Bayesian Emulation and Calibration of a Dynamic Epidemic Model for H1N1 Influenza

Marian Farah[1]

Paul Birrell[1], Stefano Conti[2], Daniela De Angelis[1,2]

[1]MRC Biostatistics Unit, Cambridge, UK
[2]Health Protection Agency, London, UK

ICERM Bayesian Nonparametrics Workshop
September 19, 2012

Motivation

- **Tracking** and **predicting** the behavior of an emerging epidemic is *essential* for a prompt public health response.
- **Inferential goals**:
  - What is happening? i.e., **real-time estimation** of the epidemic **parameters**.
  - What is going to happen next? i.e., **forecasting** the (short-term) **evolution** of the epidemic.
  - What happened? i.e., **"reconstructing"** the epidemic by **estimating** its **parameters** and **evolution dynamics**.
- **Noisy time-series** data coming from **different sources**.

Introduction

Epidemic modeling

- **Transmission model**:

$$S(t) \xrightarrow{\text{getting infected}} E(t) \xrightarrow{\text{latent period}} I(t) \xrightarrow{\text{infectious period}} R(t)$$

- Transmission depends on the **virulence**, the **mixing patterns in the population**, and the **transition rates** among the S, E, I, and R states.

- Transmission **dynamics** are typically described by a system of **differential equations**.

Birrell et al. (2011) H1N1 model

$$S(t) \longrightarrow E(t) \longrightarrow I(t) \longrightarrow R(t)$$

Birrell et al. (2011) H1N1 model

$$S(t) \longrightarrow E(t) \longrightarrow I(t) \longrightarrow R(t)$$

$\downarrow$ incubation

Expected # of symptomatic individuals

$\downarrow$ propensity to consult doctor

Expected # of doctor consultations

$\downarrow$ delay in reporting

Expected # of reported cases, $\mu(\eta, t)$

Birrell et al. (2011) H1N1 model

$$S(t) \longrightarrow E(t) \longrightarrow I(t) \longrightarrow R(t)$$

$\downarrow$ incubation

Expected # of symptomatic individuals

$\downarrow$ propensity to consult doctor

Expected # of doctor consultations

$\downarrow$ delay in reporting

Expected # of reported cases, $\mu(\boldsymbol{\eta}, t)$

- $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_6)$: underlying **parameters** of the epidemic.
- Proportion of symptomatic cases, propensity to consult, exponential growth rate, expected infectious period, a measure of the initial number of infected individuals, population interaction parameters.

Computational challenge

- The **likelihood** of reported data, $z(t)$, $t = 1, \ldots, T$, depends on μ.
- $$p(\eta \mid z_{\{1:T\}}, \mu) \propto \prod_{t=1}^{T} p\big(z(t); \mu(\eta, t)\big) \times p(\eta)$$
- $\mu(\eta, t)$ must be computed at **every MCMC** iteration.
- μ is **computationally expensive**.

Computational challenge

- The **likelihood** of reported data, $z(t)$, $t = 1, \ldots, T$, depends on μ.
- $$p(\eta \mid z_{\{1:T\}}, \mu) \propto \prod_{t=1}^{T} p\big(z(t); \mu(\eta, t)\big) \times p(\eta)$$
- $\mu(\eta, t)$ must be computed at **every MCMC** iteration.
- μ is **computationally expensive**.
- What about an efficient estimate?

Computer simulator

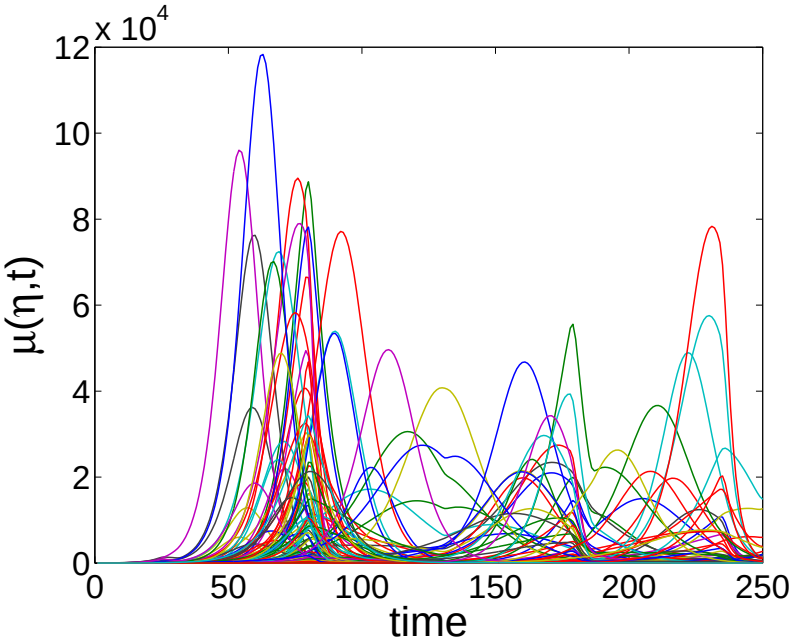
specify inputs $\eta = X$ → run code → outputs

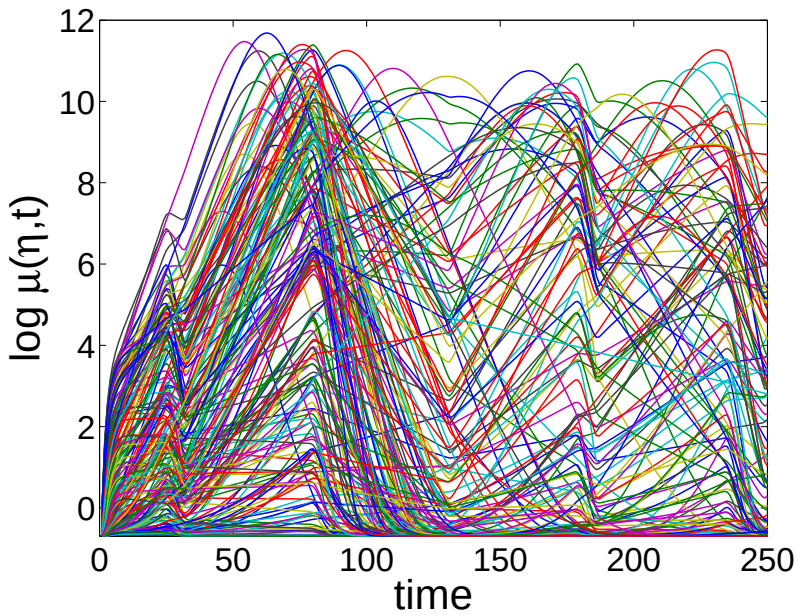
$$X = \begin{pmatrix} x_{1,1} & \dots & x_{1,6} \\ x_{2,1} & \dots & x_{2,6} \\ \vdots & \vdots & \vdots \\ x_{n,1} & \dots & x_{n,6} \end{pmatrix} \rightarrow \boxed{\text{Birrell et al. (2011)}} \rightarrow \begin{matrix} \mu(\mathbf{x}_1, 1), \dots, \mu(\mathbf{x}_1, T) \\ \mu(\mathbf{x}_2, 1), \dots, \mu(\mathbf{x}_2, T) \\ \vdots \\ \mu(\mathbf{x}_n, 1), \dots, \mu(\mathbf{x}_n, T) \end{matrix}$$

Computer simulator

specify inputs $\eta = X$ | run code | outputs

$$X = \begin{pmatrix} x_{1,1} & \dots & x_{1,6} \\ x_{2,1} & \dots & x_{2,6} \\ \vdots & \vdots & \vdots \\ x_{n,1} & \dots & x_{n,6} \end{pmatrix} \rightarrow \boxed{\text{Birrell et al. (2011)}} \rightarrow \begin{matrix} \mu(\mathbf{x}_1, 1), \dots, \mu(\mathbf{x}_1, T) \\ \mu(\mathbf{x}_2, 1), \dots, \mu(\mathbf{x}_2, T) \\ \vdots \\ \mu(\mathbf{x}_n, 1), \dots, \mu(\mathbf{x}_n, T) \end{matrix}$$





