Uncertainty Quantification for Offshore Wind Energy

Jack Kennedy
Kevin Wilson, Daniel Henderson

RSS Belfast, 2019

5/9/19

Introduction

- Mathematical modelling used across all of science

[1] Zitrou et al. [2013]

Introduction

- Mathematical modelling used across all of science
- We use **simulators** to perform **computer experiments** - vital in engineering contexts

[1] Zitrou et al. [2013]

Introduction

- Mathematical modelling used across all of science
- We use **simulators** to perform **computer experiments** - vital in engineering contexts
- Sophisticated simulators often computationally intensive

[1] Zitrou et al. [2013]

Introduction

- Mathematical modelling used across all of science
- We use **simulators** to perform **computer experiments** - vital in engineering contexts
- Sophisticated simulators often computationally intensive
- **Example**: stochastic windfarm simulation, used to model large, offshore windfarms [1]

[1] Zitrou et al. [2013]

Introduction

- Mathematical modelling used across all of science
- We use **simulators** to perform **computer experiments** - vital in engineering contexts
- Sophisticated simulators often computationally intensive
- **Example**: stochastic windfarm simulation, used to model large, offshore windfarms [1]
  - Walney Extension - 87 turbines / 659 MW capacity
  - London Array - 175 turbines / 630 MW capacity
  - Floating offshore wind farms

[1] Zitrou et al. [2013]

The Simulator

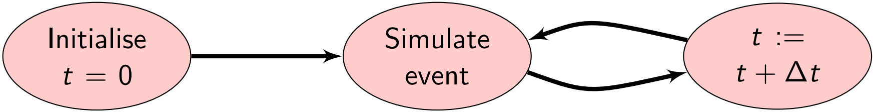
- Simulator developed as decision support tool to better understand how new offshore wind technologies react to harsh environments

The Simulator

- Simulator developed as decision support tool to better understand how new offshore wind technologies react to harsh environments
- Windfarm represented by complex Monte Carlo simulation

The Simulator

- Simulator developed as decision support tool to better understand how new offshore wind technologies react to harsh environments
- Windfarm represented by complex Monte Carlo simulation



- Inputs: Hundreds - some known; many uncertain

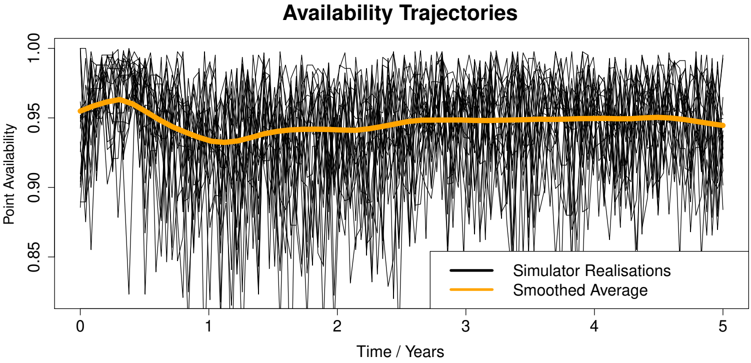
Known	Uncertain
Windfarm topology	Cable failure rates
Simulation length	Hazard function parameters
	Learning Rate

Simulator Description

- Output: Availability trajectories

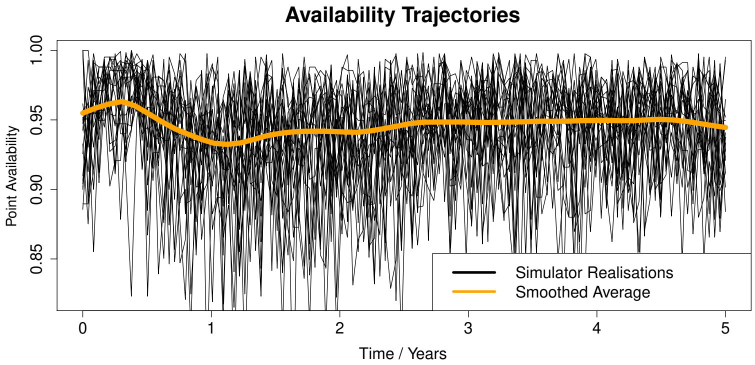
Simulator Description

- Output: Availability trajectories



Simulator Description

- Output: Availability trajectories



- Average over time period gives "fixed term availability" or $A(\boldsymbol{x})$.
- Profitable windfarm satisfies $A(\boldsymbol{x}) > 97\%$

