Control Variates for Constrained Variables

Simon Maskell, *Member, IEEE*, Yifan Zhou, and Antonietta Mira

***Abstract*—Numerical Bayesian inference methods typified by Markov chain Monte Carlo generate a set of samples from a probability distribution. When using real-valued samples to approximate the expectation of a random variable, the variance of the resulting estimator, obtained by averaging over those samples, decreases as the number of samples increases. However, it is often useful to reduce the variance without increasing the number of samples. Using control variates is one method to achieve such variance reduction and is applicable in contexts where the random variable is unconstrained. To make it possible to use control variates with constrained variables, this paper proposes the use of a non-linear mapping from an unconstrained space to the constrained space. Results indicate that significant reductions in Monte-Carlo error is achieved with negligible additional computational cost.**

***Index Terms*—Constraints, Control Variates, Markov chain Monte Carlo, Variance Reduction, Zero Variance.**

I. Introduction

Numerical Bayesian inference involves drawing samples from a target distribution, which is typically the posterior distribution of parameters given data, in order to make inferences from such data[1]: such inferences can include deriving point estimates, identifying credible intervals and hypothesis testing. The broad applicability of this generic approach has given rise to a diverse range of applications, spanning numerous aspects of signal processing (see, for example, [4], [5] and [6] for specific examples in image processing, deconvolution and interference mitigation respectively) but also pertinent in the context of, for example, health[7] and finance[8]. This paper's contribution is applicable across these contexts.

Furthermore, there are a growing number of probabilistic programming languages (e.g. Stan[9], PyMC3[10], Figaro[11] and Turing.jl[12]) being adopted with the express purpose of easing the process of exploiting numerical Bayesian inference to arbitrary applications. The majority of these implementations make use of Markov chain Monte Carlo and more specifically, the No-U-Turn-Sampler (NUTS)[13].

If $x^{(i)}$ is the ith of N samples output by a numerical Bayesian inference algorithm that is sampling from $p_x\left(x|d\right)$, where x is the parameter being inferred and d is some (fixed) data (for which we know the likelihood), then we can approximate the expectation of a function, $f_x\left(x\right)$ as:

$$\int_{x=l}^{u} p_x\left(x|d\right) f_x\left(x\right) dx \approx \frac{1}{N}\sum_{i=1}^{N} f_x\left(x^{(i)}\right). \tag{1}$$

This work was supported by the Engineering and Physical Research Council (Grant number EP/R018537/1).

Simon Maskell and Yifan Zhou are with the department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 9GJ, United Kingdom (e-mail: {s.maskell, yifan.zhou}@liverpool.ac.uk).

Antonietta Mira is with the University of Lugano, 6900 Lugano, Switzerland and Insubria University, Como, Italy (e-mail: antonietta.mira@usi.ch).

[1]Accessible introductory texts include [1], [2], [3].

The Monte-Carlo variance, i.e. the standard error[2], of such an estimator is $\frac{\left(\sigma_f\right)^2}{N}$ where σ_f is the standard deviation of $f_x\left(.\right)$. Note that even if the Monte-Carlo variance were zero, the variance of the $p_x\left(x|d\right)$ will limit the accuracy of any estimate of $f_x\left(x\right)$. None-the-less, particularly in contexts where repeatability is desirable, it can be of significant interest to minimise the Monte-Carlo variance. This can be achieved by simply having a larger N. Unfortunately, however, increasing N is, in general, computationally expensive.

An alternative is to attempt to make better use of the available N samples via post-processing. When the parameters to be inferred are continuous, this can be achieved by defining a random variable such that it has an expected value of zero. While the sample plus the 'zero mean' random variable will not alter the expectation, we can design the random variable to be negatively correlated with the sample and so reduce the variance of the estimate[14]. This concept has been extended to consider, for example, terms derived from a quadratic function of the values and the samples[15], [16], [17], [18] as well as more complex functionals[19]. The class of such methods is known as 'control variates'.

While control variates have been demonstrated to offer significant reduction in variance (we note there can, albeit infrequently, be circumstances where no variance reduction is achieved), the approaches that have been developed are not applicable in contexts involving constrained variables. The key contribution made by this paper is to enable control variates to be applied in such constrained contexts by using a mapping from the constrained space to an unconstrained space and then posing expectations in terms of non-linear functions of the unconstrained samples. We consider two kinds of control variates in this paper, those referred to previously in the literature as linear and quadratic control variates.

The paper is organised as follows: Section II proceeds with a description of control variates, highlighting that the samples need to be unconstrained; Section III then explains how a non-linear mapping can be introduced such that the approach can be applied to constrained variables; results are then presented in Section IV before conclusions are drawn in Section V.

II. Control Variates

A. Linear Control Variates

For notational convenience and ease of exposition in the context of explaining control variates to a potentially unfamiliar reader, we begin by considering x to be scalar. We go on to consider vector-valued x in Section II-C.