Calibration and Emulation

- **Calibration**: (e.g., Higdon et al., 2004)
  Posterior inference for $\boldsymbol{\eta}$ through the simulator, μ, and "field" observed data $z(t)$,

$$\text{Observed} = \text{Reality} + \text{Error}$$

$$\underset{z}{\text{Observed}} = \underset{\mu}{\text{Simulator}} + \underset{b}{\text{bias}} + \text{Error}$$

- $$p(\boldsymbol{\eta}, b \mid z_{\{1:T\}}, \mu) \propto \prod_{t=1}^{T} p\big(z(t);\, \mu(\boldsymbol{\eta}, t) + b\big) \times p(\boldsymbol{\eta}) p(b)$$

- **Emulation**: (e.g., Kennedy and O'Hagan, 2000)
  Estimating a slow computer *simulator* output, μ, using *fast* statistical model (an *emulator*), say $\hat{\mu}$.

Calibration and Emulation

- **Idea**: (e.g., Bayarri et al., 2007a)
  Replace the *slow* simulator output, μ, with the *fast* emulator estimation, $\hat{\mu}$, and obtain posterior inference for $\boldsymbol{\eta}$ through

- $$p(\boldsymbol{\eta}, b \mid z_{\{1:T\}}, \hat{\mu}) \propto \prod_{t=1}^{T} p\big(z(t);\, \hat{\mu}(\boldsymbol{\eta}, t) + b\big) \times p(\boldsymbol{\eta}) p(b)$$

Marian Farah
Biostatistics
Cambridge

Emulation and calibration of dynamic models

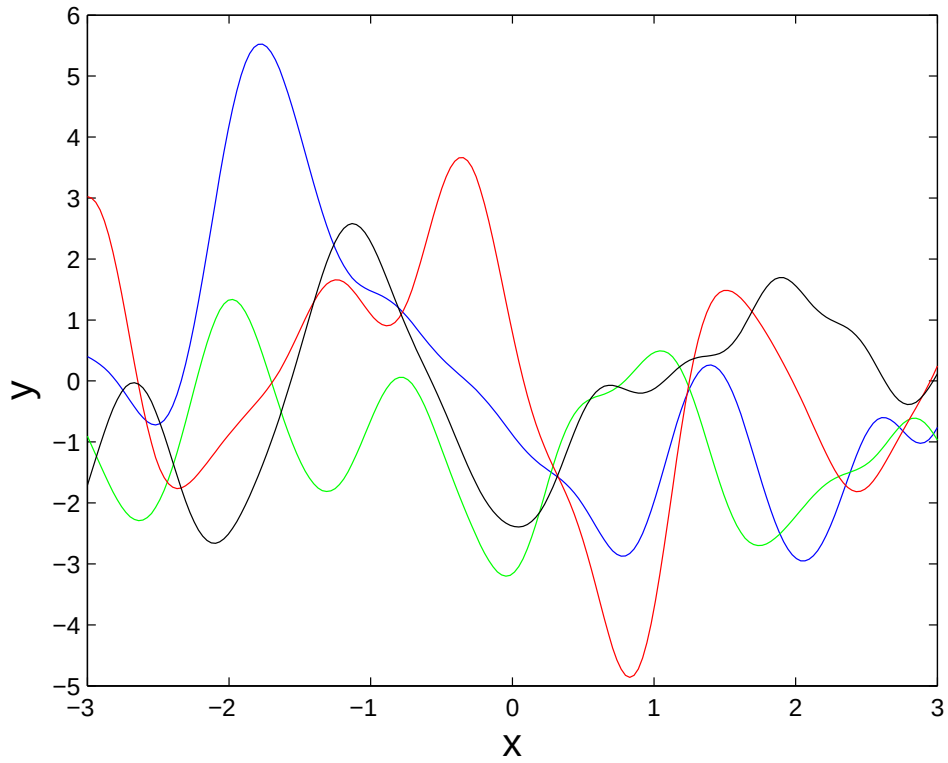
Emulation review

- A **deterministic** computer simulator is a **function** $f(\cdot)$ that maps **input** x to a unique **output** $y = f(x)$.
- The function $f(\cdot)$ is treated as **unknown** and given a **prior**.
- **Likelihood**: data are **runs of the simulator**, given a **design** over the **input space**, e.g., Latin Hypercube.
- **Emulator**: the **posterior** (predictive) distribution of $f(\cdot)$.

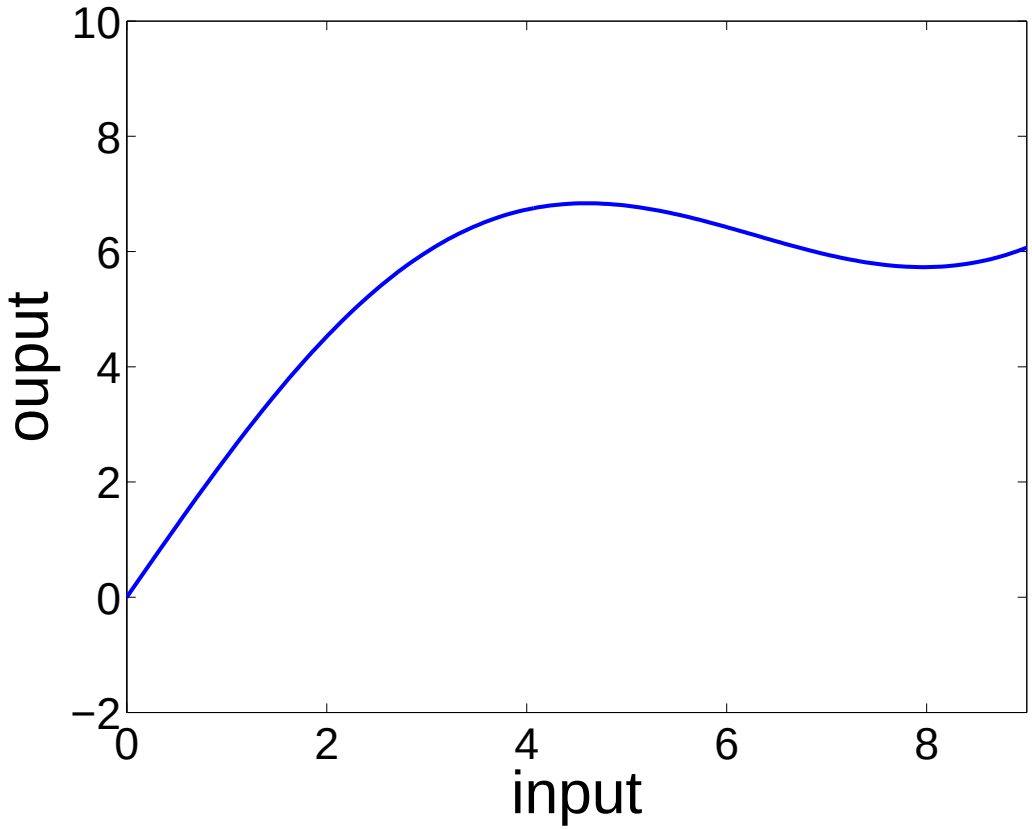
The Gaussian process

$$y(x) \sim GP(\, m(x), v\, c(x, x')\,)$$

$m(\cdot)$, v, and $c(\cdot, \cdot)$ are the mean, variance, & correlation function (e.g., Neal 1998; Rasmussen & Williams 2006).