Simulator Description

Simulator prone to uncertainty

Simulator Description

Simulator prone to uncertainty

- **Epistemic uncertainty**: simulator inputs are unknown
  $\rightarrow$ quantify uncertainty and propagate

Simulator Description

Simulator prone to uncertainty

- **Epistemic uncertainty**: simulator inputs are unknown
  $\rightarrow$ quantify uncertainty and propagate
- **Aleatory uncertainty**: simulator is stochastic
  $\rightarrow$ investigate with many simulations

Simulator Description

Simulator prone to uncertainty

- **Epistemic uncertainty**: simulator inputs are unknown
  $\rightarrow$ quantify uncertainty and propagate
- **Aleatory uncertainty**: simulator is stochastic
  $\rightarrow$ investigate with many simulations

No. Turbines	1 Run (s)	10^6 Runs (years)
9	15	0.5
200	120	3.8

- Performing an uncertainty analysis is seemingly impossible!

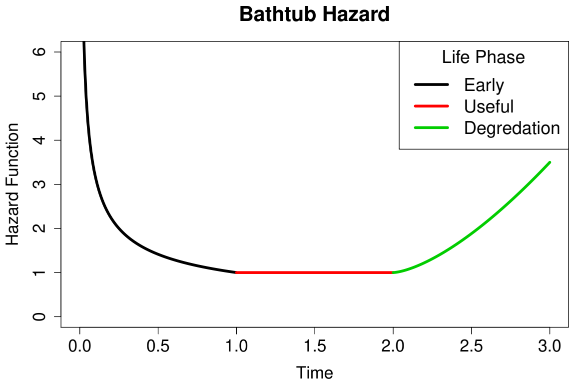
Elicitation

- Aim: Quantify **epistemic** uncertainty from engineers

Elicitation

- Aim: Quantify **epistemic** uncertainty from engineers
- Lifetimes are known to have certain forms at certain times

$$T_i \sim \mathit{Weibull}(\tau_i, \lambda_i, \kappa_i)$$



Emulators

Recall ... 10^6 runs takes ~ 4 years

[2] Kennedy and O'Hagan [2001]

Emulators

Recall ... 10^6 runs takes ~ 4 years

- Utilise Gaussian Process (GP) emulators [2] facilitate computation

[2] Kennedy and O'Hagan [2001]

Emulators

Recall ... 10^6 runs takes ~ 4 years

- Utilise Gaussian Process (GP) emulators [2] facilitate computation
- An emulator is a statistical (Bayesian) **surrogate** model of the simulator or **"model of the model"**.

[2] Kennedy and O'Hagan [2001]

Emulators

Recall ... 10^6 runs takes $\sim$ 4 years

- Utilise Gaussian Process (GP) emulators [2] facilitate computation
- An emulator is a statistical (Bayesian) **surrogate** model of the simulator or **"model of the model"**.
- Crucially: emulators are incredibly cheap to evaluate

[2] Kennedy and O'Hagan [2001]

Emulators

Basic idea of emulation:

- Simulator viewed as an unknown function $y = \eta(\boldsymbol{x})$

Emulators

Basic idea of emulation:

- Simulator viewed as an unknown function $y = \eta(\boldsymbol{x})$

Prior specification:

Emulators

Basic idea of emulation:

- Simulator viewed as an unknown function $y = \eta(\boldsymbol{x})$

Prior specification:

$$\eta(\cdot) \sim \mathcal{GP}\left\{m_0(\cdot), C_0(\cdot, \cdot)\right\}$$

Emulators

Basic idea of emulation:

- Simulator viewed as an unknown function $y = \eta(\boldsymbol{x})$

Prior specification:

$$\eta(\cdot) \sim \mathcal{GP}\{m_0(\cdot), C_0(\cdot,\cdot)\}$$

$$C_0(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma^2 \exp\left\{ -\sum_{k=1}^{K} \left(\frac{x_i^k - x_j^k}{\theta_k}\right)^2 \right\} + \lambda^2 \delta_{ij}$$

Emulators

- We observe training data $(\boldsymbol{Y}_t, \boldsymbol{x}_t)$; want to infer $\boldsymbol{Y}_p = \eta(\boldsymbol{x}_p)$

Emulators

- We observe training data $(\boldsymbol{Y}_t, \boldsymbol{x}_t)$; want to infer $\boldsymbol{Y}_p = \eta(\boldsymbol{x}_p)$

$$\begin{pmatrix} \boldsymbol{Y}_t \\ \boldsymbol{Y}_p \end{pmatrix} \sim \mathcal{N}\left\{ \begin{pmatrix} \boldsymbol{m}_0(x_t) \\ \boldsymbol{m}_0(x_p) \end{pmatrix}, \begin{pmatrix} \Sigma_{TT} & \Sigma_{TP} \\ \Sigma_{PT} & \Sigma_{PP} \end{pmatrix} \right\}$$

Emulators

- We observe training data $(\boldsymbol{Y}_t, \boldsymbol{x}_t)$; want to infer $\boldsymbol{Y}_p = \eta(\boldsymbol{x}_p)$

$$\begin{pmatrix} \boldsymbol{Y}_t \\ \boldsymbol{Y}_p \end{pmatrix} \sim \mathcal{N}\left\{ \begin{pmatrix} \boldsymbol{m}_0(x_t) \\ \boldsymbol{m}_0(x_p) \end{pmatrix}, \begin{pmatrix} \Sigma_{TT} & \Sigma_{TP} \\ \Sigma_{PT} & \Sigma_{PP} \end{pmatrix} \right\}$$

Then

$$\boldsymbol{Y}_p | \boldsymbol{Y}_t = \boldsymbol{y}_t \sim \mathcal{N}\{\boldsymbol{m}^*(\boldsymbol{x}), \Sigma^*\}$$

Where

$$\boldsymbol{m}^*(\boldsymbol{x}) = \boldsymbol{m}_0(\boldsymbol{x}_p) + \Sigma_{PT}\Sigma_{TT}^{-1}(\boldsymbol{y}_t - \boldsymbol{m}_0(\boldsymbol{x}_t))$$

$$\Sigma^* = \Sigma_{PP} - \Sigma_{PT}\Sigma_{TT}^{-1}\Sigma_{TP}$$

Emulation (design)

- Experimental Design: Latin Hypercube over 6 inputs, 50 datapoints:

Farm Characteristics	Initial Hazard Function Parameters
Learning rate	Generator Wearout Onset
Cable Failure rate	Gearbox Wearout Onset
Cable repair rate	Frequency Converter Wearout Onset

Emulation (design)

- Experimental Design: Latin Hypercube over 6 inputs, 50 datapoints:

Farm Characteristics	Initial Hazard Function Parameters
Learning rate	Generator Wearout Onset
Cable Failure rate	Gearbox Wearout Onset
Cable repair rate	Frequency Converter Wearout Onset

- Experiment result: 50000 realisations of $A(\boldsymbol{x})$; 1000 for each $\boldsymbol{x}$

Emulation (design)

- Experimental Design: Latin Hypercube over 6 inputs, 50 datapoints:

Farm Characteristics	Initial Hazard Function Parameters
Learning rate	Generator Wearout Onset
Cable Failure rate	Gearbox Wearout Onset
Cable repair rate	Frequency Converter Wearout Onset

