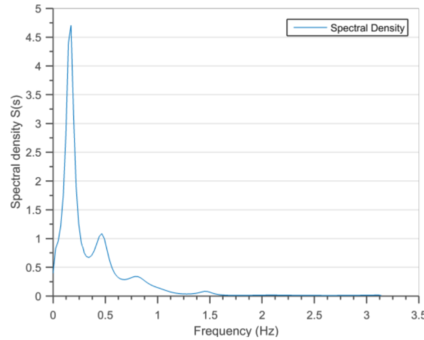
References:

DiazDelaO, F. A., and S. Adhikari. "Structural dynamic analysis using Gaussian process emulators." *Engineering Computations* 27.5 (2010)

Operational Modal analysis

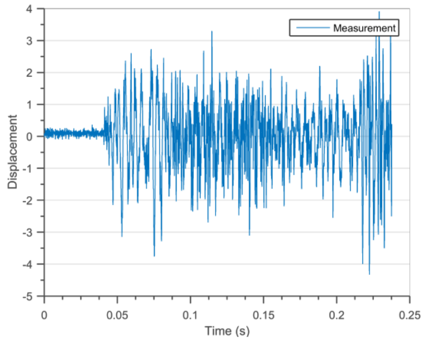


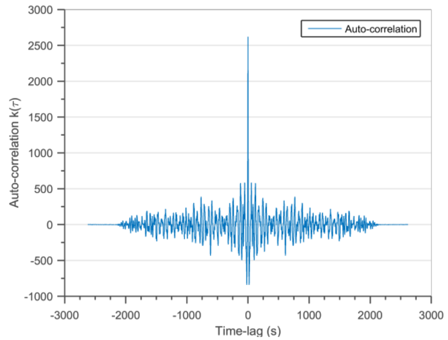


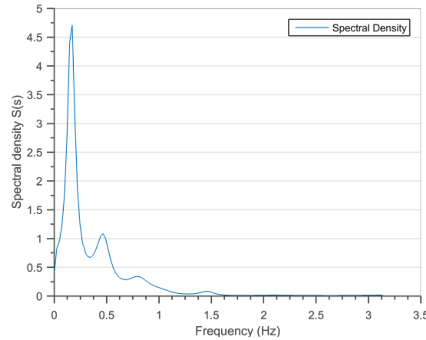


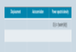
Displacement	Autocorrelation	Power spectral density
$x(t)$	$k(\tau) = \int x(t)x(t-\tau)dt$	$S(s) = fourier(k(\tau))$

Operational Modal analysis







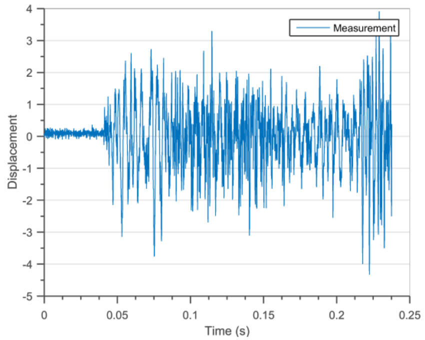
Displacement	Autocorrelation	Power spectral density
	$k(\tau) = \int x(t)x(t-\tau)dt$	$S(s) = fourier(k(\tau))$
$M\{\ddot{x}(t)\}+C\{\dot{x}(t)\}+K\{x(t)\} = \{f(t)\}$		
	$k(\tau) = \sum A_i exp(-\lambda_i \tau) sin(B_i \tau)$	$S(s) = \frac{\sum a_k(s)^k}{\sum b_l(s)^l}$

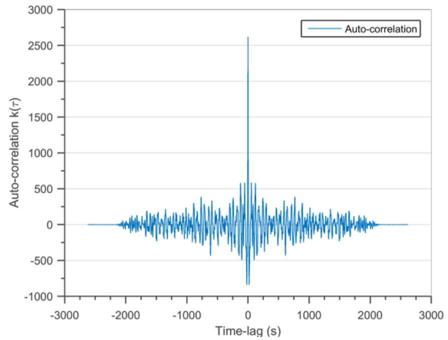
References:

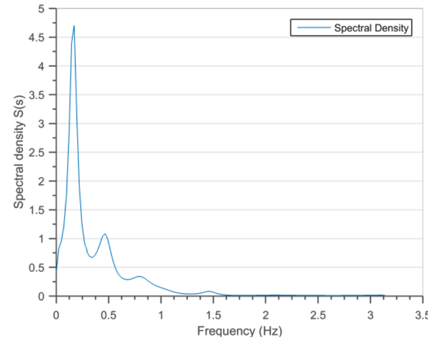
OH James III et TG Came. The Natural Excitation Technique (NExT) for Modal parameter extraction from operating structures. The International journal of analytical and experimental modal analysis, 1995.

Randall J Allemang et DL Brown. A unified matrix polynomial approach to modal identification. Journal of Sound and Vibration, 1998