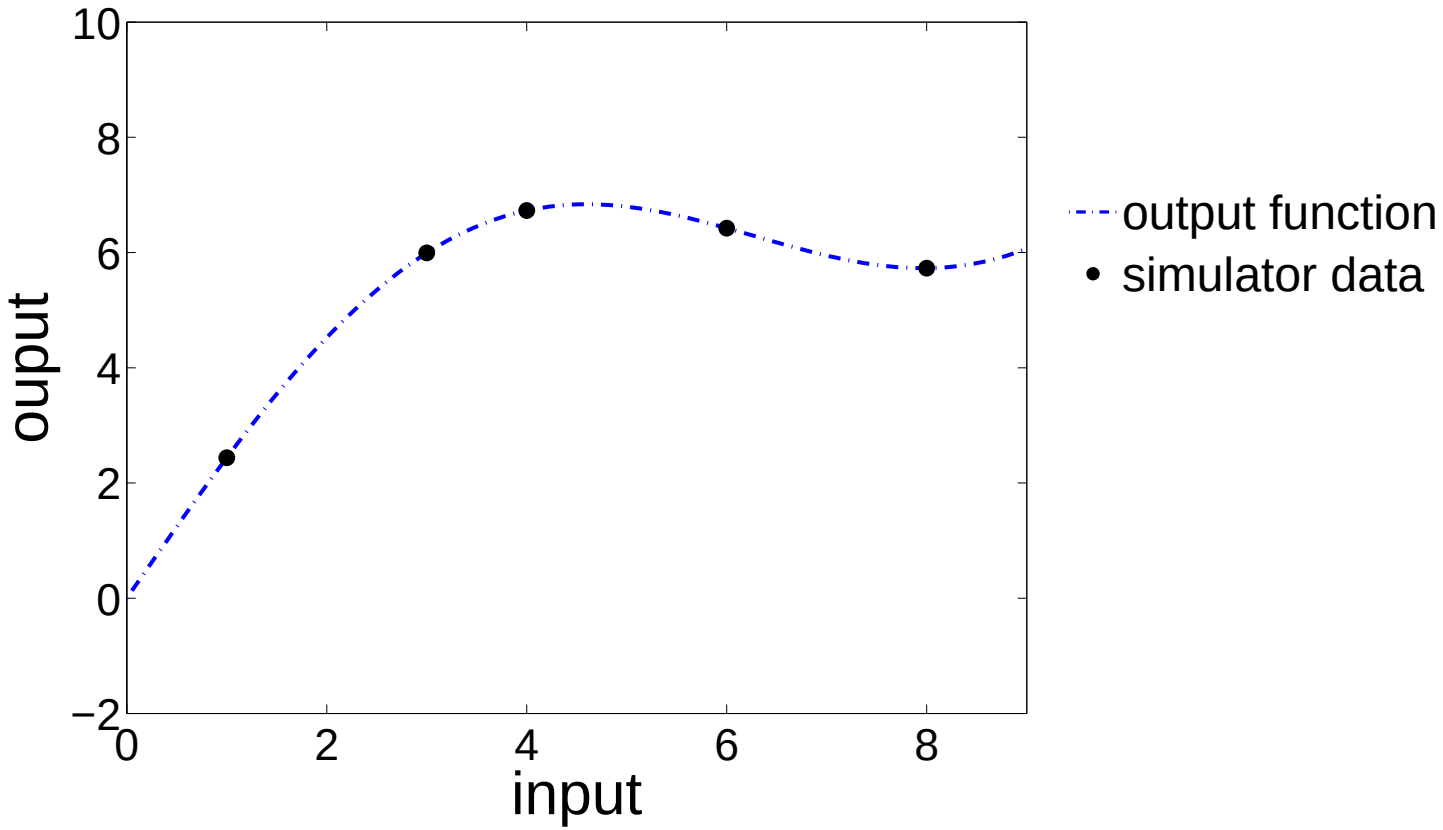
Toy example



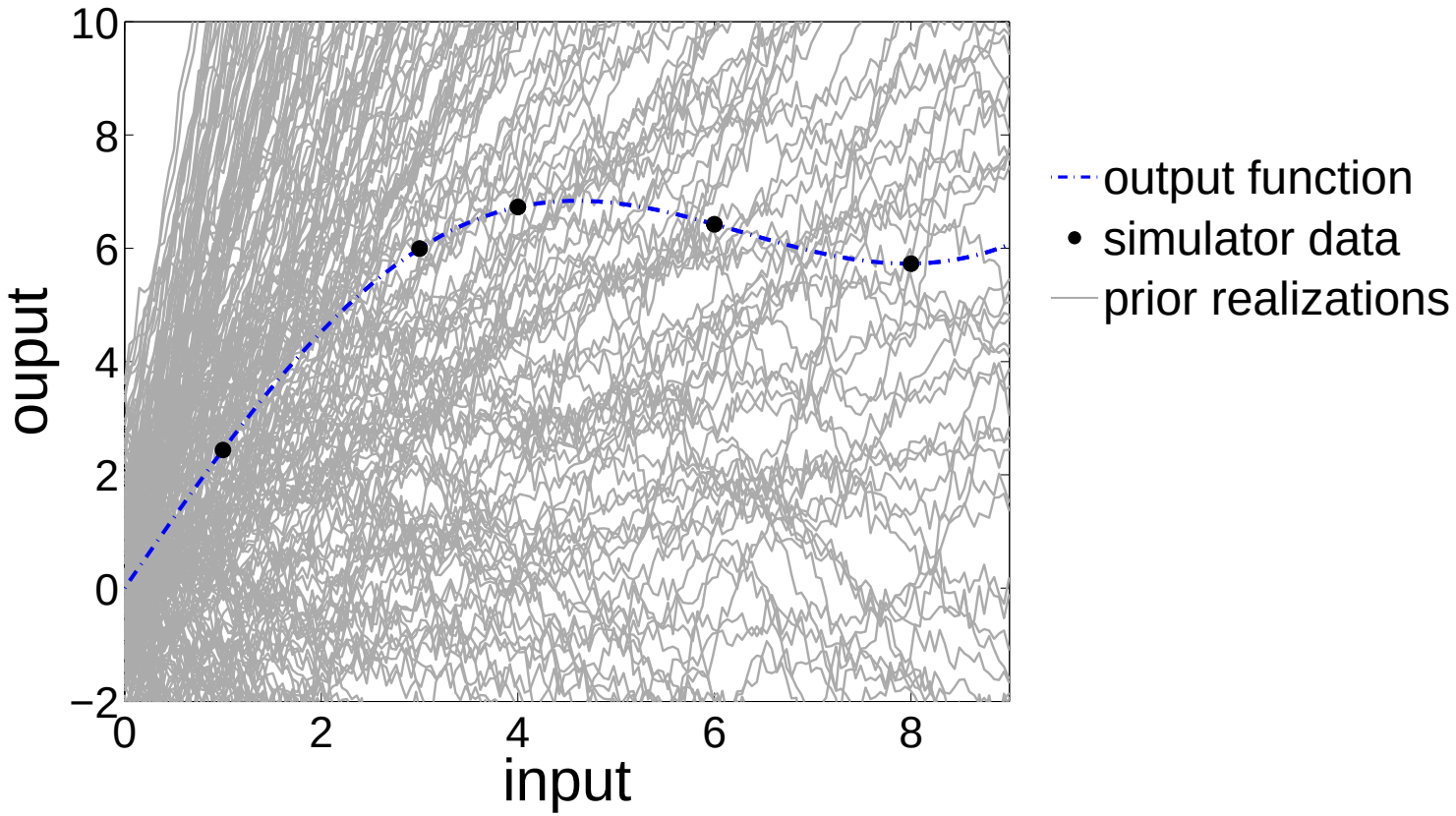
output function

$f(x) = x + 3sin(x/2)$

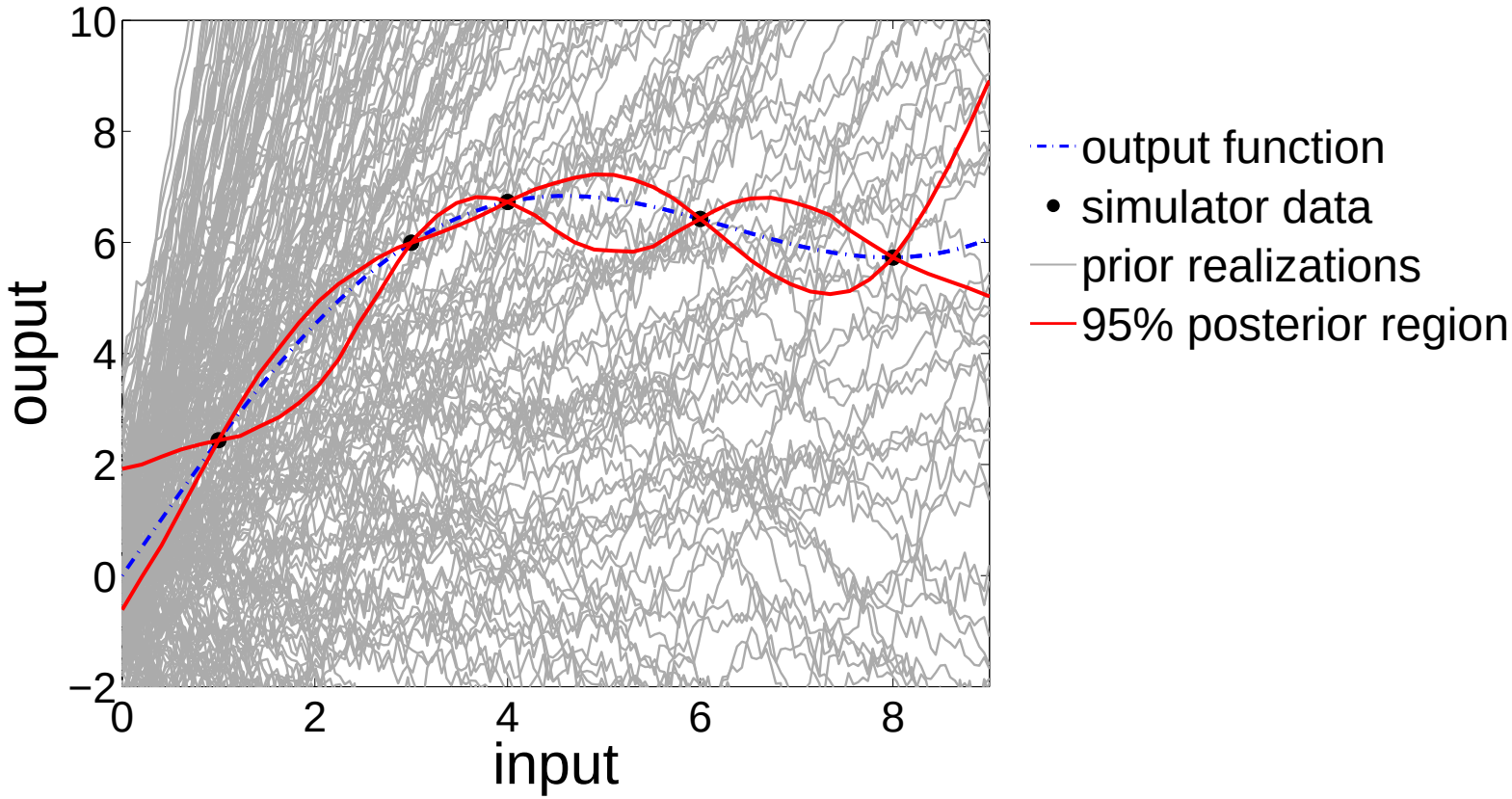
Toy example


10
8
6
4
2
0
−2
0
2
4
6
8
ouput
input
output function
simulator data