- Experiment result: 50000 realisations of $A(\boldsymbol{x})$; 1000 for each $\boldsymbol{x}$
- Will emulate the mean and variance of $A(\boldsymbol{x})$

Emulation

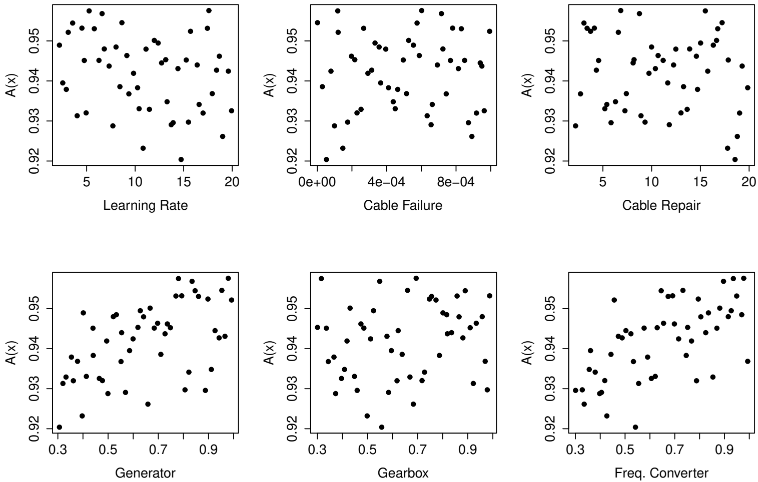


Figure 1: Mean of $A(\boldsymbol{x})$ against inputs

Emulation

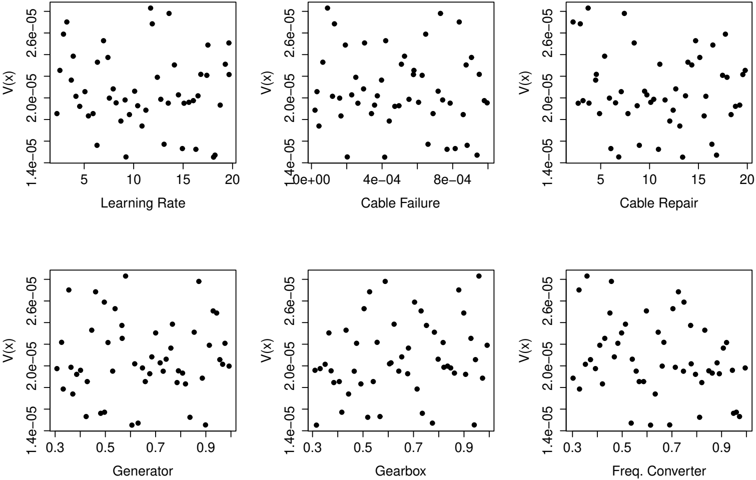


Figure 2: Variance of $A(\boldsymbol{x})$ against inputs

Emulation

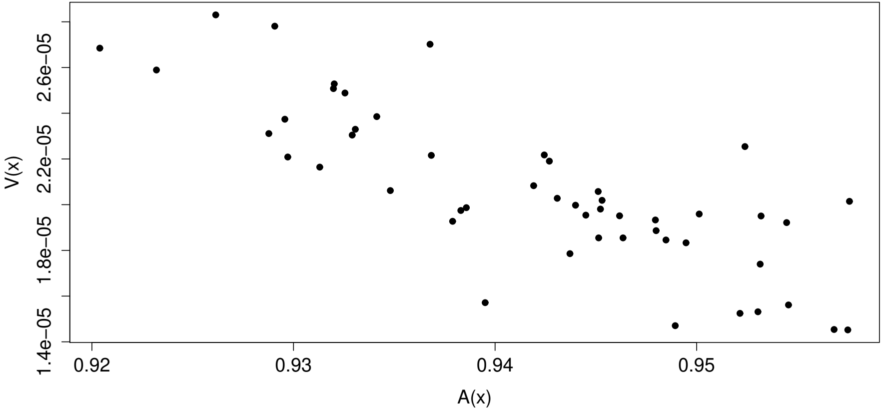


Figure 3: Variance of $A(\boldsymbol{x})$ against the mean of $A(\boldsymbol{x})$