Operational Modal analysis

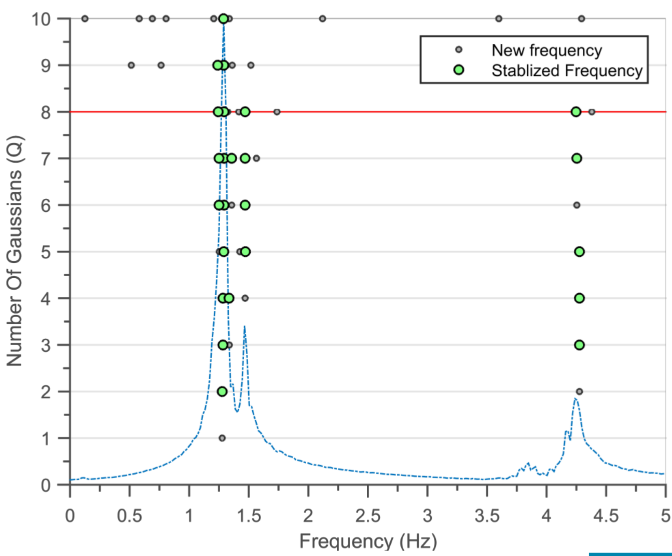


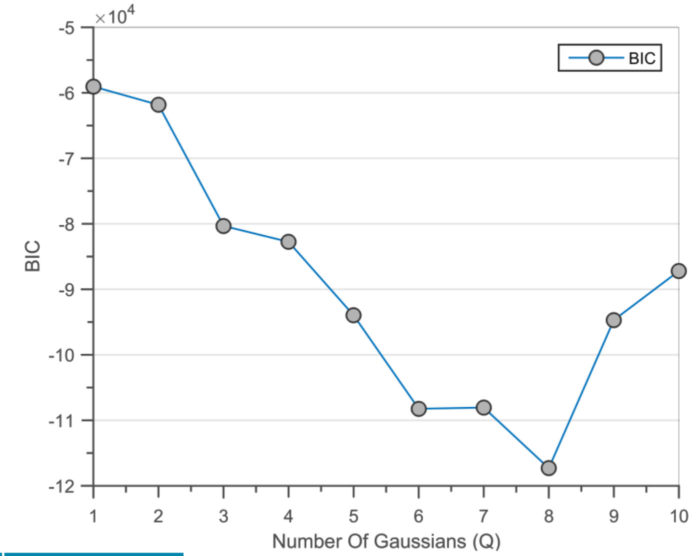




Displacement	Autocorrelation	Power spectral density
$x(t)$	$k(\tau) = \int x(t)x(t-\tau)dt$	$S(s) = fourier(k(\tau))$
$M\{\ddot{x}(t)\}+C\{\dot{x}(t)\}+K\{x(t)\} = \{f(t)\}$		
	$k(\tau) = \sum A_i exp(-\lambda_i \tau) sin(B_i \tau)$	$S(s) = \frac{\sum a_k (s)^k}{\sum b_l (s)^l}$
Spectral Mixture Covariance		
$x(t) = GP(0, cov_{SM}(t,t'))$	$k_{SM}(d,\mu,\sigma,w) = \sum_{q=1}^{Q} w_q cos(2\pi\mu_q) exp[-2\pi^2 d^2 \sigma_q^2]$	$S_{SM}(s,\mu,\sigma,w) = \sum_{q=1}^{Q} \frac{w_q}{\sqrt{2\pi\sigma_q^2}} \left(\exp\left[-\frac{(s-\mu_q)^2}{2\sigma_q^2}\right] + \exp\left[-\frac{(-s-\mu_q)^2}{2\sigma_q^2}\right] \right)$

Results on HCT dataset



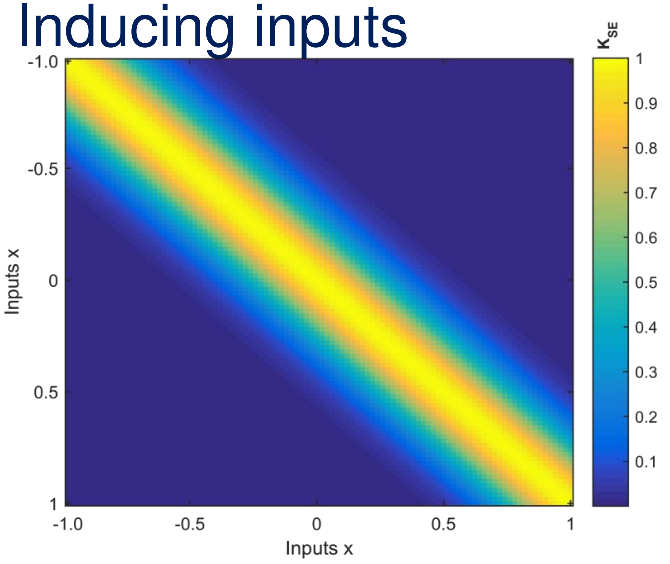


	From paper	FRF
Freq1	1,23	1,242
Freq2	1,27	1,292
Freq3	1,43	1,470
Freq4	3,86	3,866
Freq5	4,25	4,247

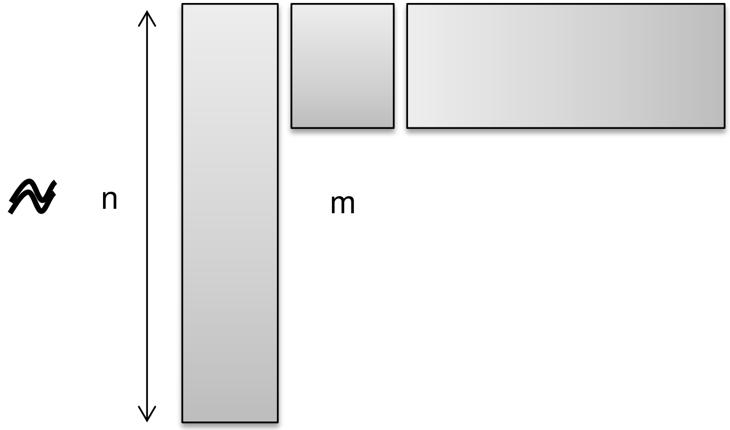
Experimental Data (HCT dataset)