Toy example


10
8
6
4
2
0
−2
ouput
0
2
4
6
8
input
output function
simulator data
prior realizations

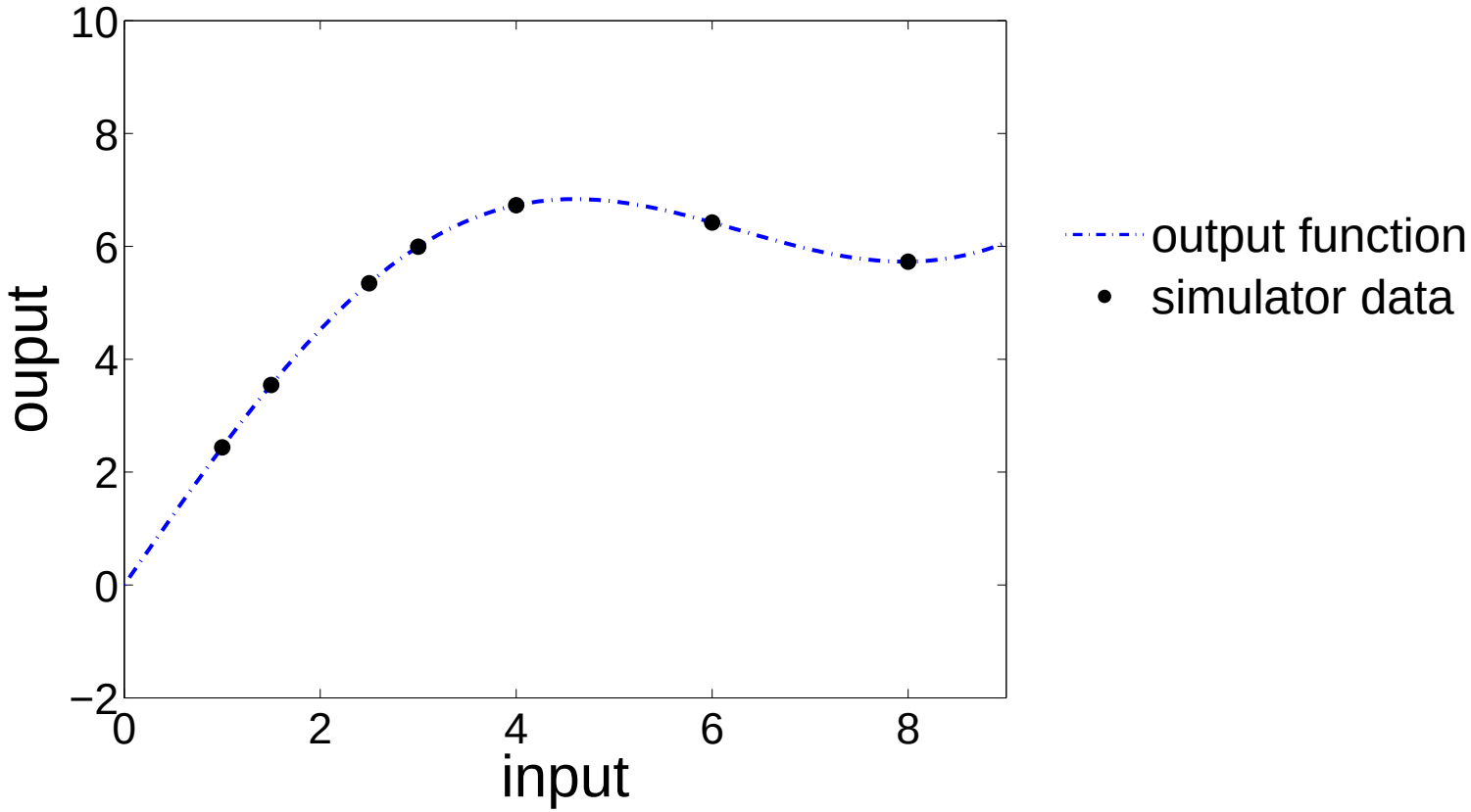
Toy example


10
8
6
4
2
0
−2
ouput
0
2
4
6
8
input
output function
simulator data
prior realizations
95% posterior region

