# Supplementary Material for *Quantifying simulator discrepancy in discrete-time dynamical simulators*

Richard D. Wilkinson[1, *], Michail Vrettas[1], Dan Cornford[2], and Jeremy E. Oakley[3]

[1]*School of Mathematical Sciences, University of Nottingham,University Park Nottingham NG7 2RD, United Kingdom. Email: r.d.wilkinson@nottingham.ac.uk. Telephone: 00441158467413.*
*Corresponding author
[2]*School of Engineering and Applied Science, Aston University, Birmingham B4 7ET, United Kingdom*
[3]*School of Mathematics and Statistics, The University Of Sheffield, Sheffield S3 7RH, United Kingdom*

August 25, 2011

This document contains supplementary material accompanying the paper '*Quantifying simulator discrepancy in discrete-time dynamical simulators*'. Section 1 contains details of the implementation of the particle filter used in the inferential algorithm. Section 2 contains an additional case study demonstrating the methodology developed in the main paper.

# 1 The Particle Filter

To sequentially generate observations from $\pi(x_{1:t} \mid y_{1:t})$ we use the bootstrap particle filter (Doucet et al., 2001). This is a sequential importance resampling algorithm, which approximates the filtering distributions $\pi(x_{1:t}|y_{1:t})$ by a weighted sample of $N$ particles $\{(x_{1:t}^{(j)}, w_t^{(j)})\}_{j=1}^{N}$ with $\sum_{j=1}^{N} w_t^{(j)} = 1$, so that any expectation can be approximated by a weighted sum:

$$\mathbb{E}(h(x_{0:t})|y_{0:t}) = \int h(x_{0:t})\pi(x_{0:t}|y_{0:t})\mathrm{d}x_{0:t} \approx \sum_{j=1}^{N} w_t^{(j)} h(x_{0:t}^{(j)}).$$

**Bootstrap particle filter**

**t = 1** (i) Initialize: For $i = 1, \ldots, N$

$$x_1^{(i)} \sim \pi(x_1)$$
$$w_1^{(i)} = \pi_1(y_1 \mid x_1^{(i)})$$

(ii) Normalise: set $W_1^{(i)} = \frac{w_1^{(i)}}{\sum_{j=1}^{N} w_1^{(j)}}$, to obtain a weighted sample $\{W_1^{(i)}, x_1^{(i)}\}$ approximating $\pi(x_1 \mid y_1)$.

(iii) Resample: if ESS$< T$, resample the particles and set $W_1^{(i)} = 1/N$ for all $i$. Set $t = 2$.

**t ≥ 2** (i) Simulate: For $i = 1, \ldots, N$

$$x_t^{(i)} \sim \pi(x_t \mid x_{t-1}^{(i)})$$
$$w_t^{(i)} = W_{t-1}^{(i)} \pi(y_t \mid x_t)$$

(ii) Normalise: set $W_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^N w_t^{(j)}}$, to obtain a weighted sample $\{W_t^{(i)}, x_{1;t}^{(i)}\}$ approximating $\pi(x_{1:t} \mid y_{1:t})$.

(iii) Resample: if ESS$<T$, resample and set $W_t^{(i)} = 1/N$ for all $i$. Set $t = t + 1$.

All distributions are conditional on $\theta$. The effective sample size (ESS) of the weighted sample is $\left(\sum_{i=1}^N (W_t^{(i)})^2\right)^{-1}$, and $T$ denotes the threshold we use to decide when to resample (typically we take $T = N/2$). We used a systematic resampling scheme (Kitagawa, 1996) throughout. To get a smoothed trajectory from $\pi(x_{1:t} \mid y_{1:T})$ we pick a single trajectory from $\{x_{1:T}^{(i)}\}$ according to the weights $\{W_T^{(i)}\}$.

To avoid the problem of degeneracy we can either use a particle smoother, such as that suggested by Godsill et al. (2004), or instead run the filter multiple times. We have tried both methods, but found it to be easier to simply run multiple independent particle filters allowing for easier parallelization.

