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

Abstract

In this paper, we develop a Bayesian framework for parameter estimation of a computationally expensive dynamic epidemic model using time series epidemic data. Specifically, we work with a model for A/H1N1 influenza, which is implemented as a deterministic computer *simulator*, taking as input the underlying epidemic parameters and calculating the corresponding time series of reported infections. To obtain Bayesian inference for the epidemic parameters, the simulator is embedded in the likelihood for the reported epidemic data. However, the simulator is computationally slow, making it impractical to use in Bayesian estimation where a large number of simulator runs is required. We propose an efficient approximation to the simulator using an *emulator*, a statistical model that combines a Gaussian process prior for the output function of the simulator with a dynamic linear model for its evolution through time. This modeling framework is both flexible and tractable, resulting in efficient posterior inference through Markov Chain Monte Carlo (MCMC). The proposed dynamic emulator is then used in a calibration procedure to obtain posterior inference for the parameters of the influenza epidemic.

KEY WORDS: Emulation; Calibration; Gaussian process; Dynamic Linear Models; Epidemic model.

1 Introduction

An emerging epidemic is a major public health concern. Not only does it increase demand upon health services, but it may also result in substantial socioeconomic problems and loss of life. Increasingly, mechanistic models are playing an important role in strategies for epidemic management, approximating the underlying dynamics, and enabling inference on key epidemic characteristics (Brauer and Castillo-Chavez, 2011; Korostil et al., 2012). In the attempt to control an epidemic, the goal of model development is to provide computationally efficient estimation of key parameters in order to allow timely assessment and prediction of the epidemic evolution as new data become available. In this paper, we address this computationally efficient parameter estimation problem.

The work here is motivated by inferential challenges posed by a dynamic epidemic model developed by Birrell et al. (2011) to reconstruct the 2009 A/H1N1 influenza epidemic in London. The model consists of two components: one for disease transmission, and the other for the reporting of symptomatic illness to healthcare facilities. Parameter estimation is carried out, using data from multiple sources, in a Bayesian framework implemented via Markov Chain Monte Carlo (MCMC). The multiplicity of data, combined with the model complexity, makes the evaluation of the model computationally expensive. This expense is exacerbated by the need for the MCMC algorithm to evaluate the model numerous times at each iteration, seriously limiting the ability for this approach to provide practical online inference.

To address this problem, we turn to the computer models literature. Computer models (henceforth *simulators*) are developed in virtually all scientific fields to encode sophisticated mathematical representations of particular physical processes (O'Hagan, 2006). Inputs to the simulator include the underlying parameters of the physical process under consideration, and the output is a quantity of interest representing some behavior of the physical process. In practice, some simulators are severely hampered by long run-times, making analyses that require a large number of simulator runs infeasible. To circumvent the incurred computational burden, a fast surrogate (an *emulator*) to the computationally expensive simulator is used in the analysis instead. The emulator is a statistical model, typically a Gaussian

process (GP), that efficiently estimates the simulator output using a small set of simulator runs (e.g., Sacks et al., 1989; Kennedy and O'Hagan, 2000; Santner et al., 2003).

The transmission-reporting model in Birrell et al. (2011) can be cast as a (dynamic) computationally expensive simulator (henceforth, the BEA simulator), where the inputs are the underlying epidemic parameters, and the output is the time series giving the number of individuals reporting illness. The BEA simulator is deterministic, that is, supplying it repeatedly with the same input point always results in the same time series output. To improve the efficiency of the epidemic parameter estimation, which requires running the BEA simulator at every MCMC iteration, we propose conducting parameter estimation using quick emulator runs. Note that parameter estimation, referred to as "calibration" in the computer model literature, is equivalent to solving the "inverse-problem", i.e., the estimation of the simulator inputs (e.g., Kennedy and O'Hagan, 2000; Higdon et al., 2004; Bayarri et al., 2007a,b). Calibration requires both simulator runs and field observations. Framing the estimation of epidemic parameters as a calibration problem, and using emulation as an efficient surrogate for the computationally expensive BEA simulator are the two major concepts underlying the work presented in this paper.

We begin by emulating the BEA simulator. Here, an appropriate emulator must take into account interdependencies driven by the inputs *within* and *across* time series corresponding to different input points. To capture these interdependencies, we propose an emulator that combines two interlinked structures: a multivariate dynamic linear model (DLM) that captures trends attributed to the inputs as well as complicated temporal interdependencies, including non-stationarity and time-varying stochastic growth; and a GP model that captures the correlation over the input space, and therefore across time series of different inputs. A similar approach is suggested in Liu and West (2009), where a dynamic emulator is constructed using a time-varying autoregressive (TVAR) model (a type of multivariate DLM). However, the emulator in Liu and West (2009) is not suitable for epidemic trajectories as it is designed for time series showing no clear trends, whereas epidemic models produce highly structured trajectories driven by the values of their parameters. In this work, we extend the Liu and West (2009) dynamic emulator to account for possible time-varying input-dependent trends in the time series outputs. Then, we derive a new

calibration approach that utilizes this extended DLM-GP emulator to efficiently estimate the underlying epidemic parameters using observed time series data.

The structure of the paper is as follows. In Section 2, we describe the BEA simulator, our motivating application. In Section 3, we review current approaches to emulation and calibration of dynamic simulators and describe our proposed methods for dynamic emulation and calibration. Section 4 includes results of the emulation and calibration methods using runs from the BEA simulator as well as simulated epidemic data. We conclude with a discussion in Section 5.

2 Motivating Application: The A/H1N1 Epidemic Simulator

The epidemic model in Birrell et al. (2011) was developed as part of the public health response in England and Wales to the global outbreak of a novel A/H1N1 influenza virus in 2009. This model is designed to integrate routinely collected surveillance data and pandemic-specific data to reconstruct the evolution of the unfolding epidemic, providing estimates of key epidemiological parameters. Interpreted as a simulator, the model consists of two main components: one describing the transmission of the virus and the rate at which susceptible individuals become infected (the transmission model), and one describing how these new infections manifest themselves in the healthcare system (the reporting model).

The transmission component is specified as a SEIR-compartment model (e.g., Jacquez, 1996). At any time, the closed population is partitioned into either: a susceptible state, $S(t)$, containing those individuals with no prior immunity to the virus and who are at risk of becoming infected; an exposed state, $E(t)$, including individuals who have been infected but are not yet infectious; an infectious state, $I(t)$; and a removed/recovered state $R(t)$. The total time spent in the E state is referred to as the latent period, and for the I state this is the infectious period. Disease transmission from those in the I state to those in the S state is governed by a number of parameters. These include: the basic reproduction number of the virus, R_0, defined as the average number of secondary infections caused by a primary infection in a fully-susceptible population; I_0, the size of the initial pool of infective individuals; the (time-dependent) mixing matrix $\boldsymbol{M}(t)$, whose $(i,j)^{\text{th}}$ element describes the

time t rate of contact between a member of population stratum i and a member of population stratum j; and the average duration of the latent and infectious periods.

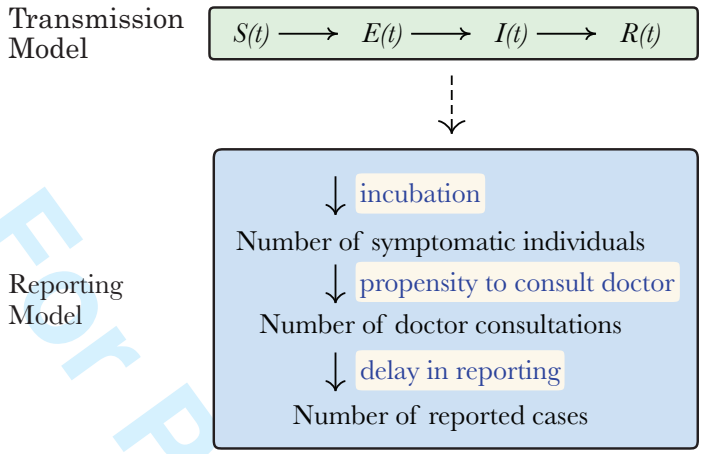

Figure 1: Flow chart of the BEA simulator's modeling components for transmission and reporting of A/H1N1 influenza.

The reporting component takes as input the number of newly infected individuals from the transmission component and outputs the number of reported consultations to the healthcare facilities at each time step t, according to the following process. A proportion of the newly infected individuals develop symptoms after a short incubation period. Of the symptomatic individuals, a further proportion will choose, or be forced due to the severity of the illness, to interact with the healthcare system in some way. Eventually, the number of symptomatic individuals accessing the specific healthcare facilities is reported through the relevant surveillance system with some further delay. The flow of individuals through the coupled transmission-reporting model, forming the BEA simulator, is shown in a flow chart in Figure 1.

To run the BEA simulator, inputs, in the form of the underlying epidemic parameters, are specified. Then, for each input point, the simulator outputs the time series of reported general practice doctor consultations for influenza-like-illness. Here, we work with six inputs, which we denote by $\boldsymbol{x} = (x_1, \ldots, , x_6)'$, described in Table 1 alongside their plausible ranges. Given a specified value of $\boldsymbol{x}$, and at each time step $t \in \{1, \ldots, T\}$, the BEA simulator outputs the number of reported consultations $\mu_t(\boldsymbol{x})$, a deterministic function of $\boldsymbol{x}$. We

denote the time series output by $\boldsymbol{\mu}_{1:T}(\boldsymbol{x}) = (\mu_1(\boldsymbol{x}), \ldots, \mu_T(\boldsymbol{x}))'$. Figure 2 shows trajectories of $\boldsymbol{\mu}_{1:250}$ corresponding to 100 different input points. This 250 days period covers early May through early January, the same time period considered by Birrell et al. (2011). The values of the components of $\boldsymbol{x}$ have a strong impact on the shape of the computed trajectory. For example, on the log-scale (see Figure 2-B), the value of x_2 is closely linked with the initial positive gradient of the output curves; the value of x_5 is linked with the value of the sustained downhill gradient near day 83; and the value of x_6 is associated with the downhill gradient in the short sections corresponding to the three shorter school holidays at around days 30, 180, and 240.