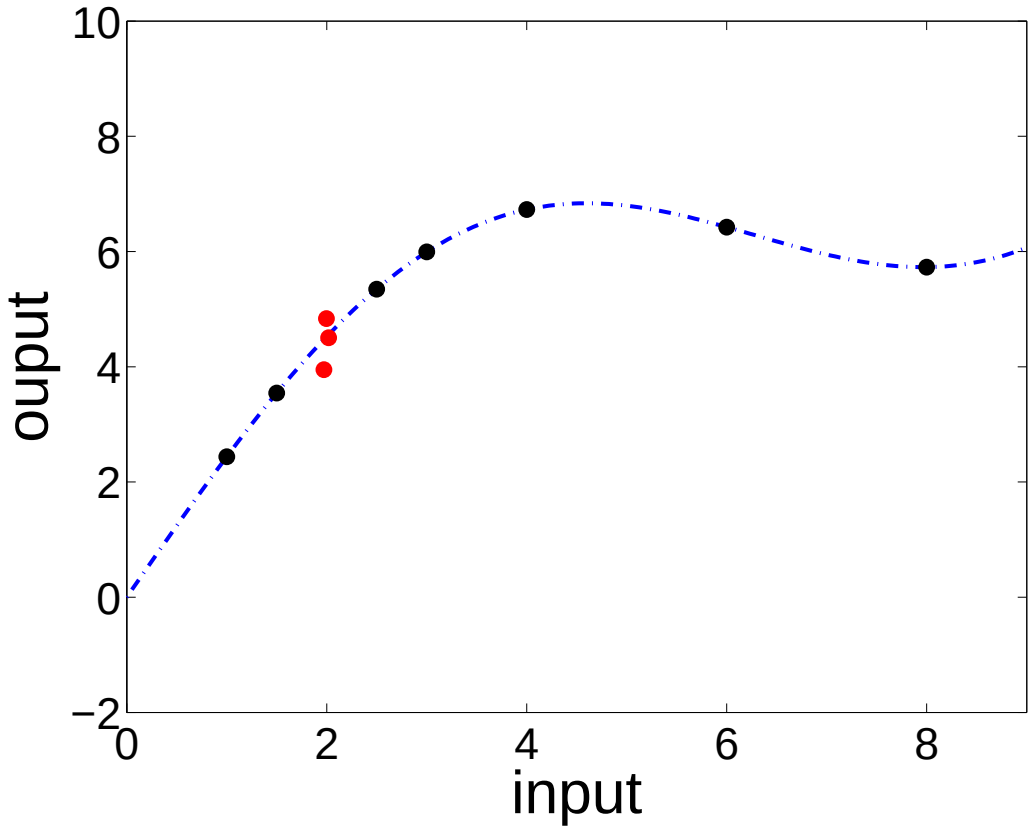
Calibration review

- Simulator: Specify $\mathbf{x} \rightarrow f(\mathbf{x})$.
- For $\mathbf{x} = \eta$, $f(\eta)$ simulates a physical system.
- η is **uncertain**.
- **Calibration**: solving the **inverse-problem**, i.e., $\eta \mid \mathbf{z}, f(\cdot)$.
- If $f(\cdot)$ is computationally expensive, it is emulated.
  - **Priors** for η and $f(\cdot)$.
  - **Likelihood**: data come from **field** observations and **simulator** runs.

Toy example


10
8
6
4
2
0
−2
ouput
0
2
4
6
8
input
output function
simulator data

Toy example

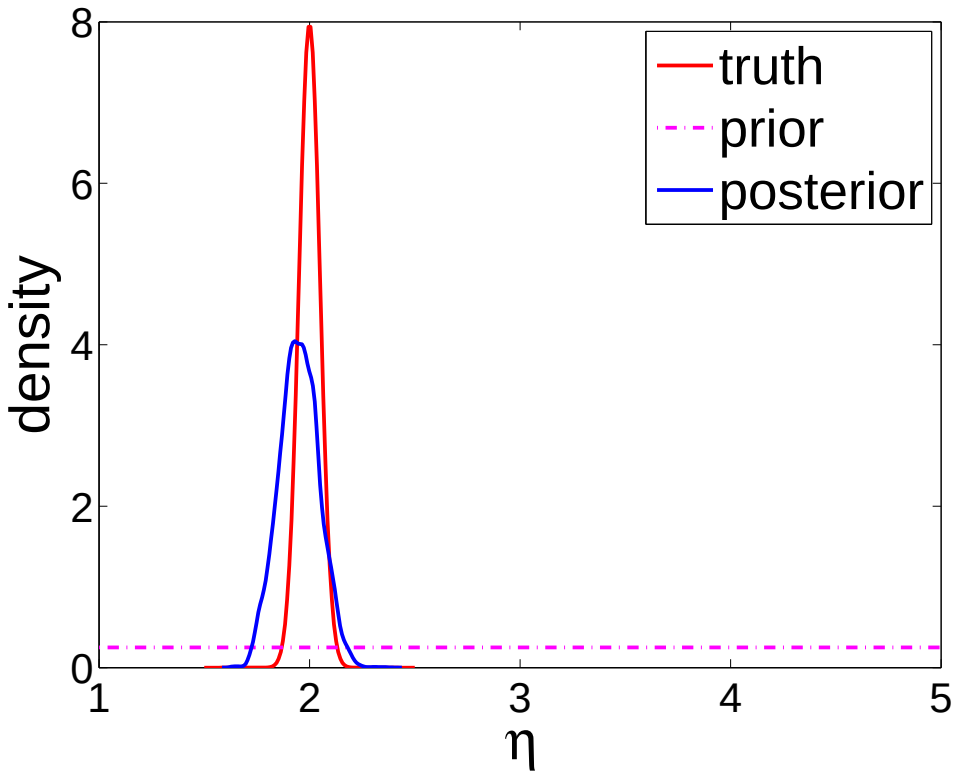


output function
- simulator data
- field data

$z \sim N(f(\eta), \sigma^2 = 0.3^2)$

$\eta \sim N(2, 0.05^2)$

Toy example



- Assuming σ^2 is known.

Dynamic emulation

- $y_t(x_i) = f(x_i, t)$ is the simulator output at input point x_i and time t.

$$
\begin{array}{lcccc}
x_1 & \longrightarrow & y_1(x_1), & y_2(x_1), & \ldots, & y_T(x_1) \\
x_2 & \longrightarrow & y_1(x_2), & y_2(x_2), & \ldots, & y_T(x_2) \\
\vdots & & \vdots & \vdots & & \vdots \\
x_n & \longrightarrow & y_1(x_n), & y_2(x_n), & \ldots, & y_T(x_n)
\end{array}
$$

Dynamic emulation

- $y_t(x_i) = f(x_i, t)$ is the simulator output at input point x_i and time t.

$$\begin{array}{lcccc} x_1 & \longrightarrow & y_1(x_1), & y_2(x_1), & \ldots, & y_T(x_1) \\ x_2 & \longrightarrow & y_1(x_2), & y_2(x_2), & \ldots, & y_T(x_2) \\ \vdots & & \vdots & \vdots & & \vdots \\ x_n & \longrightarrow & y_1(x_n), & y_2(x_n), & \ldots, & y_T(x_n) \end{array}$$

- Need to model **three** types of **interdependencies**:
  1. over the **input space**.
  2. over time **within** each **time series**.
  3. **across series** of different input points.