# 2 Case study 2: Free fall

We consider an idealised example where we assume we know the true behaviour of the system and use an incorrect model as a simulator, and then infer the simulator discrepancy using noisy observations. For the true model, consider an object in free fall near the surface of the earth. This is a two-dimensional system described by a displacement velocity state-vector $(x, v)$. Assuming the object has constant acceleration $g$ and is subject to Stokes' drag with coefficient $k$, the differential equations determining the true system behaviour are

$$\frac{\mathrm{dv}}{\mathrm{dt}} = g - kv, \qquad \frac{\mathrm{dx}}{\mathrm{dt}} = v. \tag{1}$$

We assume $x$ is observed at regular points in time ($t = 0, \Delta t, 2\Delta t, \ldots$) with zero mean Gaussian measurement error. We let $x_n$ and $v_n$ denote the position and velocity when the $n^{th}$ observation, $y_n$, is made

$$y_n \sim \mathcal{N}(x_n, \sigma^2_{obs}).$$

Equations (1) imply that the one-step-ahead dynamic updates for the system are

$$\begin{aligned} x_{n+1} &= x_n + \frac{1}{k}(\frac{g}{k} - v_n)(\mathrm{e}^{-\mathrm{k}\Delta\mathrm{t}} - 1) + \frac{\mathrm{g}\Delta\mathrm{t}}{\mathrm{k}}, \\ v_{n+1} &= (v_n - \frac{g}{k})\mathrm{e}^{-\mathrm{k}\Delta\mathrm{t}} + \frac{\mathrm{g}}{\mathrm{k}}. \end{aligned} \tag{2}$$

For the (incorrect) simulator, we suppose that the modellers neglected to include air resistance ($k = 0$ in Equation (1)), so that the simulator has one-step-ahead dynamics

$$\begin{aligned} x_{n+1} &= x_n + v_n\Delta t + \frac{1}{2}g(\Delta t)^2 \\ v_{n+1} &= v_n + g\Delta t. \end{aligned} \tag{3}$$

The discrepancy function $\boldsymbol{\delta}(x, v)$ for the difference between the system dynamics and the simulator dynamics can be calculated explicitly in this case, giving

$$\boldsymbol{\delta}(x,v) = \frac{g}{k}\left(\begin{array}{c} \frac{1}{k}(\mathrm{e}^{-\mathrm{k}\Delta \mathrm{t}}-1)+\Delta\mathrm{t}-\frac{1}{2}\mathrm{k}(\Delta\mathrm{t})^2 \\ 1-k\Delta t-\mathrm{e}^{-\mathrm{k}\Delta\mathrm{t}} \end{array}\right) - v\left(\begin{array}{c} \frac{1}{k}(\mathrm{e}^{-\mathrm{k}\Delta \mathrm{t}}-1)+\Delta\mathrm{t} \\ 1-\mathrm{e}^{-\mathrm{k}\Delta\mathrm{t}} \end{array}\right), \tag{4}$$

which is a linear function in $v$, with no dependence on $x$. The discrepancy illustrates the difficulty faced. Equation (4) depends solely on the velocity, not the displacement, yet we only observe the displacement. Hence, learning $\boldsymbol{\delta}$ relies upon our ability to infer the velocity trajectory, which can then be used to train the discrepancy model.

The discrepancy $\boldsymbol{\delta}$ is a map from $\mathbb{R}^2$ to $\mathbb{R}^2$, which we denote as

$$\boldsymbol{\delta}(x,v) = \left(\begin{array}{c} \delta_x(x,v) \\ \delta_v(x,v) \end{array}\right).$$

Note that although the true discrepancy is deterministic, we need to use a stochastic model for $\boldsymbol{\delta}$, because measurement error and the finite number of observations make the estimate of $\boldsymbol{\delta}$ uncertain. A statistical model is used in order to describe this uncertainty. In more complex situations, we may not expect a deterministic discrepancy function to exist (for example the state vector may not contain enough information to fully model the system dynamics, or if the system is stochastic), and so a statistical model will be vital.

Expert opinion can be useful when deciding which family of parametric models we should use for $\delta$. In this case, universality allows us to argue that the discrepancy will not depend on $x$ so long as we are near the surface of the earth. So in this case, we could have proposed a model for $\boldsymbol{\delta}$ that depended only on $v$. Furthermore, someone experienced with both the system and the simulator may be able to make rudimentary comments about the shape of $\boldsymbol{\delta}$, although expert elicitation of simulator error can be challenging (Goldstein and Rougier, 2009; Vernon et al., 2010).

We begin by fitting a model of the correct parametric form, namely

$$\boldsymbol{\delta}(x, v) = \begin{pmatrix} a_x + b_x v \\ a_v + b_v v \end{pmatrix} + \begin{pmatrix} e_x \\ e_v \end{pmatrix} \tag{5}$$

where $e_x \sim \mathcal{N}(0, \tau_x)$ and $e_v \sim \mathcal{N}(0, \tau_v)$, and estimate the six unknown parameters $(a_x, b_x, \tau_x, a_v, b_v, \tau_v)$. We used a time series of 100 noisy observations of the true system generated with air-resistance fixed at $k = 0.1$ and a time step between observations of $\Delta t = 0.5$. We allowed measurement error to vary to show the effect on our ability to accurately estimate $\boldsymbol{\delta}$. We used 1000 filtering particles and five smoothed trajectories in the inference scheme. The stopping rule used to decide when to terminate the iterations in the EM algorithm, was to look at the last five estimates for a particular parameter and to test whether all five estimates are within 0.001 of each other. We stopped the EM algorithm when this condition was met simultaneously for all six parameters.