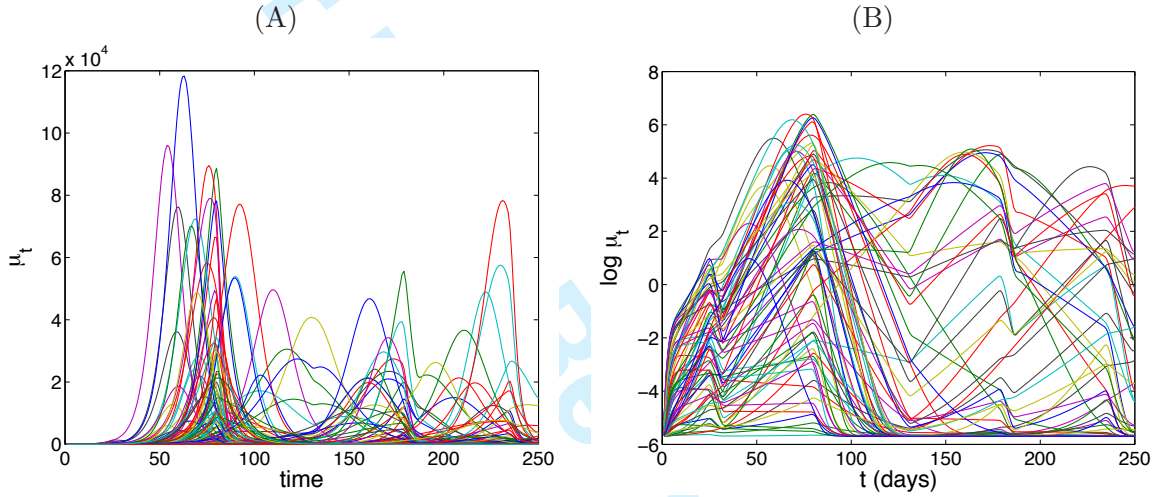

Figure 2: (A) BEA simulator outputs obtained at 100 input points, each computed over 250 days: (A) on the original scale; (B) on the $\log(\mu_t + 0.5) - 5$ scale.

The goal in Birrell et al. (2011) is calibration, that is, using the BEA simulator in combination with observed noisy consultation data, $\boldsymbol{z}_{1:T} = (z_1, \ldots, z_T)'$, to obtain posterior inference for the unknown underlying epidemic parameters, $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_6)'$, which have the same interpretation as the BEA simulator inputs. To link observed consultation data with the BEA simulator, Birrell et al. (2011) model $\boldsymbol{z}_{1:T}$ as noisy measurements of the BEA simulator output. Then, the likelihood $L(\boldsymbol{z}_{1:T} \mid \boldsymbol{\mu}_{1:T}(\boldsymbol{\eta}))$ is combined, in a Bayesian framework, with $p(\boldsymbol{\eta})$, the prior distribution for $\boldsymbol{\eta}$, to obtain the posterior distribution

$$p(\boldsymbol{\eta} \mid \boldsymbol{z}_{1:T}) \propto L(\boldsymbol{z}_{1:T} \mid \boldsymbol{\mu}_{1:T}(\boldsymbol{\eta})) \times p(\boldsymbol{\eta}) \tag{1}$$

Component	Description	Plausible Range
x_1	The proportion of infected individuals who report their illness through consultation with their doctor.	(0, 1)
x_2	The exponential growth rate. This gives the initial growth rate in the number of infected individuals and is closely linked with the value for R_0.	(0.095, 0.119)
x_3	The average infectious period (days).	(3.438, 3.460)
x_4	$= \log(x_1 l_0)$, where l_0 is the initial rate of infection of susceptible individuals. This, as outlined in Birrell et al. (2011), is a reparameterization of the parameter I_0, the size of the initially infective population.	(-13.883, -13.251)
x_5	The proportionate reduction in the population contact rate induced by the over-Summer school holiday period.	(0, 1)
x_6	The proportionate reduction in the population contact rate induced by all other school holidays.	(0, 1)

Table 1: The six inputs to the BEA simulator, the physical quantities they represent, and their plausible ranges.

Evaluation of the likelihood requires evaluation of $\boldsymbol{\mu}_{1:T}(\boldsymbol{\eta})$, which becomes computationally burdensome when making repeated calculation of the likelihood in the MCMC algorithm, limiting the ability for timely inference. Emulation is an obvious approach for efficient approximate inference, as it replaces the computationally expensive simulator runs with a more readily evaluated approximating statistical model. Here, we consider a simplified version of the BEA simulator to illustrate the utility of such an approach. Specifically, we assume the epidemic acts on a closed population, without stratification by age, and that the underlying epidemic parameters do not change over time.

3 Methods

3.1 Gaussian process emulation

Consider a deterministic simulator that takes as input a vector $\boldsymbol{x} = (x_1, \ldots, x_d)'$ with $d \geq 1$ components and produces a scalar output $y = f(\boldsymbol{x})$. The emulator treats the functional form of the simulator output, $f(\cdot)$, as a 'black box' and uses a small number of simulator runs to approximate it (e.g., Sacks et al., 1989; Santner et al., 2003). The standard statistical model for emulation is the GP, which assumes the joint distribution of simulator outputs $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)$ at any finite set of input points (vectors) $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})'$, to be multivariate Normal. From a Bayesian perspective, the GP is the prior distribution for the unknown functional form of $f(\cdot)$ (e.g., Kennedy and O'Hagan, 2001).

A GP is completely determined by its mean function $\mathbb{E}\big(f(\boldsymbol{x})\big)$ and positive semi-definite covariance function $\mathbb{C}\mathrm{ov}\big(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)\big)$, $i = 1, \ldots, n$, and $j = 1, \ldots, n$. Typically, the mean is of the form $\mathbb{E}\big(f(\boldsymbol{x})\big) = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\lambda}$, where $\boldsymbol{h}(\boldsymbol{x})$ is a vector of regressors, and $\boldsymbol{\lambda}$ is the vector of corresponding regression coefficients. For example, letting $\boldsymbol{h}(\boldsymbol{x}) = (1, \boldsymbol{x}')'$ would result in a linear regression scenario with $\boldsymbol{\lambda}$ having $d+1$ components (an intercept and d slopes). The covariance is specified as $\mathbb{C}\mathrm{ov}\big(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)\big) = v\, c(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where v is the GP variance, and

$c(\cdot,\cdot)$ is the correlation function, typically assigned the stationary separable Gaussian form

$$c(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left\{-\sum_{\ell=1}^{d}\beta_\ell(x_{i\ell}-x_{j\ell})^2\right\},\ i=1,\ldots,n,\ j=1,\ldots,n, \tag{2}$$

which implies that $f(\cdot)$ is expected to respond smoothly to changes in its inputs (Schlather, 1999). Here, $\boldsymbol{\beta}=(\beta_1,\ldots,\beta_d)'$ is the vector of *correlation range* parameters, where each $\beta_\ell>0$ controls the correlation in the GP response surface along each input dimension. A large value for β_ℓ indicates weak correlation along input dimension ℓ.

In a fully Bayesian analysis, prior distributions are also placed on the parameters of the GP mean and covariance functions, resulting in a hierarchical prior structure. The parameters $\boldsymbol{\beta}$ are typically assigned a prior distribution based on some reasonable assumptions on the smoothness of $f(\cdot)$ with respect to its inputs (e.g., Higdon et al., 2008b). Some authors, however, disregard uncertainty about $\boldsymbol{\beta}$ replacing it with a 'plug-in' estimator, $\hat{\boldsymbol{\beta}}$ (Bayarri et al., 2007a; Oakley and O'Hagan, 2004). The remaining GP parameters are typically assigned conventional *diffuse* prior distributions.

To fit the GP emulator to the simulator, the latter is run at a *design* set of input points $X=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$. This set is typically chosen according to some "space-filling" criterion, notably a maxi-min Latin hypercube design; see Santner et al. (2003) for a review. For computational considerations, the input values are scaled (or 'normalized') to range over $[0,1]^d$ (Sacks et al., 1989; Kennedy and O'Hagan, 2001; Higdon et al., 2008a). Together, X and the corresponding simulator outputs $\boldsymbol{y}=(y_1,\ldots,y_n)'$, where $y_i=f(\boldsymbol{x}_i)$ for $i=1,\ldots,n$, form the 'training' dataset, $D^s=\{X,\boldsymbol{y}\}$.

To obtain posterior inference on $f(\cdot)$, the likelihood of D^s is combined with the hierarchical prior distributions for the GP and its parameters. The emulator is ultimately given by the posterior predictive distribution of $f(\cdot)$ at any input point, conditional on the training data set. Specifically, let $\tilde{\boldsymbol{x}}$ be a generic input point; then, based on Normal theory and given the GP parameters, the posterior distribution of $f(\tilde{\boldsymbol{x}})$ is Normal with mean $m(\tilde{\boldsymbol{x}})=\boldsymbol{h}(\tilde{\boldsymbol{x}})\boldsymbol{\lambda}+\boldsymbol{r}'(\tilde{\boldsymbol{x}})\,\Sigma^{-1}\,(\boldsymbol{y}-H\boldsymbol{\lambda})$, and variance $\sigma^2(\tilde{\boldsymbol{x}})=v\left(1-\boldsymbol{r}'(\tilde{\boldsymbol{x}})\,\Sigma^{-1}\,\boldsymbol{r}(\tilde{\boldsymbol{x}})\right)$, where $H=(\boldsymbol{h}(\boldsymbol{x}_1),\ldots,\boldsymbol{h}(\boldsymbol{x}_n))'$, $\boldsymbol{r}(\tilde{\boldsymbol{x}})=(c(\tilde{\boldsymbol{x}},\boldsymbol{x}_1),\ldots,c(\tilde{\boldsymbol{x}},\boldsymbol{x}_n))'$, and Σ is the $n\times n$ correlation matrix with elements $c(\boldsymbol{x}_i,\boldsymbol{x}_j)$; see (2). Note that the emulator is an interpolator, so that

at the training data points, $m(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)$ and $\sigma^2(\boldsymbol{x}_i) = 0$, $i = 1, \ldots, n$, and uncertainty about $f(\cdot)$ is largest in regions where no training points are nearby.

3.2 Dynamic emulation

Over recent years, a number of emulator extensions have been proposed in the literature to model simulators of time-varying phenomena, where the output for each input point is a time series. Specifically, let $\tilde{\boldsymbol{x}}$ be a generic input point; then the corresponding time-series of outputs is given by $\{f_t(\tilde{\boldsymbol{x}}), t = 1, \ldots, T\}$. Adapting existing emulation theory, Bayarri et al. (2007a) construct a time-dependent emulator by treating time as an additional (discrete) input. In Bayarri et al. (2007b) wavelet representations are used to model the output curves, while in Higdon et al. (2008a) a principal component basis is utilized to reduce the dimension of the problem. Conti et al. (2009) propose a recursive model, which requires fitting an (approximate) GP emulator at each time step. All these approaches rely on a GP covariance structure that is stationary over time. Working with a multivariate GP, Conti and O'Hagan (2010) propose a 'multi-output' strategy, whereby the time series output is treated as a vector-valued output at the corresponding input point. Within this multivariate GP framework, Fricker et al. (2013) work with a nonseparable covariance structure. Relaxing the 'black box' assumption regarding $f_t(\cdot)$, Reichert et al. (2011) combine a GP prior for $f_t(\cdot)$ with a linear state-space structure based on knowledge of the state-space equation system underlying the simulator.

Apart from conceptual and computational limitations, these methods poorly handle dependence through time and dependence across time series corresponding to different input points. These are addressed by the dynamic emulator proposed by Liu and West (2009), which draws upon state-space modeling ideas to account for three types of interdependencies: the first is dependence over the input space, which is modeled using a GP; the second is dependence over time (within each time series), which is handled using a TVAR model; and the third is dependence across time series of different input points, which is achieved by linking the GP to the TVAR model through the covariance function of the GP.