Dynamic emulation

- **Modeling dependence over the input space alone**

  Typically using a **Gaussian process** prior for outputs.

$$y(x) \sim GP(\, m(x), v\, c(x, x')\,)$$

Dynamic emulation

- **Modeling dependence over the input space alone**

  Typically using a **Gaussian process** prior for outputs.

$$y(x) \sim GP(\, m(x), v\, c(x, x')\,)$$

- **Modeling dependence for a single time series**

  Typically, **TVAR(p)** model is used; e.g., $\mathbf{p = 1}$,

$$y_t(x) = \phi_t\, y_{t-1}(x) + \epsilon_t(x), \quad \epsilon_t(x) \sim N(0,\, v_t),$$
$$\phi_t = \phi_{t-1} + \omega_t, \qquad \omega_t \sim N(0,\, w_t).$$

Dynamic emulation

- **Linking across time series for different inputs** using a multivariate TVAR(p) model (Liu and West, 2009),

$$\begin{pmatrix} y_t(x_1) \\ \vdots \\ y_t(x_n) \end{pmatrix} = \begin{pmatrix} y_{t-1}(x_1) & \cdots & y_{t-p}(x_1) \\ \vdots & \ddots & \vdots \\ y_{t-1}(x_n) & \cdots & y_{t-p}(x_n) \end{pmatrix} \begin{pmatrix} \phi_{1,t} \\ \vdots \\ \phi_{p,t} \end{pmatrix} + \begin{pmatrix} \epsilon_t(x_1) \\ \vdots \\ \epsilon_t(x_n) \end{pmatrix}$$

Dynamic emulation

- **Linking across time series for different inputs** using a multivariate TVAR(p) model (Liu and West, 2009),

$$\begin{pmatrix} y_t(x_1) \\ \vdots \\ y_t(x_n) \end{pmatrix} = \begin{pmatrix} y_{t-1}(x_1) & \cdots & y_{t-p}(x_1) \\ \vdots & \ddots & \vdots \\ y_{t-1}(x_n) & \cdots & y_{t-p}(x_n) \end{pmatrix} \begin{pmatrix} \phi_{1,t} \\ \vdots \\ \phi_{p,t} \end{pmatrix} + \begin{pmatrix} \epsilon_t(x_1) \\ \vdots \\ \epsilon_t(x_n) \end{pmatrix}$$

$$\mathrm{Cov}\big(\epsilon_t(x_i), \epsilon_t(x_j)\big) = v_t\, c(x_i, x_j)$$

  - $c(x_i, x_j)$ is the (i, j) element in the $n \times n$ **correlation** matrix induced by the **Gaussian process**.

Dynamic emulation

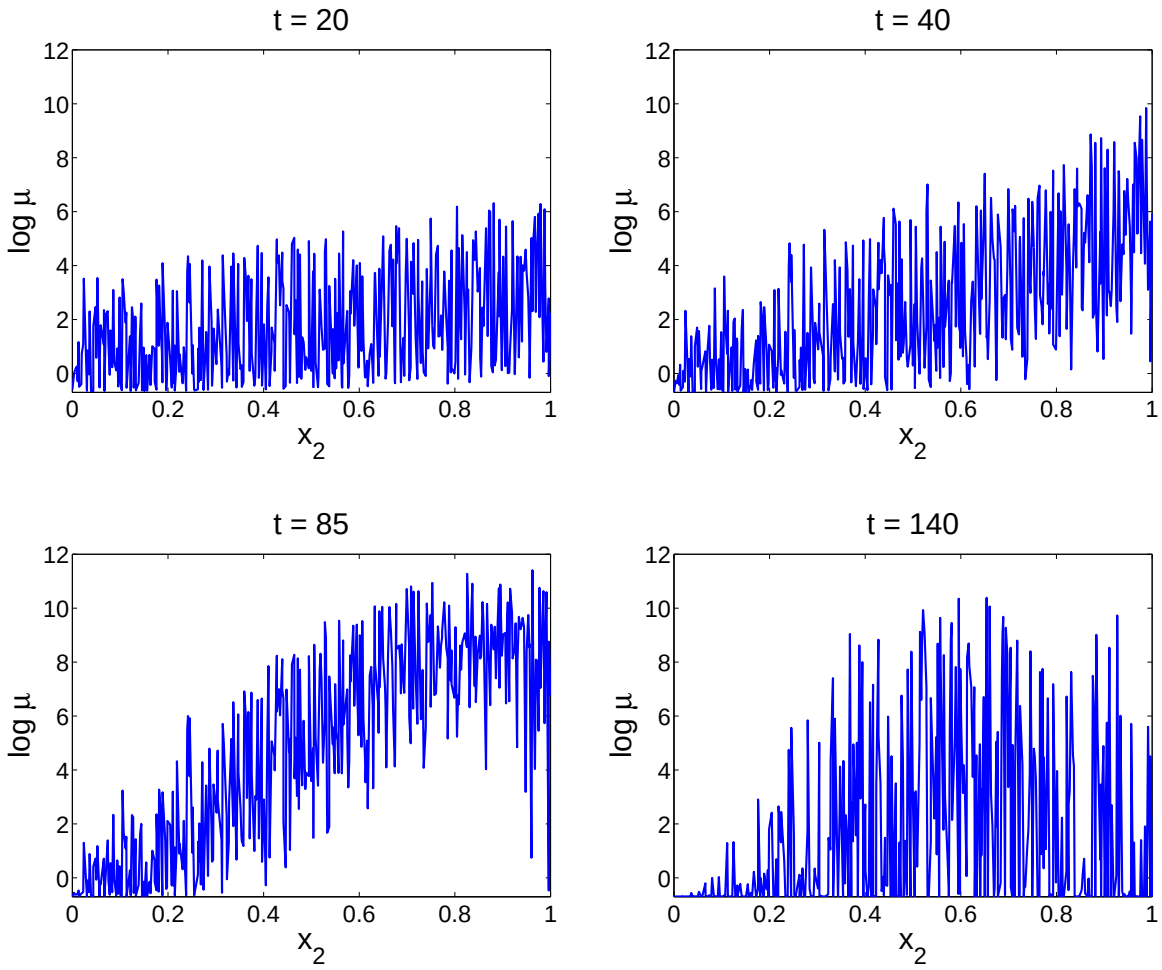
- **Linking across time series for different inputs** using a multivariate TVAR(p) model (Liu and West, 2009),

$$\begin{pmatrix} y_t(x_1) \\ \vdots \\ y_t(x_n) \end{pmatrix} = \begin{pmatrix} y_{t-1}(x_1) & \cdots & y_{t-p}(x_1) \\ \vdots & \ddots & \vdots \\ y_{t-1}(x_n) & \cdots & y_{t-p}(x_n) \end{pmatrix} \begin{pmatrix} \phi_{1,t} \\ \vdots \\ \phi_{p,t} \end{pmatrix} + \begin{pmatrix} \epsilon_t(x_1) \\ \vdots \\ \epsilon_t(x_n) \end{pmatrix}$$