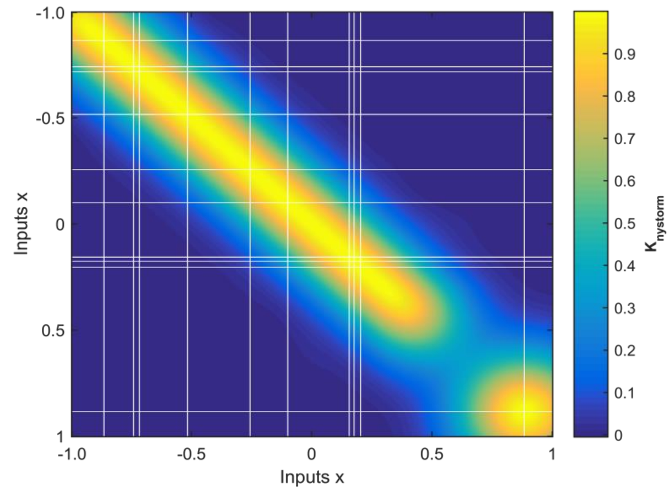
Scaling solutions to a GP

Inducing inputs



K_{xx}





$Kxx_{approximate}$

$$\boxed{m(x_*) = K_{*approximate}\left[Kxx_{approximate}\right]^{-1} y}$$

$$\boxed{var(x_*, x'_*) = K_{**approximate} - K^T_{*approximate}\left[Kxx_{approximate}\right]^{-1} K_{*approximate}}$$

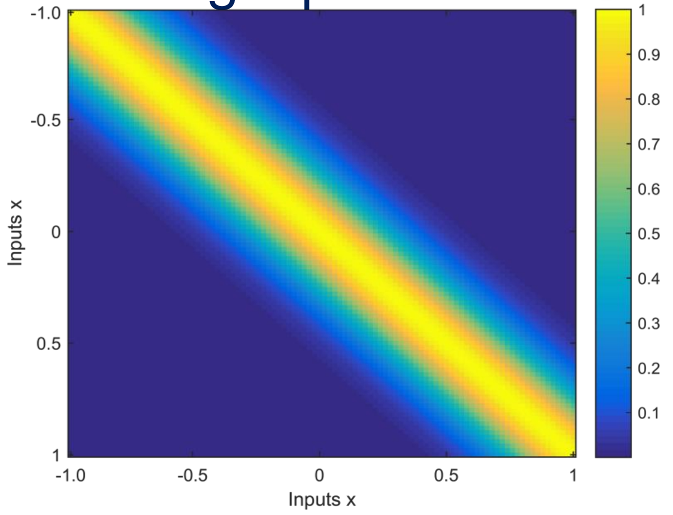
$$\boxed{log\big(p(y|X,\theta)\big) = -\frac{1}{2} y^T K_{approximate}{}^{-1} - \frac{1}{2} log\left|K_{approximate}\right| - \frac{n}{2}\log(2\pi)}$$

References:

Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In In Artificial Intelligence and Statistics 2009

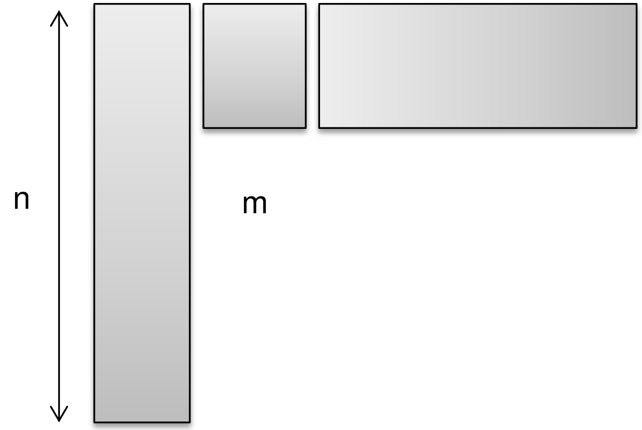
Christopher KI Williams et Matthias Seeger. Using the Nyström method to speed up kernel machines. In Advances in neural information processing systems, 2001

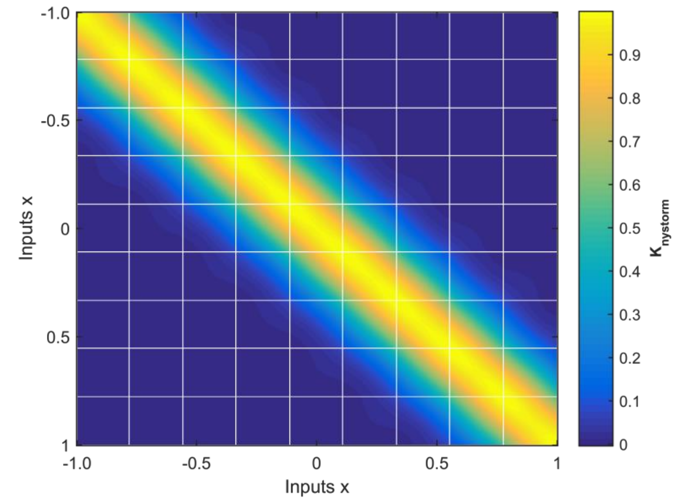
Inducing inputs



K_{xx}

$\approx$





$Kxx_{approximate}$

$$\boldsymbol{m(x_*) = K_{*approximate}\left[Kxx_{approximate}\right]^{-1} y}$$

$$\boldsymbol{var(x_*, x_*') = K_{**approximate} - K_{*approximate}^{T}\left[Kxx_{approximate}\right]^{-1} K_{*approximate}}$$