Specifically, denote the multivariate output at time t for input points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ by the vector $\boldsymbol{y}_t = (y_t(\boldsymbol{x}_1), \ldots, y_t(\boldsymbol{x}_n))' = (f_t(\boldsymbol{x}_1), \ldots, f_t(\boldsymbol{x}_n))'$, $t = 1, \ldots, T$. Liu and West (2009)

model $\boldsymbol{y}_t$ using the following multivariate TVAR structure:

$$\boldsymbol{y}_t = F_t'\boldsymbol{\phi}_t + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim N\left(\mathbf{0}, v_t\Sigma\right), \tag{3}$$

$$F_t' = \begin{pmatrix} y_{t-1}\left(\boldsymbol{x}_1\right) & \cdots & y_{t-p}\left(\boldsymbol{x}_1\right) \\ \vdots & \ddots & \vdots \\ y_{t-1}\left(\boldsymbol{x}_n\right) & \cdots & y_{t-p}\left(\boldsymbol{x}_n\right) \end{pmatrix}, \tag{4}$$

$$\boldsymbol{\phi}_t = \boldsymbol{\phi}_{t-1} + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim N(\mathbf{0}, W_t). \tag{5}$$

Here, p is the specified time-lag, implying that the correlation is negligible after p time steps, $\boldsymbol{\phi}_t = (\phi_{t,1}, \ldots, \phi_{t,p})'$ are the TVAR coefficients, which are assumed to follow a random walk over time, and $\boldsymbol{\epsilon}_t = \left(\epsilon_t(\boldsymbol{x}_1), \ldots, \epsilon_t(\boldsymbol{x}_n)\right)'$. The errors $\epsilon_t(\cdot)$ are in turn modeled independently over time using a GP prior distribution with mean zero and covariance function $v_t c(\cdot,\cdot)$, where v_t is the dynamic variance and $c(\cdot,\cdot)$ is the correlation function defined in (2), so that $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the $(i,j)^{\text{th}}$ element of the correlation matrix Σ.

Note that since $\boldsymbol{\phi}_t$, v_t, and W_t all vary over time, the model is able to capture temporal non-stationarity in the dynamic simulator output. To complete the model defined in (3)-(5), prior distributions are specified for $\boldsymbol{\phi}_\mathbf{0}$, $\{v_t, t = 1, \ldots, T\}$, $\{W_t, t = 1, \ldots, T\}$, and $\boldsymbol{\beta}$. Then, posterior inference is obtained using MCMC by utilizing the forward filtering backward sampling (FFBS) algorithm (West and Harrison, 1997) to sample the full sequence of TVAR coefficients and variances, and a Metropolis-within-Gibbs step to sample the posterior distribution of $\boldsymbol{\beta}$.

However, the emulator model setup in (3) only partially succeeds at capturing the types of interdependencies intrinsic in the BEA simulator. In its covariance structure, the model uses v_t to capture dependence through time and Σ to capture correlation across the input space, linking together time series corresponding to different input points. In its mean structure, the model uses the TVAR component to capture dependence on time lags. However, this structure does not take into account possible trends associated with the inputs, which, in the case of epidemic models, often have a dramatic effect on the output. We propose to model these trends using an input-dependent dynamic regression component,

expanding (3) into $\boldsymbol{y}_t = F_t'\boldsymbol{\phi}_t + H\boldsymbol{\lambda}_t + \boldsymbol{\epsilon}_t$, where $\boldsymbol{\lambda}_t$ are the time-varying regression coefficients associated with the regression functions $\boldsymbol{h}(\boldsymbol{x}_i)$, $i = 1, \ldots, n$, which are collected into the matrix H introduced in Section 3.1. The regression coefficients are allowed to vary over time because we expect the impact of the epidemic parameters (which are the inputs of the BEA simulator) to differ between the earlier and later stages of the epidemic. Letting $\boldsymbol{\phi}_t$ and $\boldsymbol{\lambda}_t$ follow two independent random walks over time, our expanded emulation model is given by

$$\boldsymbol{y}_t = \begin{pmatrix} F_t' & H \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi}_t \\ \boldsymbol{\lambda}_t \end{pmatrix} + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim N(\boldsymbol{0}, v_t\Sigma), \tag{6}$$

$$\tag{7}$$

$$\begin{pmatrix} \boldsymbol{\phi}_t \\ \boldsymbol{\lambda}_t \end{pmatrix} = \begin{pmatrix} \boldsymbol{\phi}_{t-1} \\ \boldsymbol{\lambda}_{t-1} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\omega}_{t1} \\ \boldsymbol{\omega}_{2t} \end{pmatrix}, \qquad \begin{pmatrix} \boldsymbol{\omega}_{t1} \\ \boldsymbol{\omega}_{2t} \end{pmatrix} \sim N(\boldsymbol{0}, W_t), \tag{8}$$

where W_t now indicates the covariance matrix of the Normal errors $(\boldsymbol{\omega}_{1t}', \boldsymbol{\omega}_{2t}')'$ for $\boldsymbol{\phi}_t$ and $\boldsymbol{\lambda}_t$, respectively. For $\boldsymbol{h}(\boldsymbol{x}) = (1, \boldsymbol{x}')'$, we obtain a dynamic linear regression component with a time-varying intercept that partially accounts for the possible effects of omitted variables. Let $\boldsymbol{y}_{1:T} = (\boldsymbol{y}_1', \ldots, \boldsymbol{y}_T')'$, $\boldsymbol{\phi}_{1:T} = (\boldsymbol{\phi}_1', \ldots, \boldsymbol{\phi}_T')'$, $\boldsymbol{\lambda}_{1:T} = (\boldsymbol{\lambda}_1', \ldots, \boldsymbol{\lambda}_T')'$, and $\boldsymbol{v}_{1:T} = (v_1, \ldots, v_T)'$; then, the likelihood of the time series outputs is given by

$$p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\phi}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{v}_{1:T}, \boldsymbol{\beta}) \propto |\, \Sigma \,|^{-(T-p)/2} \exp(-\Omega/2) \prod_{t=p+1}^{T} v_t^{-n/2}, \tag{9}$$

where

$$\Omega = \sum_{t=p+1}^{T} (\boldsymbol{y}_t - F_t'\boldsymbol{\phi}_t - H\boldsymbol{\lambda}_t)'\Sigma^{-1}(\boldsymbol{y}_t - F_t'\boldsymbol{\phi}_t - H\boldsymbol{\lambda}_t)/v_t. \tag{10}$$

To complete the model in (6) and (8), we need to specify prior distributions for the initial states $\{\boldsymbol{\phi}_0, \boldsymbol{\lambda}_0\}$, the time-varying variances, and the GP correlation range parameters. Finally, just as efficiently as in Liu and West (2009), posterior inference for all model parameters may be obtained using the FFBS algorithm for the states and variances, and a

Metropolis-Hastings-within-Gibbs step for $\boldsymbol{\beta}$; see Appendix A for technical details.

The dynamic emulator is the posterior predictive distribution at any input point $\tilde{\boldsymbol{x}}$. This distribution is obtained by integrating the conditional distribution of its output $y_t(\tilde{\boldsymbol{x}})$, typically using a Monte Carlo approach, over the posterior distribution of the emulation model parameters. Specifically, the distribution of $y_t(\tilde{\boldsymbol{x}})$, conditional on all model parameters, the output training data $\boldsymbol{y}_{1:T}$, and the previous time series values $\boldsymbol{y}_{t-1:t-p}(\tilde{\boldsymbol{x}})$, is found to be Normal with mean $m_t(\tilde{\boldsymbol{x}})$ and variance $\sigma_t^2(\tilde{\boldsymbol{x}})$ given by

$$m_t(\tilde{\boldsymbol{x}}) = \sum_{j=1}^{p} \phi_{t,j}\, y_{t-j}(\tilde{\boldsymbol{x}}) + \boldsymbol{h}(\tilde{\boldsymbol{x}})\boldsymbol{\lambda}_t + v_t^{-1}\boldsymbol{r}'(\tilde{\boldsymbol{x}})\,\Sigma^{-1}\,\boldsymbol{\epsilon}_t, \tag{11}$$

$$\sigma_t^2(\tilde{\boldsymbol{x}}) = v_t\left(1 - \boldsymbol{r}'(\tilde{\boldsymbol{x}})\,\Sigma^{-1}\,\boldsymbol{r}(\tilde{\boldsymbol{x}})\right), \tag{12}$$

where $\boldsymbol{\epsilon}_t = \big(\epsilon_t(\boldsymbol{x}_1), \ldots, \epsilon_t(\boldsymbol{x}_n)\big)'$, with $\epsilon_t(\boldsymbol{x}_i) = y_t(\boldsymbol{x}_i) - \sum_{j=1}^{p} \phi_{t,j}\, y_{t-j}(\boldsymbol{x}_i) - \boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\lambda}_t$, $i = 1, \ldots, n$. As in Section 3.1, the dynamic emulator is also an interpolator, where, at the training data points, we obtain $m_t(\boldsymbol{x}_i) = y_t(\boldsymbol{x}_i)$ and $\sigma^2(\boldsymbol{x}_i) = 0$, $i = 1, \ldots, n$.

3.3 A strategy for calibration

The purpose of calibration is to estimate the parameters underlying the modeled physical process, using field observations of the process and simulator runs (Kennedy and O'Hagan, 2001). In our case, the aim is to estimate the epidemic parameters. The standard statistical formulation of the calibration problem starts by assuming that the field observations are a noisy representation of the real-world process of interest, i.e., 'field = reality + error'. Then, to link these field observations to the simulator data, the simulator output is modeled as 'reality + bias', where the bias refers to the *discrepancy* between the simulator and the process it is modeling. Thus, a unifying model for both sources of data can be formulated as 'field = simulator + bias + error' (Kennedy and O'Hagan, 2001). In some cases, the bias term can be ignored; in others, it plays a crucial role (Higdon et al., 2008a). Estimating the bias of the BEA simulator is not the purpose of this paper. Here, we follow the modeling framework in Birrell et al. (2011), where the observed epidemic data are modeled as 'simulator + error', thus assuming the simulator is unbiased. However, we should note that

if the BEA simulator were to be substantially biased, the form of the bias would affect the posterior distribution of the calibration parameters. For a discussion on the inclusion of a bias term, see Kennedy and O'Hagan (2001); Higdon et al. (2004); Loeppky et al. (2006).