$$\mathrm{Cov}\big(\epsilon_t(x_i), \epsilon_t(x_j)\big) = v_t\, c(x_i, x_j)$$

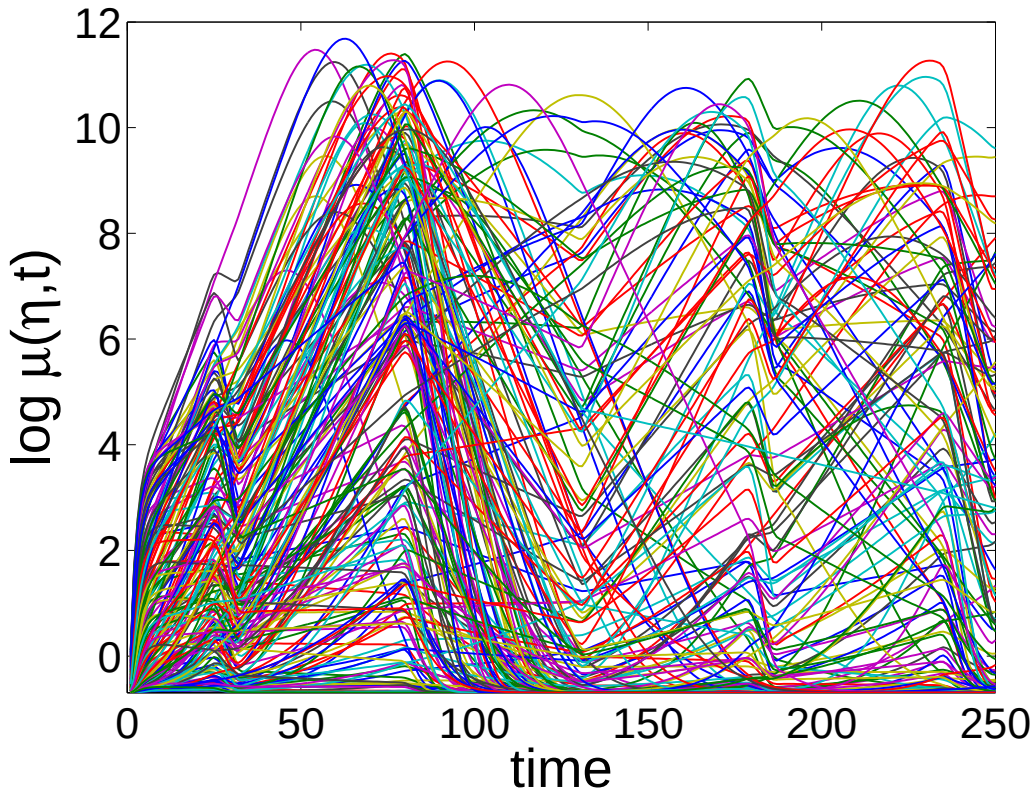
  - $c(x_i, x_j)$ is the (i, j) element in the $n \times n$ **correlation** matrix induced by the **Gaussian process**.
  - $\boldsymbol{\phi}_t = \boldsymbol{\phi}_{t-1} + \boldsymbol{\omega}_t$, where $\boldsymbol{\phi}_t = (\phi_{1t}, \ldots, \phi_{pt})'$.

Birrell et al. (2011) simulator



- x_2: Exponential growth rate.

Birrell et al. (2011) simulator



Dynamic emulation

- **Extending Liu and West (2009)**
  - Modeling **input-dependent trends**:

$$y_t(x) = \phi_t\, y_{t-1}(x) + h(x)\beta_t + \epsilon_t$$

  - Modeling **systematic temporal trend**:

$$y_t(x) = \theta_t + \phi_t\, y_{t-1}(x) + h(x)\beta_t + \epsilon_t$$

$$\begin{pmatrix} \boldsymbol{\theta}_t \\ \phi_t \\ \beta_t \end{pmatrix} = \begin{pmatrix} \theta_{t-1} \\ \phi_{t-1} \\ \beta_{t-1} \end{pmatrix} + \begin{pmatrix} \omega_{1t} \\ \omega_{2t} \\ \omega_{3t} \end{pmatrix}$$

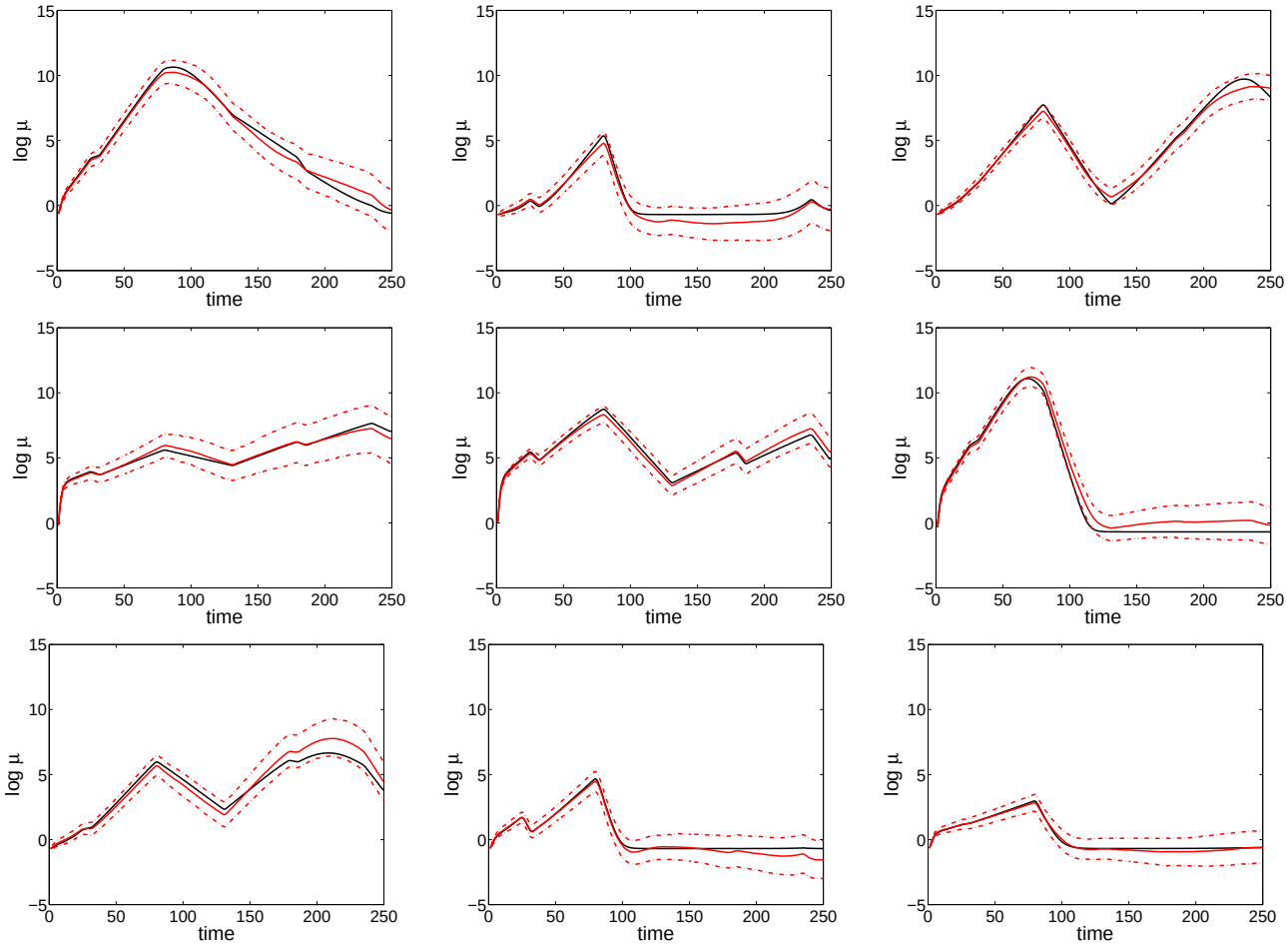
  - Posterior inference through Forward-Filtering Backward-Sampling.

Calibration

- **Two** sources of data:
  - **Simulator** data: $D^s = \{(y_t, \mathbf{x}); t = 1, \ldots, T\}$. Model parameters are *specified* as inputs $\mathbf{x}$.
  - **"Field"** observed epidemic data $D^F = \{z_t; t = 1, \ldots, T\}$. Model parameters, η, are unknown.
- **Two-stage** calibration (e.g., Bayarri et al., 2007b)
  - **Stage 1**: Estimate the emulator model parameters using only D^s.
  - **Stage 2**: Model z_t using a parametric distribution centered on the emulator model. Then, conditional on stage 1, estimate $p(\eta \mid D^F, D^s)$.