The results are shown in Table 1. The first thing to note is that the estimated coefficient of $v$ in the $\delta_v$ discrepancy, $b_v$, always matches the true value to two significant figures. The other parameter estimates in $\delta_x$ and $\delta_v$ are of the right magnitude and sign, but are further from the true values.

| $\sigma_{obs}$ | #iters. | $\delta_x$ estimate | $\delta_v$ estimate | $\tau_x$ | $\tau_v$ |
|---|---|---|---|---|---|
| 0.01 | 17 | $-0.019 - 0.0062v$ | $-0.18 - 0.049v$ | $3.5 \times 10^{-5}$ | $2.0 \times 10^{-5}$ |
| 0.1 | 30 | $-0.014 - 0.0062v$ | $-0.18 - 0.049v$ | $2.7 \times 10^{-4}$ | $9.9 \times 10^{-5}$ |
| 0.5 | 173 | $-0.019 - 0.0062v$ | $-0.18 - 0.049v$ | $1.6 \times 10^{-4}$ | $9.5 \times 10^{-6}$ |
| 1 | 215 | $-0.066 - 0.0056v$ | $-0.17 - 0.049v$ | $2.9 \times 10^{-4}$ | $7.3 \times 10^{-6}$ |
| 5 | 267 | $-0.037 - 0.0058v$ | $-0.15 - 0.049v$ | $1.3 \times 10^{-4}$ | $1.3 \times 10^{-4}$ |
| 10 | 519 | $-0.050 - 0.0059v$ | $-0.19 - 0.049v$ | $8.7 \times 10^{-6}$ | $2.2 \times 10^{-4}$ |

Table 1: Parameter estimates found using Equation (5) as the discrepancy function. $\sigma_{obs}$ is the standard deviation of the Gaussian measurement error used, #iters. is the number of iterations required by the EM algorithm in order to reach convergence, and $\tau_x$ and $\tau_v$ are the estimates of the variances of the Gaussian white noise part of the discrepancies for $x$ and $v$. All values are reported to two significant figures. The true values of the two discrepancy functions are $\delta_x(x, v) = -0.020 - 0.012v$ and $\delta_v(x, v) = -0.12 - 0.049v$, with $\tau_x = \tau_v = 0$ in both cases.

We believe parameter $b_v$ is well estimated because it is the key determining factor for the simulator discrepancy and if we estimate $v$ incorrectly at stage $n$, the subsequent estimate of $x$ will be wrong at time $n + 1$. The parameters estimates for $b_x$ and $a_v$ appear to be biased, but this appears to be a quirk of the time-period and number of observations used. The two variance parameters, $\tau_x$ and $\tau_v$, are unable to reach zero, because the other parameters are not accurately estimated and so the variance is inflated to account for this.

More precise estimates of the parameters could be found by replacing the crude stopping rule used to terminate the EM algorithm, and using a larger number of filtering and smoothing particles. However the aim of this paper is to show the improvement that can be made to the forecasting system, and the estimates above are more than sufficient to do this. If we take these discrepancy estimates and use them in a forecasting system, the improvements in predictive power immediately become clear. Table 2 shows the performance

| Forecasting framework | MSE | NS (%) | CRPSS (%) |
|---|---|---|---|
| Simulator only | $9.57 \times 10^6$ | -585.2 | -124.86 |
| Simulator only, corrected | 1708 | 99.9 | 1.94 |
| Simulator plus white noise | 11900 | 99.1 | 95.03 |
| Sim. + discrep. (line 2 Table 1) | 0.0296 | 100.0 | 99.99 |

Table 2: Predictive performance of various different forecasting systems, as measured by the mean-square-error (MSE), the Nash-Sutcliffe statistic (NS), and the continuously ranked probability skill score (CRPSS). Low values of MSE indicate good predictive performance, as do NS and CRPSS values close to 100%. The reference forecast used for the NS statistic and the CRPSS was a Gaussian distribution with mean and variance estimated from the observations. Values of NS and CRPSS greater than 0 indicate the predictions are superior to the reference forecast.