The proposed calibration approach thus relies on the availability of two distinct data streams: (i) a time series $\boldsymbol{z}_{1:T}$ of (suitably rescaled and log-transformed) field observations; and (ii) a training set of selected simulator runs $D^s = \{X, \boldsymbol{y}_{1:T}\}$ consisting of simulator inputs $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$ and corresponding time series outputs $\boldsymbol{y}_{1:T} = (\boldsymbol{y}_1', \ldots, \boldsymbol{y}_T')'$, where $\boldsymbol{y}_t = \big(f_t(\boldsymbol{x}_1), \ldots, f_t(\boldsymbol{x}_n)\big)'$ for $t = 1, \ldots, T$. Then, assuming the observation errors of the field data are *i.i.d.* over time and independent of the simulator data, z_t and $\boldsymbol{y}_t$ can be modeled hierarchically at each time step t as

$$\text{observation model:} \qquad z_t = y_t(\boldsymbol{\eta}) + \epsilon_{z_t},\ \epsilon_{z_t} \overset{iid}{\sim} N(0, \sigma_z^2) \tag{13}$$

$$\text{simulator model:} \qquad y_t(\boldsymbol{\eta}) = M_t(\boldsymbol{\eta}) + \epsilon_t(\boldsymbol{\eta})\,,\ \epsilon_t(\cdot) \sim GP\big(0, v_t\, c(\cdot,\cdot)\big) \tag{14}$$

$$\text{dynamic coefficients:} \qquad \begin{pmatrix} \boldsymbol{\phi}_t \\ \boldsymbol{\lambda}_t \end{pmatrix} = \begin{pmatrix} \boldsymbol{\phi}_{t-1} \\ \boldsymbol{\lambda}_{t-1} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\omega}_{t1} \\ \boldsymbol{\omega}_{2t} \end{pmatrix}, \qquad \begin{pmatrix} \boldsymbol{\omega}_{t1} \\ \boldsymbol{\omega}_{2t} \end{pmatrix} \sim N(\mathbf{0},\, W_t), \tag{15}$$

where $y_t(\boldsymbol{\eta})$ is the output at time t from the BEA computer model when evaluated at the underlying epidemic parameters $\boldsymbol{\eta}$, σ_z^2 is the constant variance of the observation errors, $M_t(\cdot) = \sum_{j=1}^{p} \phi_{t,j} y_{t-j}(\cdot) + \boldsymbol{h}(\cdot)\boldsymbol{\lambda}_t$, and $c(\cdot,\cdot)$ is the Gaussian correlation function specified in (2). Note that the observation model (13) reflects the previously discussed assumption that the simulator is an unbiased proxy to the data-generating physical system. Now, denote with $\boldsymbol{\xi}_t = \{\boldsymbol{\phi}_t, \boldsymbol{\lambda}_t, v_t, W_t, \boldsymbol{\beta}\}$ the collection of parameters specifying the GP in (14) and (15), and let $\boldsymbol{\xi}_{1:T} = (\boldsymbol{\xi}_1', \ldots, \boldsymbol{\xi}_T')'$. Then, from (13)–(15), it follows that the joint distribution $p(z_t, \boldsymbol{y}_t \mid \boldsymbol{\eta}, \sigma_z^2, \boldsymbol{\xi}_t, X)$ of the field and simulator data at time t is given by the following $(n+1)$-variate Normal distribution

$$N\left(\begin{pmatrix} M_t(\boldsymbol{\eta}) \\ \boldsymbol{M}_t(X) \end{pmatrix},\ \begin{pmatrix} \sigma_z^2 + v_t & v_t \boldsymbol{r}'(\boldsymbol{\eta}) \\ v_t \boldsymbol{r}(\boldsymbol{\eta}) & v_t \Sigma \end{pmatrix}\right), \tag{16}$$

where $\boldsymbol{M}_t(X) = \big(M_t(\boldsymbol{x}_1), \ldots, M_t(\boldsymbol{x}_n)\big)'$ and $\boldsymbol{r}(\boldsymbol{\eta}) = \big(c(\boldsymbol{\eta}, \boldsymbol{x}_1), \ldots, c(\boldsymbol{\eta}, \boldsymbol{x}_n)\big)'$.

Finally, given the likelihood specified by (16) and prior distributions over the parameters in (13)–(15), the joint posterior distribution of all the emulation-calibration model parameters is obtained as

$$p(\boldsymbol{\eta}, \sigma_z^2, \boldsymbol{\xi}_{1:T} \mid D^s, \boldsymbol{z}_{1:T}) \propto \\ p(\boldsymbol{\eta})p(\sigma_z^2)p(\boldsymbol{\xi}_{1:T}) \prod_{t=p+1}^{T} p\big((z_t, \boldsymbol{y}_t) \mid \boldsymbol{\eta}, \sigma_z^2, \boldsymbol{\xi}_t, \boldsymbol{z}_{t-1:t-p}, \boldsymbol{y}_{t-1:t-p}\big), \tag{17}$$

where $\boldsymbol{z}_{(t-1):(t-p)} = (z_{t-1}, \ldots, z_{t-p})' \subset \boldsymbol{z}_{1:T}$. The prior distribution for $\boldsymbol{\xi}_{1:T}$ is discussed in Section 3.2 and Appendix A, and the prior distributions for $\boldsymbol{\eta}$ and σ_z^2 are discussed in Section 4.2.

While the emulation model in Section 3.2 allows for efficient posterior sampling of $\boldsymbol{\xi}_{1:T}$ through the FFBS algorithm, an MCMC scheme to sample the posterior distribution in (17) does not. In particular, the posterior distributions of the emulator model variance components $(\boldsymbol{v}_{1:T}, \boldsymbol{W}_{1:T})$ will have to be sampled using Metropolis-Hastings steps, an extremely computationally challenging scenario. Hence, we adopt a pragmatic approach that breaks up the analysis into two stages, where in the first stage, the emulator is *trained* using only the simulator data, utilizing the FFBS algorithm to sample the posterior distribution of $\boldsymbol{\xi}_{1:T}$. Then, in the second stage, the posterior distribution of $(\boldsymbol{\eta}, \sigma_z^2)$ is obtained using the field data and conditional on the simulator data and the trained emulation model from the first stage. 'Modularization', as this two-stage approach is referred to in the literature (Bayarri et al., 2007b; Liu et al., 2009; Rougier, 2008), offers an attractive solution in terms of: efficiency, since the MCMC is split into two lower dimensional blocks; computational tractability, since the MCMC in the emulation block relies on sampling from standard distributions, so that the FFBS algorithm could be utilized; and improved MCMC diagnostics.

Ultimately, the posterior distribution of interest in calibration is the marginal posterior distribution given by

$$p(\boldsymbol{\eta}, \sigma_z^2 \mid D^s, \boldsymbol{z}_{1:T}) = \int p(\boldsymbol{\eta}, \sigma_z^2 \mid \boldsymbol{\xi}_{1:T}, D^s, \boldsymbol{z}_{1:T}) p(\boldsymbol{\xi}_{1:T} \mid D^s, \boldsymbol{z}_{1:T}) d\boldsymbol{\xi}_{1:T} \tag{18}$$

Modularization is equivalent to replacing $p(\boldsymbol{\xi}_{1:T} \mid \boldsymbol{z}_{1:T}, D^s)$ in (18) with $p(\boldsymbol{\xi}_{1:T} \mid D^s)$, thus

allowing for the posterior of $\boldsymbol{\xi}_{1:T}$ to be obtained from the emulation model alone. For our purposes, the two-stage implementation involves first obtaining $p(\boldsymbol{\xi}_{1:T} \mid D^s)$ according to the methods discussed in Section 3.2. In the second stage, the (modularized) posterior distribution of $(\boldsymbol{\eta}, \sigma_z^2)$ is sampled by repeatedly drawing from the MCMC sample obtained in Stage 1, followed by a draw from the distribution of $(\boldsymbol{\eta}, \sigma_z^2)$ given $\boldsymbol{z}_{1:T}$ and D^s; see Appendix B for implementation details. A simplified (and more computationally efficient) two-stage approach can be implemented using plug-in estimators for some (Bayarri et al., 2007a) or all (Henderson et al., 2009) components of the emulation model parameters.

4 Results

4.1 Training the emulator

To generate simulator runs to train the emulator, feasible ranges are determined for the BEA simulator inputs (Table 1), and, using a maxi-min Latin hypercube design over the space determined by the input ranges, a sample of $n = 200$ input points, $\boldsymbol{x}_i, i = 1, \ldots, 200$, is drawn. The BEA simulator is then run for each $\boldsymbol{x}_i$ to obtain the time series outputs $\boldsymbol{\mu}_{1:T}(\boldsymbol{x}_i)$, where $T = 250$. The length of the time series is chosen to match the life span of the epidemic scenario considered in Birrell et al. (2011). This set of simulator data is used to train the DLM-GP emulator according the methods presented in Section 3.2. To assess the predictive performance of the emulator, we generate a different set of 200 input points (chosen using a new Latin hyper cube design), for which we also obtain the BEA simulator time series outputs. We use the new set of 200 inputs and corresponding 200 time series outputs as a "validation" dataset and denote it by $D^v = \{(\boldsymbol{x}_j^v, \mu_t(\boldsymbol{x}_j^v)), j = 1, \ldots, 200, t = 1 \ldots, 250\}$.

Here, we present results based on a DLM-GP model with a regression component having $\boldsymbol{h}(\boldsymbol{x}) = (1, \boldsymbol{x}')'$. In deciding on the degree (number of lags) of the TVAR component of the DLM, we investigate two models with TVAR(1) and TVAR(2) components, each sharing the same regression component $\boldsymbol{h}(\cdot)$. Following Liu and West (2009), we compare the predictive performance of the two resulting emulators over the validation dataset, D^v, using the mean

square error (MSE) defined as

$$MSE(\boldsymbol{x}_j^v) = \frac{1}{T}\sum_{t=1}^{T}\left(\hat{\mu}_t(\boldsymbol{x}_j^v) - \mu_t(\boldsymbol{x}_j^v)\right)^2, \quad j = 1, \ldots, 200, \tag{19}$$

where $\hat{\mu}_t(\cdot)$ is the emulator's median at time t. Recall that the emulator is the posterior predictive distribution of $f_t(\cdot)$, which is conditionally Normal with parameters given in (11) and (12). Validated over D^v, the TVAR(1) emulator yields MSEs with mean and standard deviation 1.03 and 1.50, respectively, while the TVAR(2) emulator yields MSEs with mean and standard deviation 1.04 and 1.45, respectively. Thus, the two TVAR models are well-matched in their predictive performance, motivating the use of the simpler, TVAR(1), model.