Results

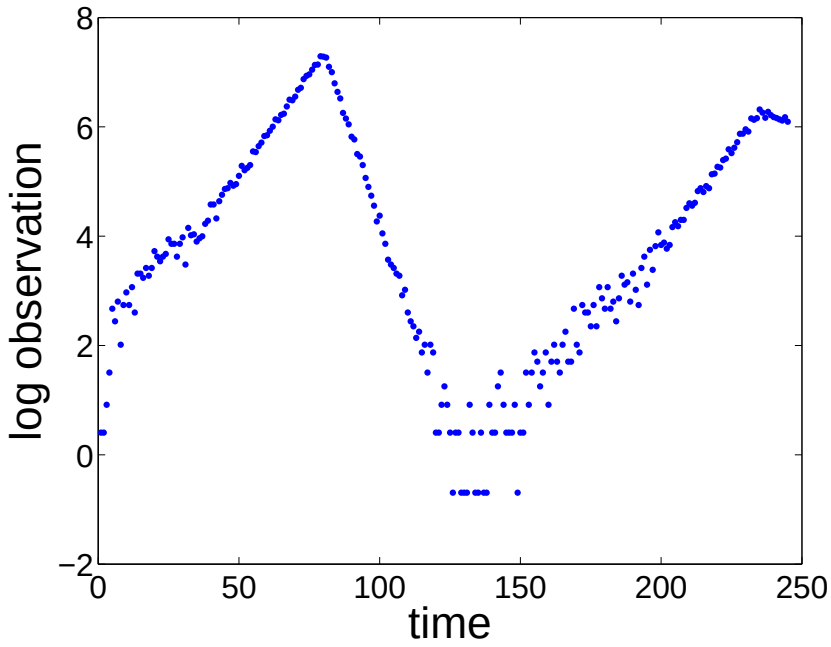
Validating the emulator



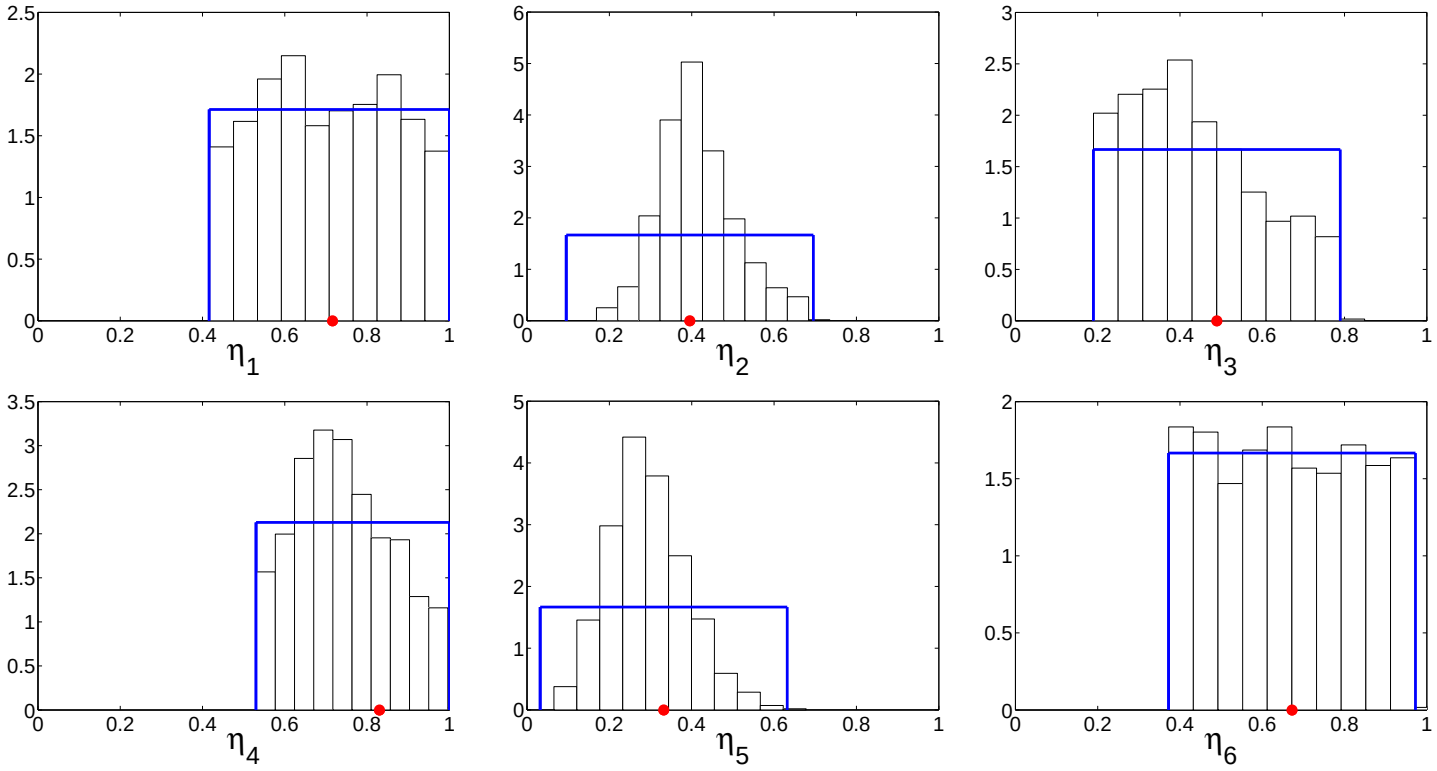
- Simulation runs (black), emulator's median & 95% region (red).
- Plots based on a MVTVAR(1) and Gaussian correlation function.

Calibration

- Generated synthetic epidemic data.
- Set $\eta = \eta_0$. Then, $z \sim \text{Poisson}\big(\mu(\eta_0, t)\big)$



Calibration



- η_2 is exponential growth rate, and η_5 is effect of summer holiday on population interaction.

Discussion

- What we have done:
  - **Estimation** of epidemic dynamics by combining a **statistical emulator** with **reported epidemic data**.
  - **Dynamic emulation** through modeling dependencies across time and epidemic parameter space.
- Still to do:
  - Consider different **age** groups in the population.
  - Incorporate **additional sources** of information.
  - Real-time calibration and forecasting using epidemic data.
  - ...

- What we have done:
  - **Estimation** of epidemic dynamics by combining a **statistical emulator** with **reported epidemic data**.
  - **Dynamic emulation** through modeling dependencies across time and epidemic parameter space.
- Still to do:
  - Consider different **age** groups in the population.
  - Incorporate **additional sources** of information.
  - Real-time calibration and forecasting using epidemic data.
  - ...

Thank you!

References:

- Bayarri, M., Berger, J., Paulo, R., Sacks, J., Cafeo, J., Cavendish, J., Lin, C., and Tu, J. (2007a), "A framework for validation of computer models," *Technometrics*, 49, 138–154.
- Bayarri, M. J., Berger, J. O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R., Paulo, R., Sacks, J., and Walsh, D. (2007b), "Computer model validation with functional output," *Annals of Statistics*, 35, 1874–1906.
- Birrell, P. J., Ketsetzis, G., Gay, N. J., Cooper, B. S., Presanis, A. M., Harris, R. J., Charlett, A., Zhang, X.-S., White, P. J., Pebody, R. G., and De Angelis, D. (2011), "Bayesian modeling to unmask and predict infuenza A/H1N1pdm dynamics in London," *Proceedings of the National Academy of Sciences*.
- Kennedy, M. C. and O'Hagan, A. (2000), "Predicting the output from a complex computer code when fast approximations are available," *Biometrika*, 87, 1–13.
- Liu, F. and West, M. (2009), "A Dynamic Modelling Strategy for Bayesian Computer Model Emulation," *Bayesian Analysis*, 4, 393–412.