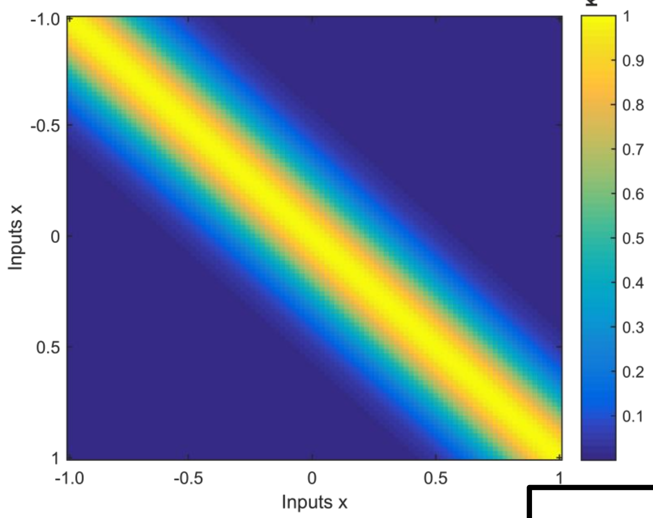
$$log\big(p(y|X,\theta)\big) = -\frac{1}{2}\boldsymbol{y^T} K_{approximate}{}^{-1} - \frac{1}{2}\boldsymbol{log}\left|K_{approximate}\right| - \frac{n}{2}\log(2\pi)$$

References:

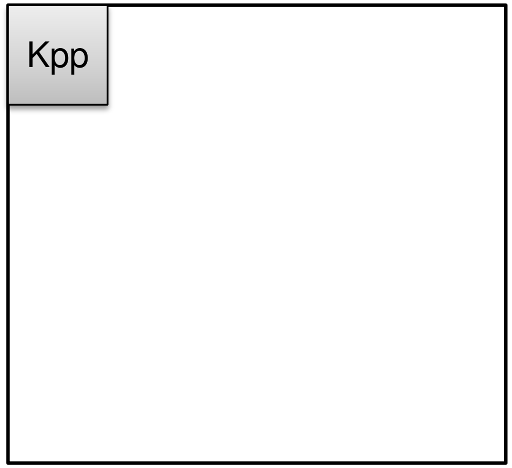
Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In In Artificial Intelligence and Statistics 2009

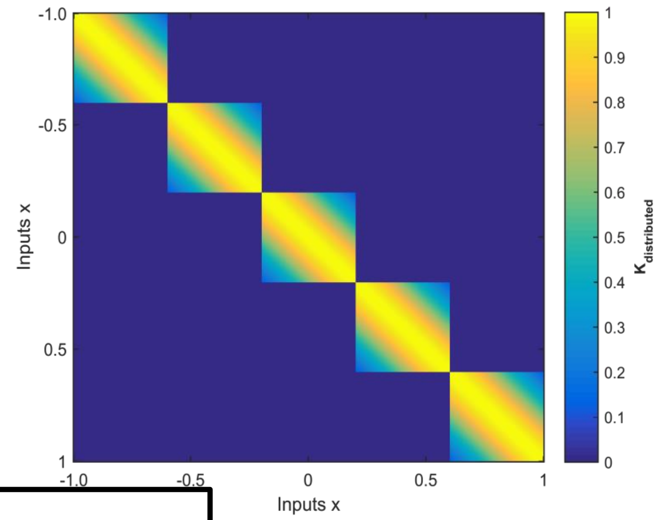
Christopher KI Williams et Matthias Seeger. Using the Nyström method to speed up kernel machines. In Advances in neural information processing systems, 2001

Distributed Gaussian Process



$\approx$



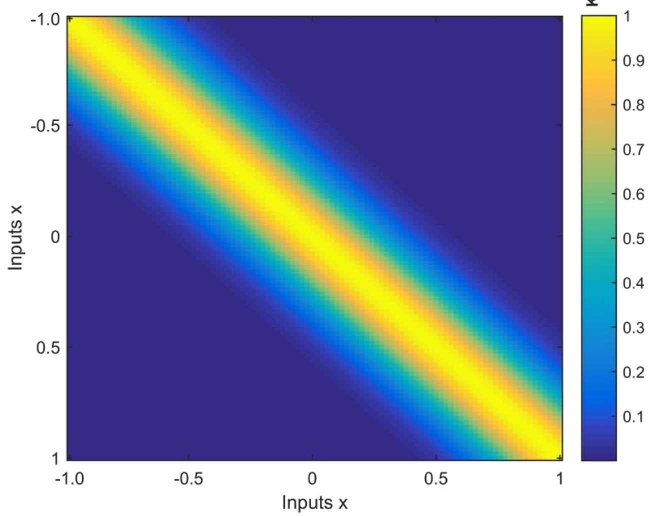


$$m(y_*) = Kx_*x_*^{-2} \sum_k \beta_k \, Cov_k(y_*)^{-2} \, m_k(y_*)$$

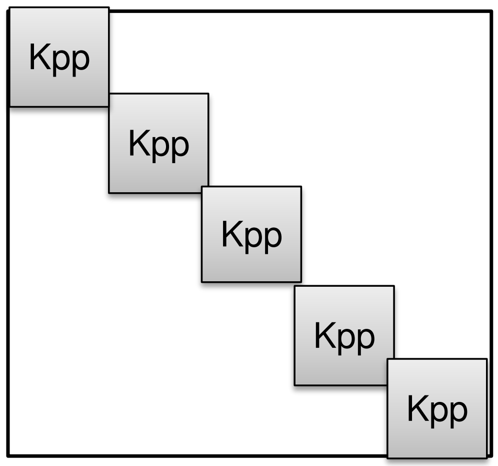
$$\beta_k = \log\left(K_{x_*x_*}{}^{-2}\right) - \log\left(Cov_k(y)^{-2}\right)$$