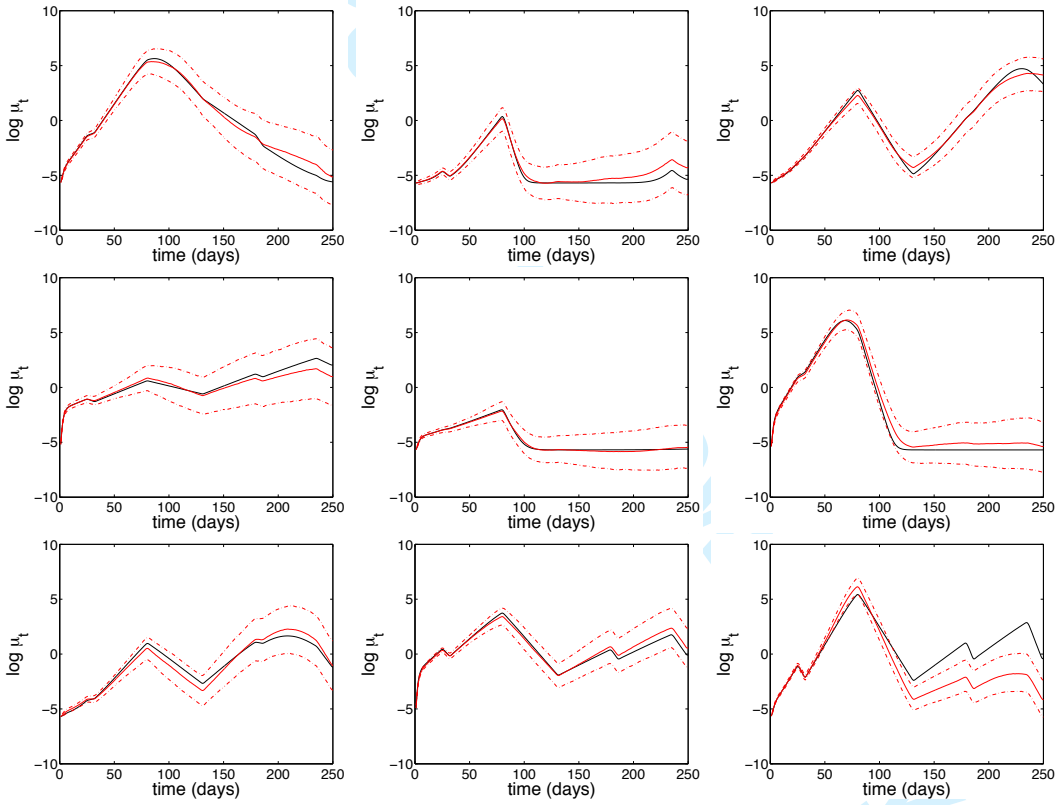

Figure 3: Emulator medians and 95% credible intervals (in red) for the validation simulator runs (in black) based on a DLM-GP emulator with TVAR(1) and $\boldsymbol{h}(\boldsymbol{x}) = (1, \boldsymbol{x}')'$ components.

Figure 3 provides a graphical illustration of the predictive performance of the emulator. Here, posterior predictive medians and 95% credible intervals are plotted over time for 9

different validation points corresponding to distinct epidemic trajectories selected to exhibit one ore more peaks over the lifetime of the epidemic. In general, the emulator captures the shape of the simulator's time series output reasonably well, with posterior uncertainty increasing with time, as expected since the variances increase with time in forward filtering (see Figure 2-B). At each time step, a handful of validation points had standardized residuals greater than 2.5, an extreme example of which is seen in the bottom rightmost plot in Figure 3. Here, the large MSE is likely due to the input point for this time series being in a region (i) not adequately covered by the original training Latin Hypercube design or (ii) where the simulator output is too volatile to be captured by our emulator. In this example, emulator is trained using 200 simulator runs. A greater number of simulator training runs would result in a more accurate emulator (O'Hagan, 2006). However, the improved emulator performance would be at the expense of obtaining more computationally expensive simulator runs.

In fitting the DLM-GP emulator, we worked with a number of different specifications for the hyper-parameters defining the initial priors of the DLM. Our prior sensitivity analysis showed posterior inference to be virtually unchanged (see Appendix A for a discussion of the prior specifications used). In the fully Bayesian emulation model, we set the prior for the GP correlation parameters to $p(\boldsymbol{\beta}) \propto \prod_{\ell=1}^{d} \left(1 - \exp(-\beta_\ell/4)\right)^{0.9} \exp\left(-\beta_\ell/4\right)$. This prior, also used by Liu and West (2009), reflects a prior belief that only a subset of the inputs are influential in their impact on the simulator output (Higdon et al., 2008b).

Figure 4 shows plots of the posterior distributions of $\boldsymbol{\beta}$. Here, the posteriors of β_2 and β_5 have considerably greater values in their support than other components of $\boldsymbol{\beta}$. This indicates that the output response surface is more sensitive to small (local) changes in inputs x_2 and x_5 than other inputs. This is not surprising, since these two inputs correspond to parameters for the exponential growth rate of the epidemic and the reduction in the population contact rate due to the school Summer holiday period, two parameters dramatically controling the behavior of the epidemic (see Table 1). On the other hand, the smallest correlation range appears to be β_3, which is associated with the input representing the average infectious period. This means the simulator's response surface is not very sensitive to changes in the values of this input. Thus, for the purpose of obtaining outputs that reasonably capture patterns along the response surface, a more efficient design over the inputs should populate

the design matrix, X, more densely along the 2^{nd} and 5^{th} input dimensions, and perhaps to a lesser extent along the 3^{rd} dimension.

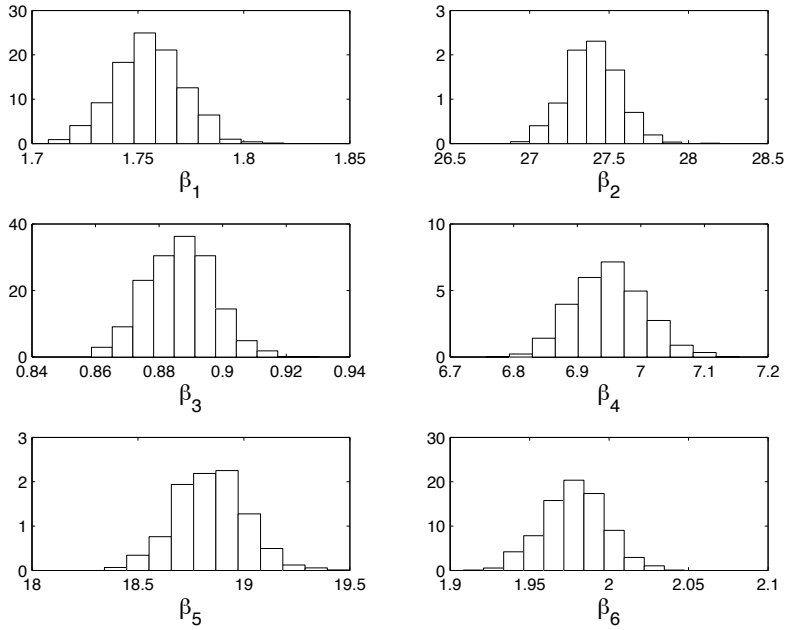

Figure 4: Posterior inference for the correlation range parameters $\beta_1, \ldots, \beta_6$ using a DLM-GP emulator with TVAR(1) and $\boldsymbol{h}(\boldsymbol{x}) = (1, \boldsymbol{x}')'$ components.

We further investigated using plug-in estimators, $\hat{\boldsymbol{\beta}}$, for the GP correlation parameters $\boldsymbol{\beta}$. Specifically, we set $\hat{\boldsymbol{\beta}}$ to be the posterior mode of $\boldsymbol{\beta}$ from the fully Bayesian emulator model (although other plug-in estimators can be used, e.g., Bayarri et al. (2007a)). The semi-Bayesian emulator (based on $\hat{\boldsymbol{\beta}}$) resulted in a MSE distribution with mean 1.02 and standard deviation 1.48. Thus, while not accounting for uncertainty in $\boldsymbol{\beta}$, we find the predictive performance of the semi-Bayesian emulator to be nearly the same as that of the fully Bayesian one, a conclusion also reached by other authors using this type of plug-in approach (e.g., Bayarri et al., 2007a). Replacing $\boldsymbol{\beta}$ with $\hat{\boldsymbol{\beta}}$ in the emulation model results in a MCMC algorithm entirely made up of Gibbs steps, and, thus, in a much faster posterior sampling scheme. Subsequent reported results are obtained under the semi-Bayesian emulation model.

Figure 5-A shows summary posterior trajectories (median and 95% credible intervals) for the TVAR coefficient, ϕ_{t1} over 250 days. The posterior for ϕ_{t1} initially has an average around 0.9 smoothly increasing to 1 within the first 50 days, which implies that the simulator

output changes smoothly both locally and globally over time. Figure 5-B shows summary posterior trajectories for the emulator variance v_t. These trajectories show global non-stationarity with periods of apparent instability corresponding to intervals of rapid changes in the simulator output near days 30, 83, and 180, which coincide with the beginning of school holidays. We note that, under the fully Bayesian emulation model, we found the posterior inference for ϕ_{t1} and v_t to be virtually undistinguishable from that obtained under the semi-Bayesian approach.

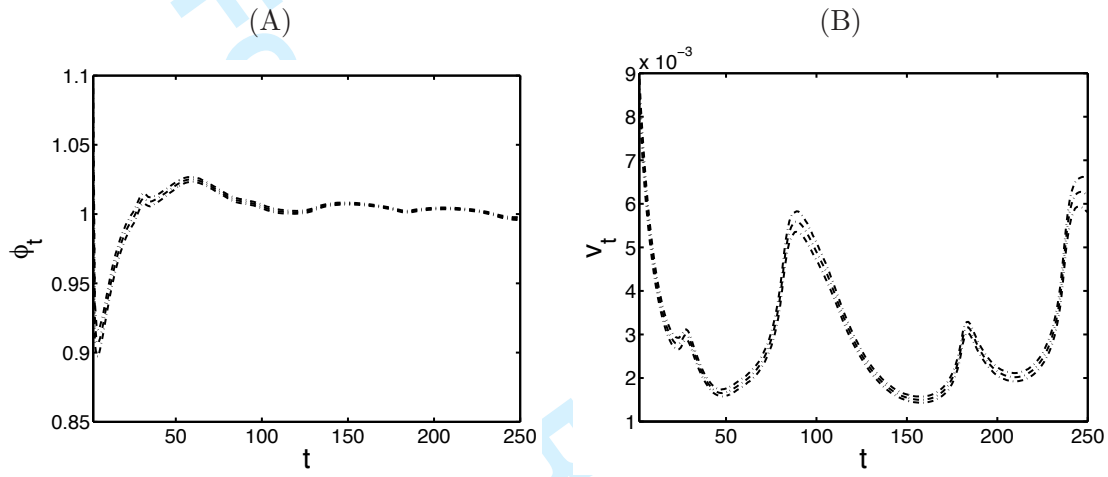

Figure 5: Pointwise posterior medians and 95% credible intervals for the trajectory of the TVAR coefficient ϕ_{t1} (A) and the observational error v_t (B) over 250 days.