Fitting the Emulator

We can emulate the windfarm simulator with a 4 stage approach

Fitting the Emulator

We can emulate the windfarm simulator with a 4 stage approach

1. Generate training data: $(\boldsymbol{y}_t, \boldsymbol{x}_t)$; compute means and variances

Fitting the Emulator

We can emulate the windfarm simulator with a 4 stage approach

1. Generate training data: $(\boldsymbol{y}_t, \boldsymbol{x}_t)$; compute means and variances
2. Construct emulator for $\Phi^{-1}\big(\mathrm{E}\{A(\boldsymbol{x})\}\big)$

Fitting the Emulator

We can emulate the windfarm simulator with a 4 stage approach

1. Generate training data: $(\boldsymbol{y}_t, \boldsymbol{x}_t)$; compute means and variances
2. Construct emulator for $\Phi^{-1}\left(\mathrm{E}\{A(\boldsymbol{x})\}\right)$
3. Construct emulator for $\log\left\{\mathrm{Var}\left[A(\boldsymbol{x})|\mathrm{E}(A(\boldsymbol{x}))\right]\right\}$

Fitting the Emulator

We can emulate the windfarm simulator with a 4 stage approach

1. Generate training data: $(\boldsymbol{y}_t, \boldsymbol{x}_t)$; compute means and variances
2. Construct emulator for $\Phi^{-1}\big(\mathrm{E}\{A(\boldsymbol{x})\}\big)$
3. Construct emulator for $\log\left\{\mathrm{Var}\big[A(\boldsymbol{x})|\mathrm{E}(A(\boldsymbol{x}))\big]\right\}$
4. Reintroduce the stochasticity: $A(\boldsymbol{x}) \sim Beta\big\{a(\boldsymbol{x}), b(\boldsymbol{x})\big\}$

Fitting the Emulator

We can emulate the windfarm simulator with a 4 stage approach

1. Generate training data: $(\boldsymbol{y}_t, \boldsymbol{x}_t)$; compute means and variances
2. Construct emulator for $\Phi^{-1}\big(\mathrm{E}\{A(\boldsymbol{x})\}\big)$
3. Construct emulator for $\log\left\{\mathrm{Var}\big[A(\boldsymbol{x})|\mathrm{E}(A(\boldsymbol{x}))\big]\right\}$
4. Reintroduce the stochasticity: $A(\boldsymbol{x}) \sim Beta\big\{a(\boldsymbol{x}), b(\boldsymbol{x})\big\}$

Fitting the Emulator

We can emulate the windfarm simulator with a 4 stage approach

1. Generate training data: $(\boldsymbol{y}_t, \boldsymbol{x}_t)$; compute means and variances
2. Construct emulator for $\Phi^{-1}\big(\mathrm{E}\{A(\boldsymbol{x})\}\big)$
3. Construct emulator for $\log\left\{\mathrm{Var}\big[A(\boldsymbol{x})|\mathrm{E}(A(\boldsymbol{x}))\big]\right\}$
4. Reintroduce the stochasticity: $A(\boldsymbol{x}) \sim Beta\big\{a(\boldsymbol{x}), b(\boldsymbol{x})\big\}$

Result: Super cheap way to obtain realisations of windfarm's fixed term availability