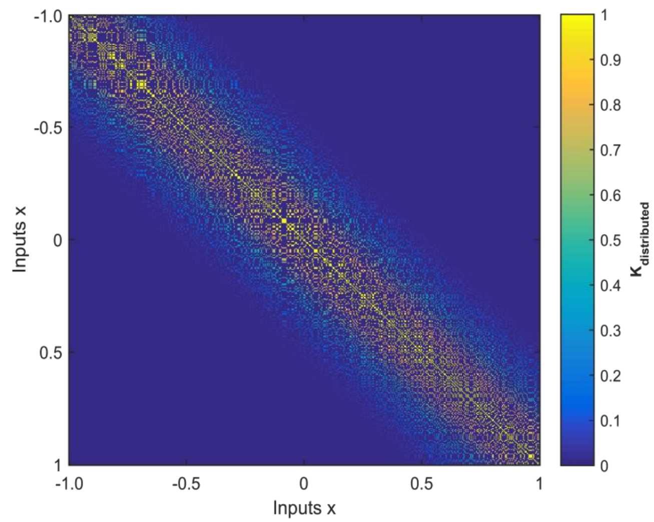
References:
Tao Chen et Jianghong Ren. Bagging for Gaussian process regression. Neurocomputing, 2009.
Marc Peter Deisenroth et Jun Wei Ng. Distributed Gaussian Processes. 2015

Distributed Gaussian Process



$\approx$





$$m(y_*) = K_{x_* x_*}^{-2} \sum_k \beta_k \, Cov_k(y_*) \, m_k(y_*)$$

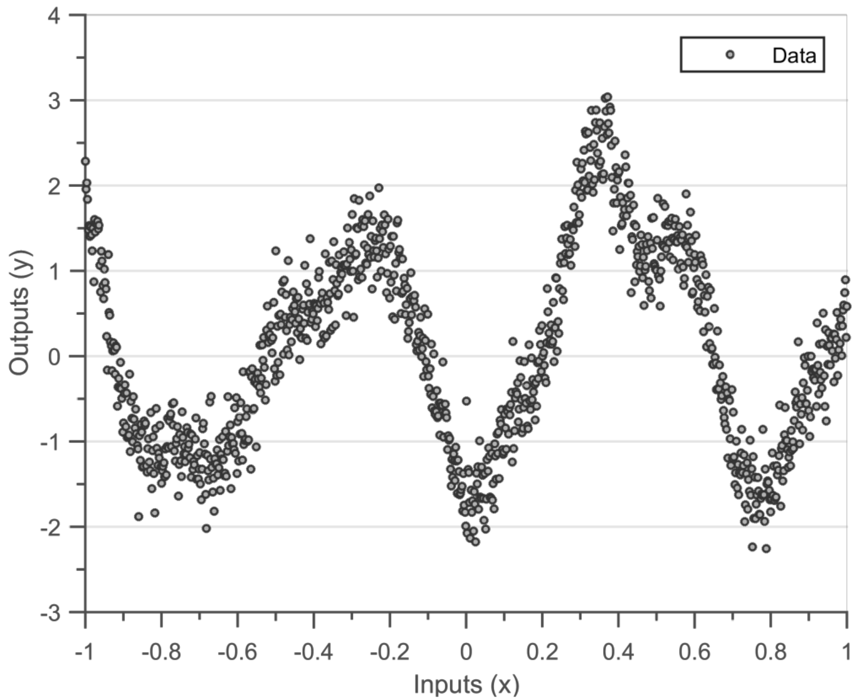
$$\beta_k = \log\left(K_{x_* x_*}{}^{-2}\right) - \log\left(Cov_k(y)^{-2}\right)$$

References:

Tao Chen et Jianghong Ren. Bagging for Gaussian process regression. Neurocomputing, 2009.
Marc Peter Deisenroth et Jun Wei Ng. Distributed Gaussian Processes. 2015

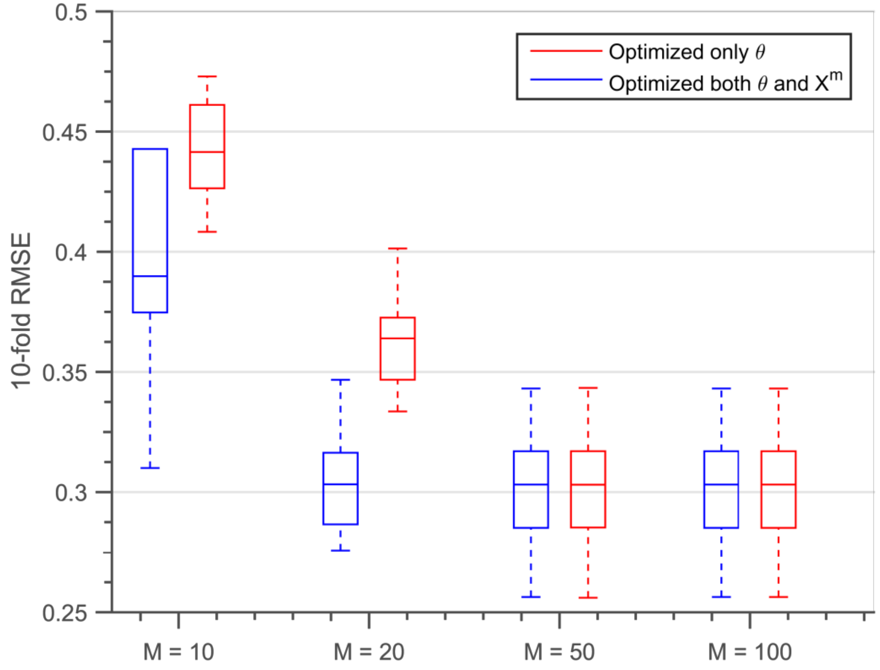
Comparison between two approximation methods