Summary posterior trajectories for the dynamic regression coefficients, $\lambda_{t0}, \ldots, \lambda_{t6}$ are shown in Figure 6. Each of these parameters was given a Normal initial prior distribution with mean zero and variance 5. It is evident that there is substantial 'global' non-stationarity over time, with some of the slopes changing from positive to negative or vice versa. Note that the posterior trajectories for λ_{t3} cover zero for almost all time points, and that posterior trajectories for λ_{t1} are zero or near zero for most of their path. However, the emulator is able to pick out a clear effect of school holidays. Specifically, the posterior trajectories of λ_{t5} capture the effect of a changing population contact rate near day 83, which is the start of the Summer holiday; and the posterior trajectories of λ_{t6} capture changes in the population contact rate near days corresponding to other school holidays, in particular days 30 and 180. Furthermore, λ_{t4} appears to converge to zero after about 150 days, which is consistent with the fact that x_4 is an initial condition (related to the initial number of infective individuals

in the population), so its effect is expected to diminish with time.

A sample of size $100,000$ was drawn from the joint posterior distribution of the DLM-GP emulator parameters, to be employed in the calibration stage.

4.2 Calibration

Here, we are interested in inference for all the elements of $\boldsymbol{\eta}$ using the field observations and conditional on the emulation stage. To investigate the performance of our proposed calibration approach, we use synthetic epidemic data generated from a Poisson distribution centered on the BEA simulator, given the 'true' normalized parameter values $\boldsymbol{\eta} = \boldsymbol{\eta}_0 = (0.73, 0.39, 0.51, 0.84, 0.33, 0.67)'$. This synthetic epidemic data are plotted in Figure 7. As discussed in Section 3.3, working on the log scale, we model the field data according to (13) at each time step using a Normal distribution with mean equal to the BEA simulator and variance σ_z^2. We specify proper Uniform priors for $\boldsymbol{\eta}$ over their normalized ranges, shown on the original scale in Table 1. Also, we specify an Inverse-Gamma$(2.05, 1.05)$ prior for σ_z^2, a relatively diffuse specification. Then, the modular posterior distribution for $(\boldsymbol{\eta}, \sigma_z^2)$ is obtained by combining the priors for $\boldsymbol{\eta}$ and σ_z^2 with the likelihood for the field data, conditional on the first stage (emulation) analysis and the simulator data (see Section 3.3).

Figure 8 shows the posterior distributions for the components of $\boldsymbol{\eta}$ plotted against their priors. Posterior learning is strongest for η_2, η_4, and η_5, which represent the exponential growth rate of the epidemic, a function of the size of the initially infective population, and the reduction in population contact rate induced by the Summer school holiday, respectively. The posterior medians (95% credible intervals) for these parameters are approximately 0.50 (0.22, 0.87), 0.69 (0.49, 0.87), and 0.37 (0.15, 0.70), respectively. Additionally, some prior-to-posterior learning is evident is the posterior histogram of η_6, which represents the reduction in the population contact rate induced by all school holidays other than the Summer holiday. This parameter is not expected to have as big an impact on the evolution of the epidemic as η_5 since it corresponds to shorter duration holidays. The posterior median (95% credible interval) for η_6 is 0.70 (0.21, 0.97).

With regard to posterior inference for η_1 and η_3, we find there is essentially no prior-to-posterior learning. These two inputs represent the proportion of infected individuals

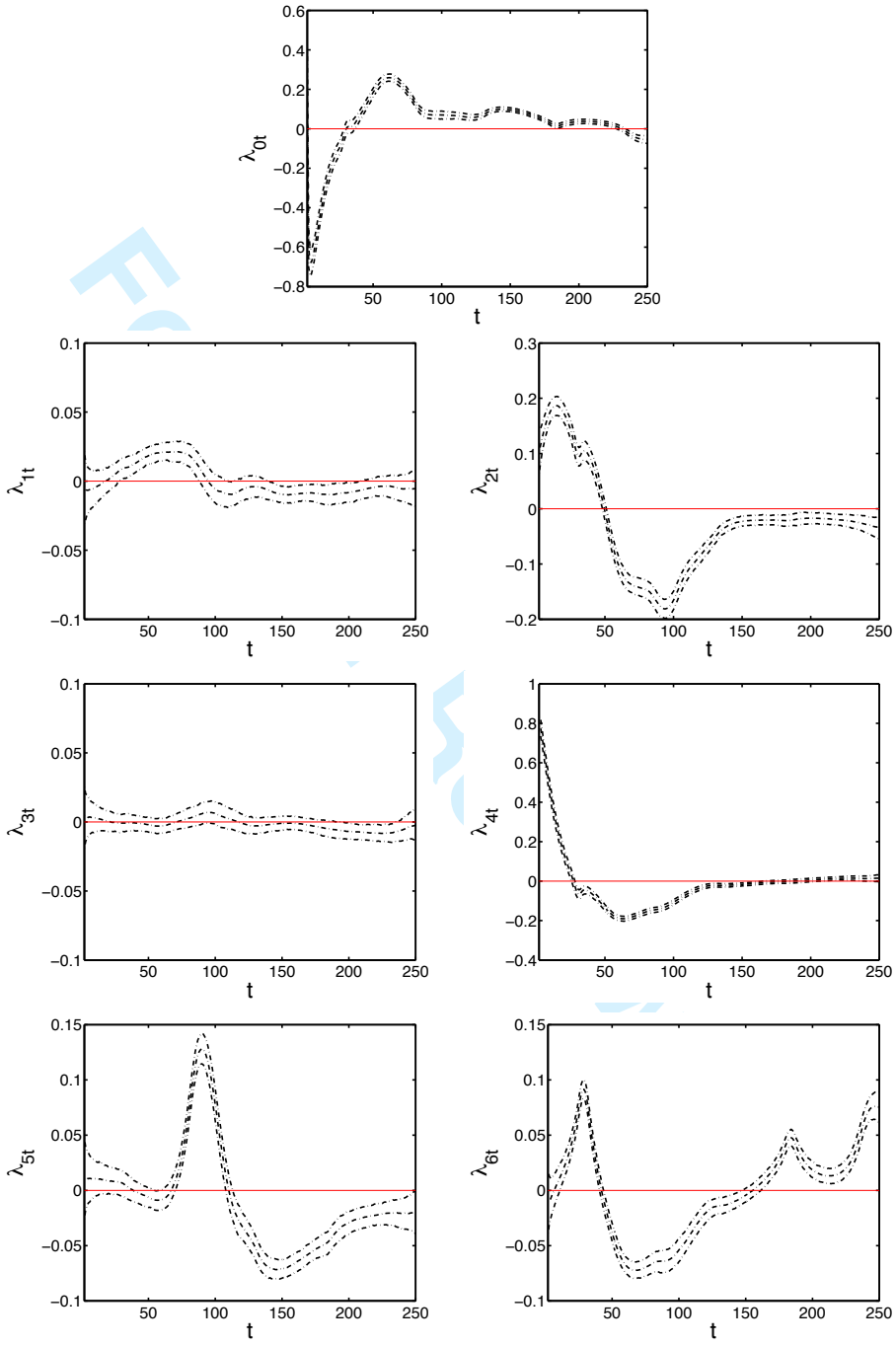

Figure 6: Pointwise posterior medians and 95% credible intervals for the trajectory of the regression component coefficients $\lambda_{t0}, \ldots, \lambda_{t6}$. The zero line is in red.

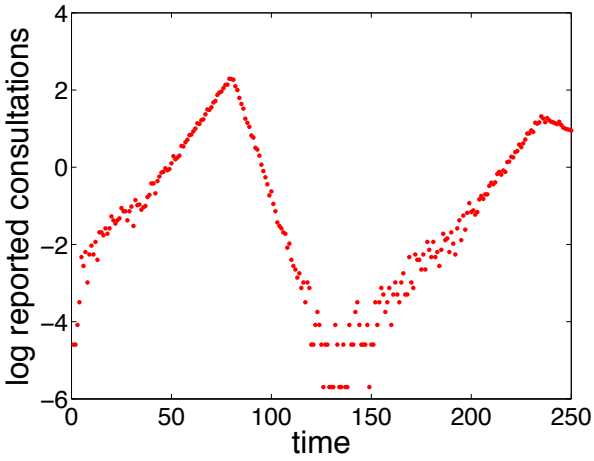

Figure 7: Synthetic observed epidemic data on the log scale.

who consult their doctors and the average infectious period, respectively. As seen in the emulation stage, the BEA simulator response surface is not sensitive to changes in these two inputs, and the regression coefficients corresponding to their input values were either zero or near zero over the lifetime of the epidemic. Thus, the BEA simulator surface, which we model as the the mean of the field data, is relatively flat and 'insensitive' with respect to these two inputs, so it is not surprising that the calibration procedure is not able to 'learn' their underlying values from field observations. In fact, the values specified for η_3, the average infectious period, in the training set correspond to a small range of approximately $(3.44, 3.46)$ days, which is a very informative range already. With regard to η_1, this is actually the product of two quantities respectively representing the proportion of symptomatic individuals and the propensity to consult a doctor. Here, perhaps a further investigation into the ranges of these two quantities is needed in order to determine a more informative range for η_1. Finally, posterior inference for the observational error variance, σ_z^2, is summarized in Figure 9, which shows significant prior-to-posterior learning.

5 Discussion

We have developed a novel framework for the calibration of a dynamic epidemic simulator for A/H1N1 influenza. Our approach involves emulating an epidemic simulator, which is both analytically intractable and computationally expensive. The emulator, which is

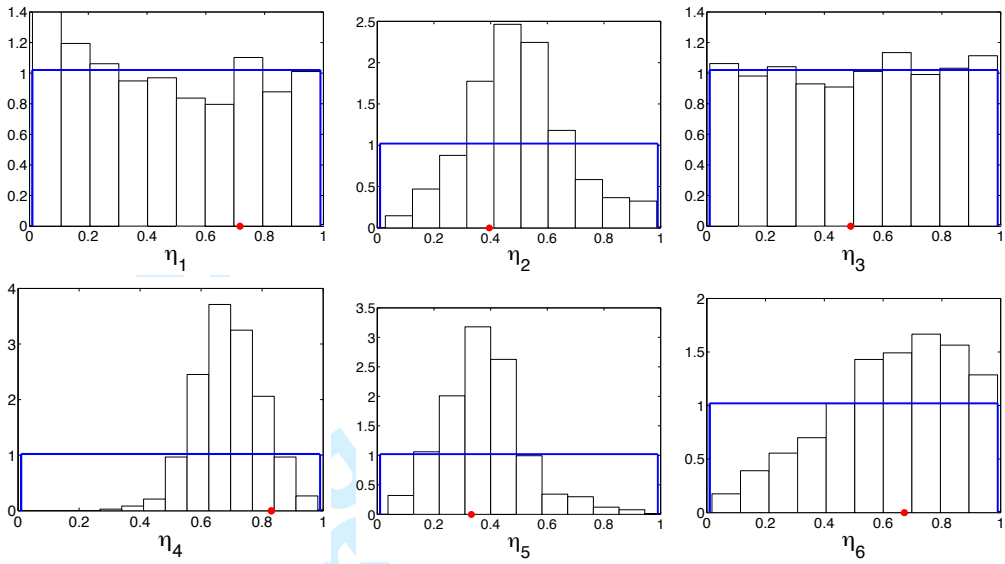

Figure 8: Posterior distributions of the parameters $\eta_1, \ldots, \eta_6$. Uniform prior distributions are in blue, and true values are in red.