of various different forecasting systems. The test data used was a sequence of 100 new observations (with measurement error $\sigma^2_{obs} = 0.1$) generated from different starting values of $x$ and $v$ to those used in the training data (we used an object fired upwards which then decelerates before falling back to the ground). The first row of the table contains the forecast error from simply running the deterministic simulator from the initial conditions and adding measurement error. The second row was found by using the true value of the state vector with the deterministic simulator to predict the next observation (note this would not be possible in a real situation). The third row is the result of using the simulator plus a white noise simulator discrepancy, with variances estimated from the data to be $\tau_x = 3084$ and $\tau_v = 27.4$. The fourth row is the simulator plus the discrepancy estimated in Table 1 when $\sigma_{obs} = 0.1$ (row 2). The results show the vast improvement in predictive power that can be achieved in this case by training a discrepancy term.

These results also highlight the danger of relying on a single diagnostic measure to judge predictive performance. The Nash-Sutcliffe statistic indicates very similar (and excellent) performance for the simulator plus white noise, and the simulator plus discrepancy frameworks because it only judges the mean prediction of the simulator. The CRPSS also takes into account the uncertainty quantification of the forecasts, and show that superior predictions are made by the full simulator plus discrepancy framework.

Figure 1 shows the forecast errors $y_{t+1} - \mathbb{E}(y^{\text{rep}}_{t+1}|y_{1:t})$ plotted against the fitted values $\mathbb{E}(y^{\text{rep}}_{t+1}|y_{1:t})$, where $y^{\text{rep}}_{t+1}$ denotes theoretical repetitions of the t+1$^{th}$ observation assuming the model is true. These plots are the analogue of the residual plots used as diagnostic tools in linear regression. If the framework is correct, we should see the residuals form an uncorrelated band distributed symmetrically about $y = 0$. Only the residual plot for the full discrepancy model looks like this. The tail near 2000 appears because these observations are made at low velocities where the discrepancy correction is less effective. The other plots all show a high degree of correlation between the errors, indicating problems with the forecasting system.

This case-study has shown the ability of a discrepancy function to improve forecast accuracy and we have demonstrated that the methodology works in this case. Although this was a simple example, it has demonstrated the difficulty involved in making such inferences. Here we successfully inferred the structure of a discrepancy which depends on a variable (v) that is never observed. Finally, note that the dynamics and observation process are linear and Gaussian, and so in this case the Kalman smoother could be used rather than the particle filter. This would be much more efficient and would significantly decrease the computation time.

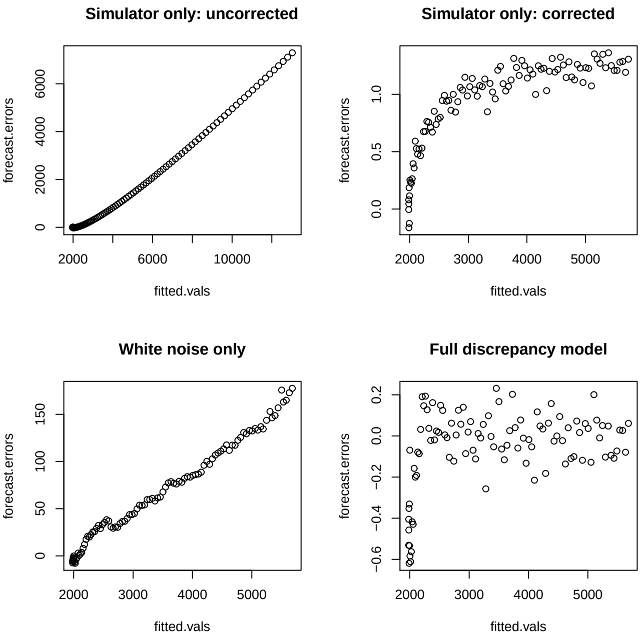

Figure 1: A residual plot showing the one-step-ahead forecast errors versus the fitted values. The four plots correspond to the first four cases described in Table 2.

# References

Doucet, A., de Freitas, N., Gordon, N., 2001. Sequential Monte Carlo Methods in Practice. Springer.

Godsill, S., Doucet, A., West, M., 2004. Monte Carlo smoothing for nonlinear time series. J. Am. Stat. Soc. 99 (465), 156–168.

Goldstein, M., Rougier, J., 2009. Reified Bayesian modelling and inference for physical systems (with discussion). J. Stat. Plan. Infer. 139, 1221–1239.

Kitagawa, G., 1996. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. J. Comput. Graph. Stat. 5, 1–25.

Vernon, I. R., Goldstein, M., Bower, R. G., 2010. Galaxy formation: a Bayesian uncertainty analysis. Bayesian Analysis 5, 619–670.