- $f(x) = GP(0, K_{se}(1, 0.1))$
- $y(x) = f(x) + (0.3)$*rand
- N = 1000
- 10 fold Cross-validation

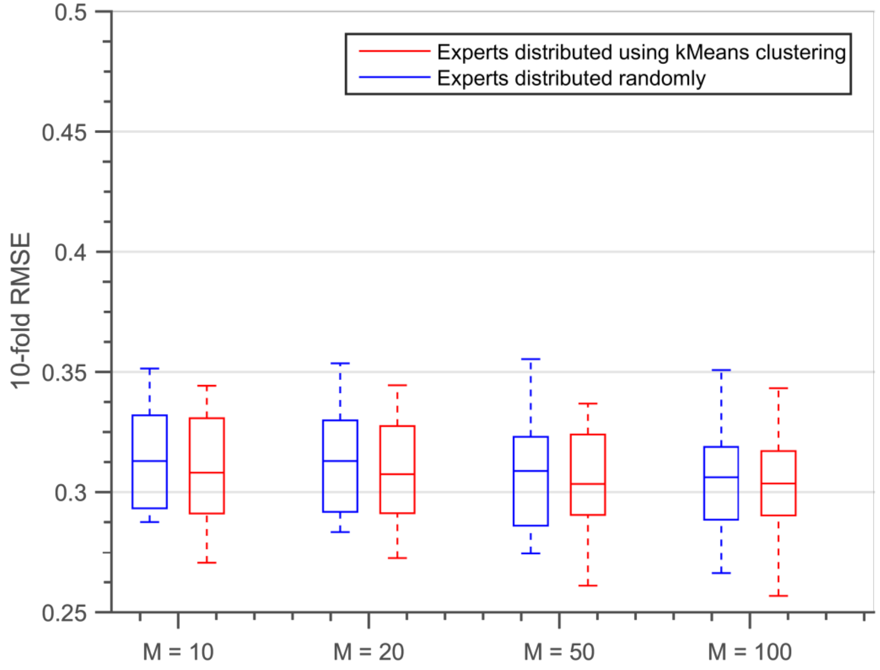


Comparison between two approximation methods

- $f(x) = GP(0, K_{se}(1, 0.1))$
- $y(x) = f(x) + (0.3)*rand$
- $N = 1000$
- 10 fold Cross-validation



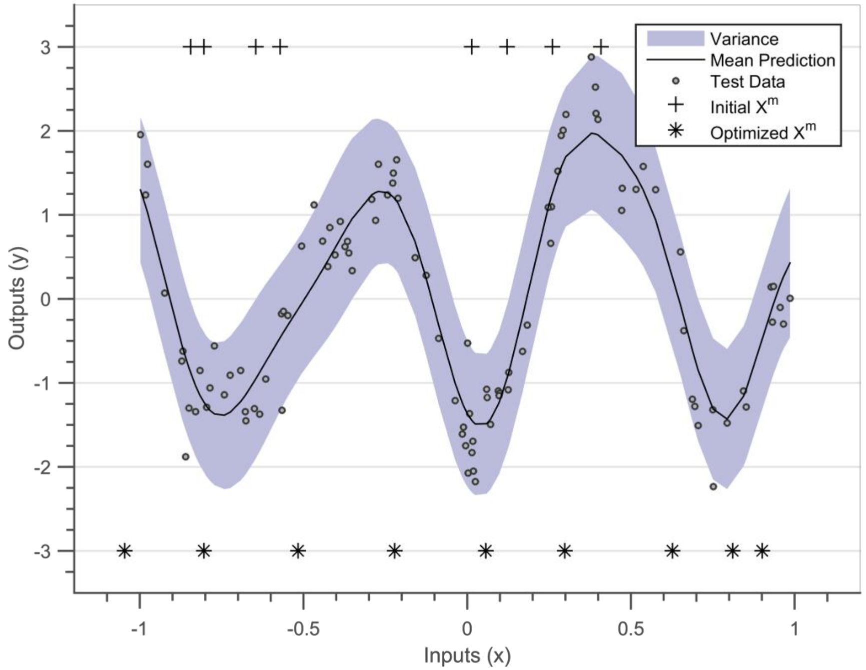
Inducing inputs



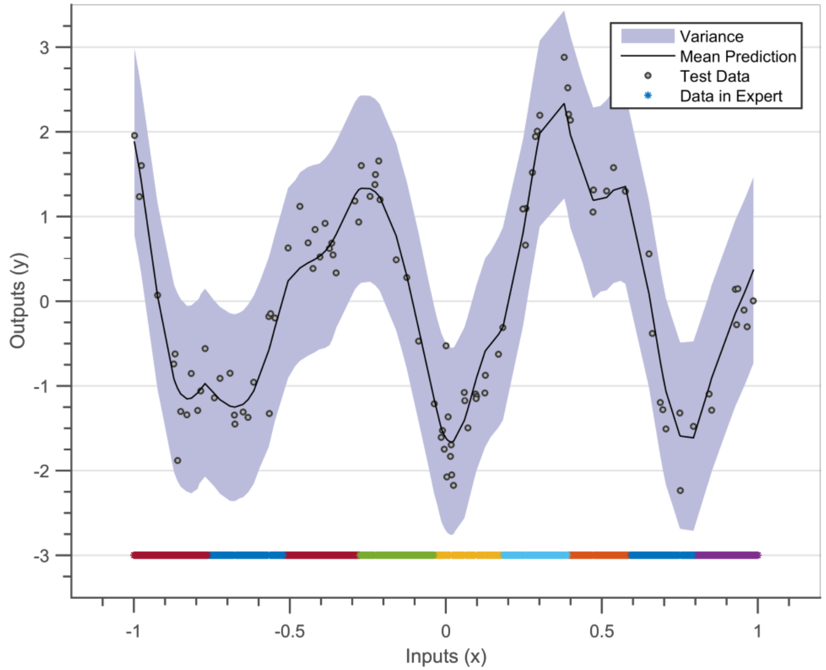
Distributed GP

Comparison between two approximation methods

- $f(x) = GP(0, K_{se}(1, 0.1))$
- $y(x) = f(x) + (0.3)*rand$
- $N = 1000$
- 10 fold Cross-validation