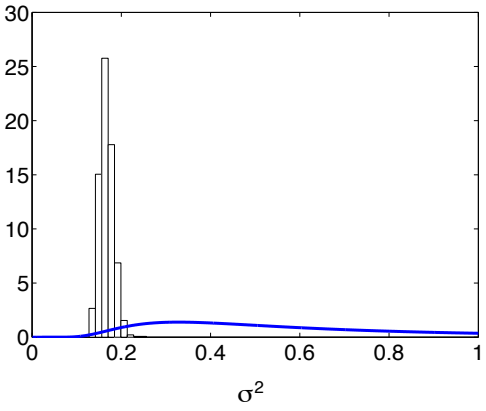

Figure 9: Posterior distribution for σ_z^2. The prior distribution, an Inverse-Gamma, is in blue.

embedded in a two-stage calibration procedure, is based on a coupled DLM-GP prior model for the time series simulator outputs. Specifically, the DLM component included a dynamic regression term to capture time-varying effects of the simulator inputs. The GP component modeled the correlation structure over the input space and across time series of different input points. Having all the desirable properties of the multivariate Normal distribution, the GP results in tractable computations. We have shown that our approach works well for estimating parameters that are important in determining the shape of the dynamic simulator's output.

We believe that this emulation-calibration approach is likely to work well with other dynamic simulators that have a small to moderate number of inputs. For simulators with a large number of inputs, our approach requires estimating a correspondingly large number of correlation range parameters, which is challenging under a fully Bayesian approach since those parameters are obtained using Metropolis-Hastings steps that require careful tuning of the proposal distributions. However, it is often the case that the simulator output is only affected by a subset of the inputs. Here, input screening to identify 'active' inputs (Welch et al., 1992; Koehler and Owen, 1996; Linkletter et al., 2006) and probabilistic sensitivity analysis to quantify their impact on the simulator output (Saltelli et al., 2000; Oakley and O'Hagan, 2004; Farah, 2013) may be utilized as tools to reduce the dimension of the input space.

In terms of implementation, the BEA simulator was coded in C++ and implemented using optimized parallel processing over a cluster of Linux machines. The DLM-GP emulator is implemented in Matlab with the code running serially on one MS Windows machine. Under these two very different implementations, calibration using the emulator does not yield significant gains relative to the computer model in terms of run time. A fair comparison of computational run time is to consider the complexity of each algorithm. A single run of the BEA simulator requires that at every time step, t, parameters specifying the SEIR states are obtained by solving a system of differential equations. Then, the number of reported doctor consultations $\mu_t(\boldsymbol{x})$ is obtained as convolution over three indices (incubation duration, symptoms duration, and reporting time). This process is repeated at every time step until one epidemic outcome (i.e. one simulator run) is obtained. On the other hand,

once the emulator is trained, an emulator run is equivalent to sequentially obtaining T univariate Normal distributions with mean and variance given by (11) and (12), respectively, where T is the maximum number of time steps considered. Most software packages give a near instantaneous answer that would significantly outperform, in terms of run time, sequentially solving the SEIR model and at each time step calculating a time-consuming convolution. Clearly, the investment is on the training of the emulator, which, in Matlab, takes a few hours.

Further, it is worth noting that we found convergence and mixing diagnostics of the MCMC chains for the $\boldsymbol{\eta}$ parameters to be a lot more favorable using the emulator rather than directly using the simulator. In particular, convergence was fast (within the first few thousand iterations) using the emulator, while Birrell et al. (2011) had reported that convergence required hundreds of thousands of simulator runs.

Although the overall performance of the dynamic emulator is good, there is still room for improvement, particularly with regard to the specification of the GP correlation function as well as its mean function components. In situations where the assumption of stationarity over the input space is not appropriate, a more flexible covariance specification mechanism may have to be used. Here, 'Treed GPs' (Gramacy and Lee, 2008) may prove to be useful as they partition the input space into different regions, each having a different stationarity specification, resulting in a non-stationary process over the input space. For time series with cyclical behavior, a 'seasonal trend' component can be included in the mean function of the GP. With regard to the observational errors, ϵ_{z_t}, defined in (13), a more flexible modeling approach could be attained by allowing their distribution to vary with time. One way to accomplish this is by linking the variance of the observational error in the calibration stage to the emulator variance v_t, letting $\epsilon_{z_t} \mid v_t \overset{ind}{\sim} N(0, \sigma^2_{z_t} = \kappa v_t)$, and placing a prior on κ. In fact, we implemented this approach and obtained virtually the same calibration results for $\boldsymbol{\eta}$ presented in Section 4.2.

An important area of future work includes utilizing the emulator's forecasting structure to obtain epidemic trajectories based on the calibrated epidemic model parameters. Another important area includes sequential implementation of the dynamic emulator to carry out real-time calibration for scenarios where batches of data arrive sequentially in time.

Based on the real-time calibration results, the expected behavior of the epidemic may be forecasted and used to guide epidemic management policies.

Appendix A: Technical details for the MCMC algorithm to sample $(\boldsymbol{\phi}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{v}_{1:T}, W_{1:T}, \boldsymbol{\beta})$

The overall MCMC algorithm iterates between Gibbs sampling (using FFBS) for the sequences $(\boldsymbol{\phi}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{v}_{1:T}, W_{1:T})$ and a Metropolis-Hastings step for $\boldsymbol{\beta}$.

Forward-filtering backward-sampling

For simplicity, the model in (6) and (8)may be expressed as

$$\boldsymbol{y}_t = \tilde{\boldsymbol{F}}_t' \boldsymbol{\Phi}_t + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim N\left(\mathbf{0}, v_t \Sigma\right),$$
$$\boldsymbol{\Phi}_t = \boldsymbol{\Phi}_{t-1} + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim N(\mathbf{0}, W_t),$$

where $\tilde{\boldsymbol{F}}_t = (\boldsymbol{F}_t', H)$, $\boldsymbol{\Phi}_t = (\boldsymbol{\phi}_t, \boldsymbol{\lambda}_t)'$, and $\boldsymbol{w}_t = (\boldsymbol{\omega}_{t1}', \boldsymbol{\omega}_{2t}')'$. Then priors are specified for $\boldsymbol{\Phi}_0$, $\{v_t, t = 1, \ldots, T\}$, $\{W_t, t = 1, \ldots, T\}$, and $\boldsymbol{\beta}$.

Let D_0 be the initial set of information, which includes all initial model-defining quantities, and let $D_t = \{y_t, D_{t-1}\}$. Following West and Harrison (1997), we assume that $(v_0^{-1} \mid D_0) \sim \text{Gamma}(n_0/2, d_0/2)$ and $(\boldsymbol{\Phi}_0 \mid v_0, D_0) \sim N(\boldsymbol{m}_0, \boldsymbol{C}_0)$, where $(n_0, d_0, \boldsymbol{m}_0, \boldsymbol{C}_0)$ are pre-specified. The results presented in this paper are based on the vague prior specifications $\boldsymbol{m}_0 = \mathbf{0}$, $n_0 = 1$, $d_0 = 1$ and $C_0 = 5\, I_8$, where I_m denotes the identity matrix of order m. The priors for v_t and W_t are specified sequentially, using two separate variance discount factors δ_v, and δ_W. Discount factors, which take values between 0 and 1, control the 'relative stability' over time of stochastic changes in the sequences v_t and $\boldsymbol{\omega}_t$. Under discounting, $v_t^{-1} \mid D_{t-1} \sim \text{Gamma}(\delta_v n_{t-1}/2, \delta_v d_{t-1}/2)$, and $W_t \mid D_{t-1} = (1 - \delta_W) C_{t-1}/\delta_W$, where $C_{t-1} = \text{Cov}(\boldsymbol{\Phi}_{t-1} \mid D_{t-1})$.

Forward Filtering

To simplify notation, we assume that, within the FFBS algorithm, $\boldsymbol{\beta}$ is conditioned upon and contained in D_t, $t = 1, \ldots, T$. Then, the following are the forward-filter updating equations (West and Harrison, 1997; Liu and West, 2009).

$$(v_{t-1}^{-1} \mid D_{t-1}) \sim \text{Gamma}(n_{t-1}/2, d_{t-1}/2)$$
$$(\boldsymbol{\Phi}_{t-1}|D_{t-1}) \sim T_{n_{t-1}}(\boldsymbol{m}_{t-1}, \boldsymbol{C}_{t-1})$$
$$(v_t^{-1} \mid D_{t-1}) \sim \text{Gamma}(\delta_v n_{t-1}/2, \delta_v d_{t-1}/2)$$
$$(\boldsymbol{\Phi}_t|D_{t-1}) \sim T_{n_{t-1}}(\boldsymbol{a}_t, \boldsymbol{R}_t)$$
$$(\boldsymbol{Y}_t|D_{t-1}) \sim T_{\delta_v n_{t-1}}(\boldsymbol{f}_t, \boldsymbol{Q}_t)$$
$$(v_t^{-1} \mid D_t) \sim \text{Gamma}(n_t/2, d_t/2)$$
$$(\boldsymbol{\Phi}_t|D_t) \sim T_{n_t}(\boldsymbol{m}_t, \boldsymbol{C}_t)$$

$$\boldsymbol{a}_t = \boldsymbol{m}_{t-1} \qquad \boldsymbol{R}_t = \boldsymbol{C}_{t-1}/\delta_W$$
$$\boldsymbol{f}_t = \tilde{\boldsymbol{F}}_t' \boldsymbol{a}_t \qquad \boldsymbol{Q}_t = \tilde{\boldsymbol{F}}_t' \boldsymbol{R}_t \tilde{\boldsymbol{F}}_t + s_{t-1}\Sigma$$
$$s_{t-1} = d_{t-1}/n_{t-1} \qquad s_t = d_t/n_t$$
$$\boldsymbol{m}_t = \boldsymbol{a}_t + \boldsymbol{A}_t \boldsymbol{e}_t \qquad \boldsymbol{C}_t = (\boldsymbol{R}_t - \boldsymbol{A}_t \boldsymbol{Q}_t \boldsymbol{A}_t')s_t/s_{t-1}$$
$$\boldsymbol{A}_t = \boldsymbol{R}_t \tilde{\boldsymbol{F}}_t \boldsymbol{Q}_t^{-1} \qquad \boldsymbol{e}_t = \boldsymbol{Y}_t - \boldsymbol{f}_t$$
$$n_t = \delta_v n_{t-1} + n \qquad d_t = \delta_v d_{t-1} + s_{t-1} \boldsymbol{e}_t' \boldsymbol{Q}_t^{-1} \boldsymbol{e}_t$$