Emulator Validation

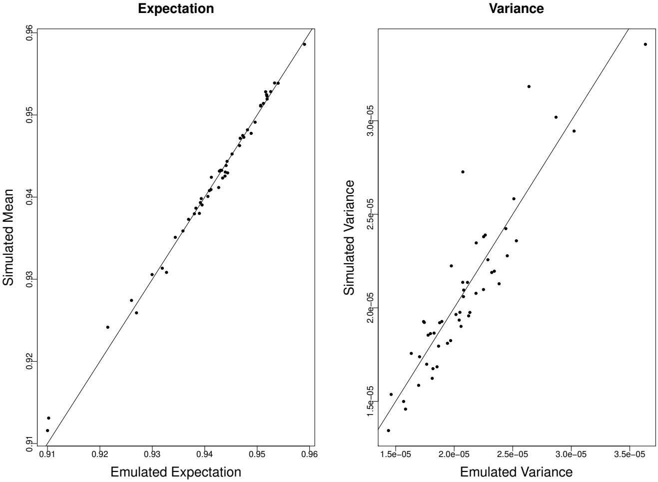


Figure 4: Observed vs predicted values of simulator mean and variance on independently generated validation data

Emulator Validation

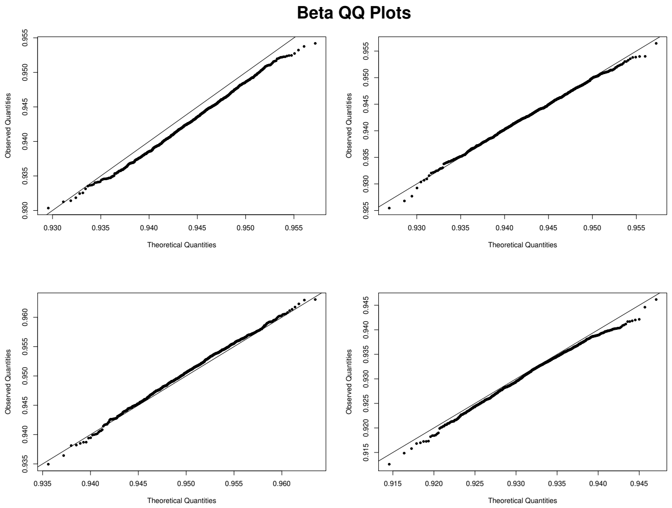


Figure 5: Beta QQ plots based on emulated mean and variance of validation data

Conclusions and Further Work

- Simulator takes $\sim 4 - 5$ hours to obtain distribution of $A(\boldsymbol{x})$ for one $\boldsymbol{x}$. Emulator can accurately approximate many such distributions in under one second.

Conclusions and Further Work

- Simulator takes $\sim 4-5$ hours to obtain distribution of $A(\boldsymbol{x})$ for one $\boldsymbol{x}$. Emulator can accurately approximate many such distributions in under one second.
- Utilise cheaper versions of the model to construct better emulators at a fixed computational budget

Conclusions and Further Work

- Simulator takes $\sim 4-5$ hours to obtain distribution of $A(\boldsymbol{x})$ for one $\boldsymbol{x}$. Emulator can accurately approximate many such distributions in under one second.
- Utilise cheaper versions of the model to construct better emulators at a fixed computational budget
- Perform an elicitation on input parameters to assess $\mathrm{P}(A > 97\%)$ - probably for floating offshore wind turbines

Conclusions and Further Work

- Simulator takes $\sim 4-5$ hours to obtain distribution of $A(\boldsymbol{x})$ for one $\boldsymbol{x}$. Emulator can accurately approximate many such distributions in under one second.
- Utilise cheaper versions of the model to construct better emulators at a fixed computational budget
- Perform an elicitation on input parameters to assess $\mathrm{P}(A > 97\%)$ - probably for floating offshore wind turbines
- In performing the UQ, we need to identify important inputs and elicit uncertainty over these inputs; this can be done using expert knowledge and mathematical techniques

References

MC Kennedy and A O'Hagan. Bayesian calibration of computer models. *Journal Of The Royal Statistical Society Series B-Statistical Methodology*, 63:425–450, 2001. ISSN 1369-7412.

Athena Zitrou, Tim Bedford, Lesley Walls, Kevin Wilson, and Keith Bell. Availability growth and state-of-knowledge uncertainty simulation for offshore wind farms. In *22nd ESREL conference 2013*, September 2013. URL `https://strathprints.strath.ac.uk/45377/`.