Inducing inputs



Distributed GP

Multi-Output Gaussian Process /Non-Linear

Case 3: When the outputs are related by a known relation/function

Given: $f_1 = g(f_2, x)$

If g is **non-linear**

$$p\left(\begin{matrix} f_1 \\ f_2 \end{matrix}\middle| M\right) = GP\left(\begin{matrix} 0 \\ 0 \end{matrix}\middle| \begin{matrix} g(g(K_{22}, x_2), x_1) & g(K_{22}, x_2) \\ g(K_{22}, x_1) & K_{22} \end{matrix}\right)$$

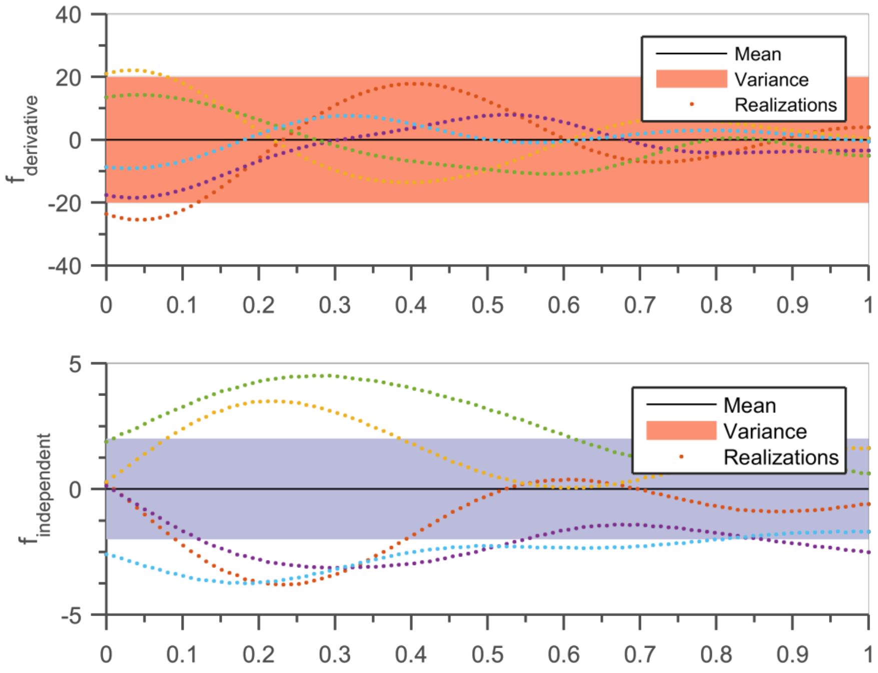
Is not a gaussian

$$p\left(\begin{matrix} f_1 \\ f_2 \end{matrix}\middle| M\right) = GP\left(\begin{matrix} 0 \\ 0 \end{matrix}\middle| \begin{matrix} LK_{22}L^T & LK_{22} \\ K_{22}L^T & K_{22} \end{matrix}\right) \{L = \frac{\partial g}{\partial y_2}\}$$