Backward sampling

Forward filtering operates (sequentially) in forward time over $t = 1, \ldots, T$. Backward sampling operates in reverse time, generating posterior draws from $p(\boldsymbol{\Phi}_{1:T}, \boldsymbol{v}_{1:T} \mid D_T)$, where D_T represents all available information at the last time step, $t = T$. Having completed forward filtering, the distribution $p(v_T^{-1} \mid D_T)$ is known. Specifically,

$$v_T^{-1} \mid D_T \sim \text{Gamma}(n_T/2, d_T/2)$$

Then, it can be shown (Liu, 2007) that $v_t \mid v_{t+1}, D_t$, $t = T-1, \ldots, 1$, may be sampled

recursively from

$$v_t^{-1} \sim \delta_v v_{t+1}^{-1} + \text{Gamma}\big((1-\delta_v)/2, d_t/2\big),$$

Having completed forward filtering, and having obtained the backward samples for $\boldsymbol{v}_{1:T}$, the distribution of $p(\boldsymbol{\Phi}_T, \boldsymbol{v}_{1:T} \mid D_T)$ is known. Specifically,

$$\boldsymbol{\Phi}_T \mid \boldsymbol{v}_{1:T}, D_T \sim N(\boldsymbol{m}_T, \boldsymbol{C}_T)$$

Then, it can be shown (Liu, 2007) that $\boldsymbol{\Phi}_t$, $t = T-1, \ldots, 1$ may be sampled recursively from

$$\boldsymbol{\Phi}_t \mid \boldsymbol{\Phi}_{t+1}, \boldsymbol{v}_{1:T}, D_t \sim N\big((1-\delta_W)\boldsymbol{m}_t + \delta_W \boldsymbol{\Phi}_{t+1}\,,\, (1-\delta_W)\boldsymbol{C}_t\big)$$

Metropolis-Hastings step for $\boldsymbol{\beta}$

The posterior full conditional distribution of the correlation ranges $\boldsymbol{\beta}$ is given by

$$\begin{aligned} p(\boldsymbol{\beta} \mid \boldsymbol{\Phi}_{1:T}, \boldsymbol{v}_{1:T}, \boldsymbol{y}_{1:T}) &\propto p(\boldsymbol{y}_{1:T} \mid \boldsymbol{\Phi}_{1:T}, \boldsymbol{v}_{1:T}, \boldsymbol{\beta}) p(\boldsymbol{\beta}) \\ &\propto \mid \Sigma \mid^{(T-p)/2} \exp\left(-\frac{1}{2} \sum_{t=p+1}^{T} (\boldsymbol{y}_t - \tilde{\boldsymbol{F}}' \boldsymbol{\Phi}_t)' \Sigma^{-1} (\boldsymbol{y}_t - \tilde{\boldsymbol{F}}' \boldsymbol{\Phi}_t)/v_t \right) p(\boldsymbol{\beta}) \end{aligned}$$

Thus, $\boldsymbol{\beta}$ is sampled using a Metropolis-Hastings step.

Appendix B: Technical details for the calibration stage

Consider (18). Then, $p(\boldsymbol{\eta}, \sigma_z^2 \mid \boldsymbol{\xi}_{1:T}, D^s, \boldsymbol{z}_{1:T})$ is proportional to

$$\begin{aligned} & p(\boldsymbol{\eta}) p(\sigma_z^2) \prod_{t=p+1}^{T} p(z_t, \boldsymbol{y}_t \mid \boldsymbol{\eta}, \sigma_z^2, \boldsymbol{\xi}_t, \boldsymbol{z}_{t-1:t-p}, \boldsymbol{y}_{t-1:t-p}) \\ =& p(\boldsymbol{\eta}) p(\sigma_z^2) \prod_{t=p+1}^{T} p(z_t \mid \boldsymbol{y}_t, \boldsymbol{\eta}, \sigma_z^2, \boldsymbol{\xi}_t, \boldsymbol{z}_{t-1:t-p}, \boldsymbol{y}_{t-1:t-p}) p(\boldsymbol{y}_t \mid \boldsymbol{\xi}_t, \boldsymbol{y}_{t-1:t-p}) \\ \propto& p(\boldsymbol{\eta}) p(\sigma_z^2) \prod_{t=p+1}^{T} p(z_t \mid \boldsymbol{y}_t, \boldsymbol{\eta}, \sigma_z^2, \boldsymbol{\xi}_t, \boldsymbol{z}_{t-1:t-p}, \boldsymbol{y}_{t-1:t-p}), \end{aligned}$$

where $p(z_t \mid \boldsymbol{y}_t, \boldsymbol{\eta}, \sigma_z^2, \boldsymbol{\xi}_t, \boldsymbol{z}_{t-1:t-p}, \boldsymbol{y}_{t-1:t-p})$ is Normal with mean and variance given respectively by

$$\mathbb{M}_t = M_t(\boldsymbol{\eta}) + \boldsymbol{r}'(\boldsymbol{\eta})\Sigma^{-1}(\boldsymbol{y}_t - \boldsymbol{M}_t(X)), \text{ and}$$
$$\mathbb{S}_t = (\sigma_z^2 + v_t) - v_t \boldsymbol{r}'(\boldsymbol{\eta})\Sigma^{-1}\boldsymbol{r}(\boldsymbol{\eta}).$$

Then,

$$\log p(\boldsymbol{\eta}, \sigma_z^2 \mid \boldsymbol{\xi}_{1:T}, D^S, \boldsymbol{z}_{1:T}) \propto -\frac{1}{2}\sum_{t=p+1}^{T} \log \mathbb{S}_t - \frac{1}{2}\sum_{t=p+1}^{T} (z_t - \mathbb{M}_t)^2 / \mathbb{S}_t + \log p(\boldsymbol{\eta}) + \log p(\sigma_z^2)$$

Thus, once priors are specified for $\boldsymbol{\eta}$ and σ_z^2, their posterior distribution may be sampled using one (or two) Metropolis-Hastings step(s) at every MCMC sample from Stage 1.

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 Influenza A/H1N1pdm Dynamics in London," *Proceedings of the National Academy of Sciences*, 108, 18238–18243.

Brauer, F. and Castillo-Chavez, C. (2011), *Mathematical Models in Population Biology and Epidemiology*, Texts in Applied Mathematics, New York, USA: Springer, 2nd ed.

Conti, S., Gosling, J. P., Oakley, J. E., and O'Hagan, A. (2009), "Gaussian Process Emulation of Dynamic Computer Codes," *Biometrika*, 96, 663–676.

Conti, S. and O'Hagan, A. (2010), "Bayesian Emulation of Complex Multi-Output and Dynamic Computer Models," *Journal of Statistical Planning and Inference*, 140, 640–651.

Farah, M. (2013), "Bayesian Nonparametric Methods for Emulation, Sensitivity analysis, and Calibration of Computer Simulators," Ph.D. thesis, University of California, Santa Cruz, to appear.

Fricker, T. E., Oakley, J. E., and Urban, N. M. (2013), "Multivariate Gaussian Process Emulators With Nonseparable Covariance Structures," *Technometrics*, 55, 47–56.

Gramacy, R. B. and Lee, H. K. H. (2008), "Bayesian Treed Gaussian Process Models with an Application to Computer Modeling," *Journal of the American Statistical Association*, 103, 1119–1130.

Henderson, D. A., Boys, R. J., Krishnan, K. J., Lawless, C., and Wilkinson, D. J. (2009), "Bayesian Emulation and Calibration of a Stochastic Computer Model of Mitochondrial DNA Deletions in Substantia Nigra Neurons," *Journal of the American Statistical Association*, 104, 76–87.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008a), "Computer Model Calibration Using High-Dimensional Output," *Journal of the American Statistical Association*, 103, 570–583.

Higdon, D., Kennedy, M. C., Cavendish, J., Cafeo, J., and Ryne, R. D. (2004), "Combining Field Data and Computer Simulations for Calibration and Prediction," *SIAM Journal on Scientific Computing*, 26, 448–466.

Higdon, D., Nakhleh, C., Gattiker, J., and Williams, B. (2008b), "A Bayesian calibration approach to the thermal problem," *Computer Methods in Applied Mechanics and Engineering*, 197, 2431–2441.

Jacquez, J. A. (1996), *Compartmental analysis in biology and medicine*, 3rd edition, BioMedware, Ann Arbor.

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.

— (2001), "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63, 425–464.

Koehler, J. R. and Owen, A. B. (1996), "Computer Experiments," in *Design and Analysis of Experiments*, Amsterdam: North-Holland, vol. 13 of *Handbook of Statistics*, pp. 261–308.

Korostil, I. A., Peters, G. W., Cornebise, J., and Regan, D. G. (2012), "Adaptive Markov Chain Monte Carlo Forward Projection for Statistical Analysis in Epidemic Modelling of Human Papillomavirus," *Statistics in Medicine*, 1–37.

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., and Ye, K. Q. (2006), "Variable Selection for Gaussian Process Models in Computer Experiments," *Technometrics*, 48, 478–490.

Liu, F. (2007), "Bayesian Functional Data Analysis for Computer Model Validation," Ph.D. thesis, Duke University.

Liu, F., Bayarri, M. J., and Berger, J. O. (2009), "Modularization in Bayesian Analysis, with Emphasis on Analysis of Computer Models," *Bayesian Analysis*, 4, 119–150.

Liu, F. and West, M. (2009), "A Dynamic Modelling Strategy for Bayesian Computer Model Emulation," *Bayesian Analysis*, 87, 393–412.

Loeppky, J. L., Bingham, D., and Welch, W. J. (2006), "Computer Model Calibration or Tuning in Practice," Tech. rep., University of British Columbia.

Oakley, J. E. and O'Hagan, A. (2004), "Probabilistic Sensitivity Analysis of Complex Models: A Bayesian Approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66, 751–769.

O'Hagan, A. (2006), "Bayesian Analysis of Computer Code Outputs: A Tutorial," *Reliability Engineering & System Safety*, 91, 1290–1300.

Reichert, P., White, G., Bayarri, M. J., and Pitman, E. B. (2011), "Mechanism-Based Emulation of Dynamic Simulation Models: Concept and Application in Hydrology," *Computational Statistics & Data Analysis*, 55, 1638–1655.

Rougier, J. (2008), "Efficient Emulators for Multivariate Deterministic Functions," *Journal of Computational and Graphical Statistics*, 17, 827–843.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and Analysis of Computer Experiments (C/R: p423-435)," *Statistical Science*, 4, 409–423.

Saltelli, A., Chan, K., and Scott, E. M. (eds.) (2000), *Sensitivity Analysis*, Wiley Series in Probability and Statistics, Chichester, UK: John Wiley & Sons Ltd.

Santner, T., Williams, B., and Notz, W. (2003), *The Design and Analysis of Computer Experiments*, Springer-Verlag.

Schlather, M. (1999), "Introduction to Positive Definite Functions and to Unconditional Simulation of Random Fields," Tech. Rep. ST-99-10, Department of Mathematics and Statistics, Lancaster University, UK.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992), "Screening, Predicting, and Computer Experiments," *Technometrics*, 34, 15–25.

West, M. and Harrison, J. (1997), *Bayesian Forecasting and Dynamic Models*, Springer, 2nd ed.