First order taylor series expansion

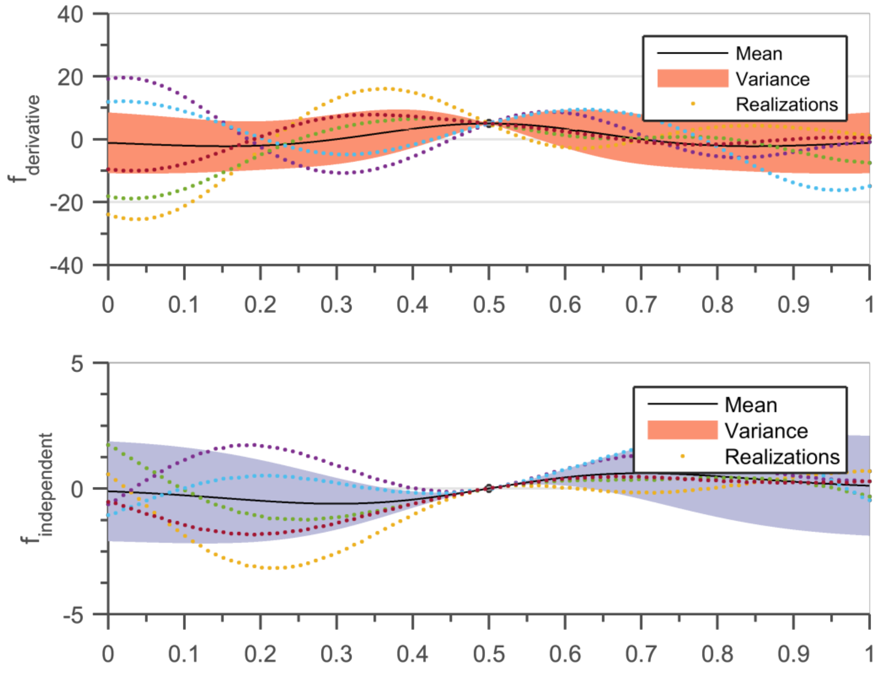
Differential relation

$$f_{derivative} = \frac{df_{independent}}{dx}$$

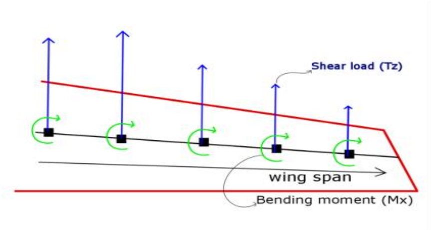


$$f_{derivative}\Big|_{x=0} = 5$$

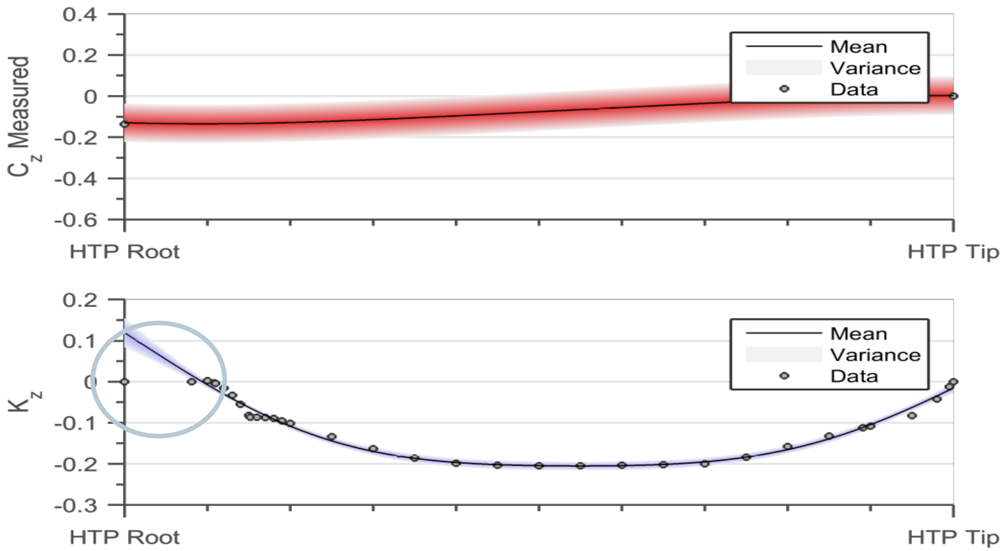
$$f_{independent}\Big|_{x=0} = 0$$

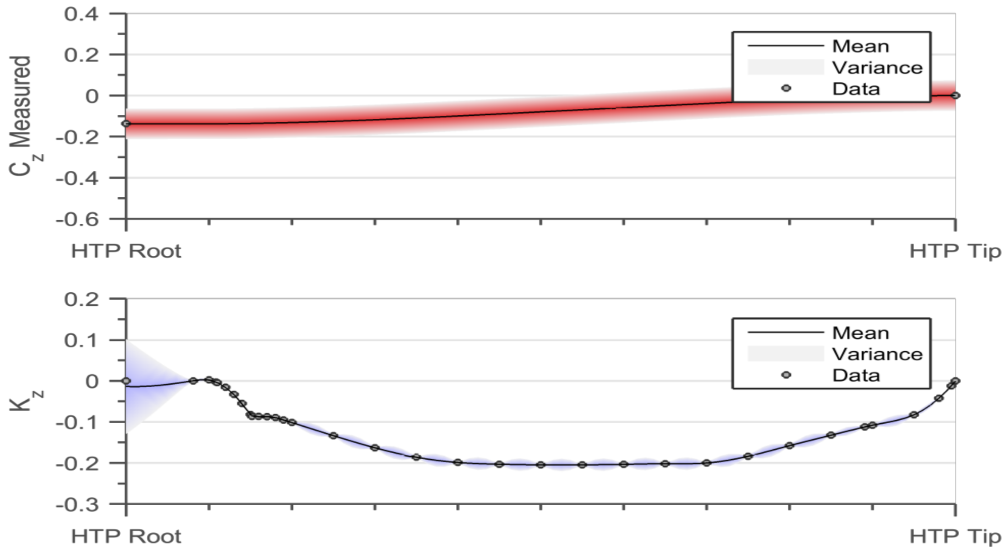


Flight test - Relationship between Cz and Kz



$$C_z(\eta) = \int_{\eta_{edge}}^{\eta_{root}} k_Z(\eta) d\eta$$





$$k_{SE}(x, x', \theta) = \theta_1^2 exp[-\frac{d^2}{2\theta_2^2}]$$

$$k_{Mat2}(x, x', \theta) = \theta_1^2 \left(1 + \frac{\sqrt{5}d}{\theta_2} + \frac{5d^2}{3\theta_2^2}\right) exp\left[-\frac{\sqrt{5}d}{\theta_2}\right]$$

Kriging vs Gaussian Process

Topics	Kriging	Gaussian Process
Original Formulation	[Matheron, G., 1967] [Petrie, Gregg M., et al. 2002]	
New formulation	[Kennedy, Marc C., et al 2001]	[Rasmussen et al 2006]
Multi-fidelity	[Kennedy, et al 2000]	
Optimization	[Jones, Donald R. 2001]	[Mockus, Jonas. 2012]
Constrained Optimization	[Sasena, 2002]	[Maatouk, Hassan, et al 2017]
GEK	[Liu, Weiyu. 2003]	[Solak, Ercan, et al. 2003]
Dimensionality Reduction	[Bouhlel, M., et al. 2016]	[Tripathy, Rohit et al 2016]
Parallel approximations		[Deisenroth, et al 2015].
Automatic kernel		[Duvenaud 2014]; [Wilson 2011]