[2]We emphasise that the standard error is the uncertainty associated with estimation accuracy of the mean, whereas the standard deviation is the uncertainty associated with the distribution of samples about the mean.

We assume that x is bounded such that $l \leq x \leq u$ for finite l and u. We note that we can calculate the expectation (with respect to samples from the posterior) of the derivative of the log (unnormalised) target density as follows:

$$\int_{x=l}^{u} p_x\left(x|d\right) \frac{d \log p_x\left(x,d\right)}{dx} dx$$
$$= \int_{x=l}^{u} p_x\left(x|d\right) \frac{d \log \left[p_x\left(x|d\right) p\left(d\right)\right]}{dx} dx \quad (2)$$
$$= \int_{x=l}^{u} p_x\left(x|d\right) \frac{d \log \left[p_x\left(x|d\right)\right] + \log\left[p\left(d\right)\right]}{dx} dx \quad (3)$$
$$= \int_{x=l}^{u} p_x\left(x|d\right) \frac{d \log \left[p_x\left(x|d\right)\right]}{dx} dx \quad (4)$$
$$= \int_{x=l}^{u} \underbrace{p_x\left(x|d\right) \frac{1}{p_x\left(x|d\right)}}_{=1} \frac{dp_x\left(x|d\right)}{dx} dx \quad (5)$$
$$= p_x\left(u|d\right) - p_x\left(l|d\right) \quad (6)$$

such that if $p_x\left(u|d\right) = p_x\left(l|d\right) = 0$ then the expectation of the derivative of the log density is zero. Note that if x is unconstrained, $p_x\left(x|d\right)$ must tend to zero as $|x| \to \infty$ (since $\int p_x\left(x|d\right) dx = 1$, i.e. is finite, $p_x\left(x|d\right)$ cannot tend to a non-zero value when $|x| \to \infty$).

B. Quadratic Control Variates

A similar argument can be used in the context of the expectation of the product of an (assumed scalar for notational convenience) state and the derivative of the log density:

$$\int_{x=l}^{u} p_x\left(x|d\right) x \frac{d \log p_x\left(x,d\right)}{dx} dx = \int_{x=l}^{u} x \frac{dp_x\left(x|d\right)}{dx} dx \quad (7)$$
$$= u p_x\left(u|d\right) - l p_x\left(l|d\right) - \underbrace{\int_{x=l}^{u} p_x\left(x|d\right) dx}_{=1} \quad (8)$$

where, much as before, if x is unconstrained, $x p_x\left(x|d\right)$ must tend to zero as $|x| \to \infty$ and $p\left(x|d\right)$ has a finite mean.

Let us consider now a bivariate $x = \left[x_1, x_2\right]^T$, and a control variate consisting of the following expectation

$$\int_{x_1=l_1}^{u_1} \int_{x_2=l_2}^{u_2} p_x\left(x_1, x_2|d\right) x_1 \frac{d \log p_x\left(x_1, x_2, d\right)}{dx_2} dx_2 dx_1 \quad (9)$$
$$= \int_{x_1=l_1}^{u_1} u_2 p_x\left(x_1, l_2|d\right) - l_2 p_x\left(x_1, u_2|d\right) dx_1 \quad (10)$$
$$= u_2 p_x\left(u_2|d\right) - l_2 p_x\left(l_2|d\right) \quad (11)$$

which, once again, is zero if $p\left(x|d\right)$ has a finite mean (since $\int_{x_1=l_1}^{u_1} p_x\left(x_1, x_2|d\right) dx_1 = p_x\left(x_2|d\right)$).

C. Vectors of Control Variates

Given that the ith vector-valued sample from the MCMC is $x^{(i)} \in \mathbf{R}^D$, we can then form a vector of control variates, all of which have zero mean with respect to the target distribution, by considering

$$z_i^x = \left[\; z_{i,1}^x \cdots z_{i,D}^x, z_{i,(1,1)}^x \cdots z_{i,(1,D)}^x, z_{i,(2,2)}^x \cdots z_{i,(D,D)}^x \; \right]^T \quad (12)$$

where $z_{i,(j,k)}^x$ is only present for $k \geq j$ and where[3]

$$z_{i,j}^x = \frac{d \log \left[p_x\left(x^{(i)}|d\right)\right]}{dx_j} \quad (13)$$
$$z_{i,(j,j)}^x = x_j^{(i)} z_{i,j}^x - 1 \quad (14)$$
$$z_{i,(j,k)}^x = x_k^{(i)} z_{i,j}^x + x_j^{(i)} z_{i,k}^x. \quad (15)$$

Note that there are $D + \frac{1}{2}D \times (D-1) = \frac{1}{2}D \times (D+1)$ control variates. Since this scales quadratically with D, if D is large, it can be convenient to not calculate $z_{i,(j,k)}^x$ for $j \neq k$.

D. Reducing Estimation Variance

If we wish to calculate an expectation of a function $f_x\left(x\right)$, have N samples from $p_x\left(x|d\right)$ and can calculate $\frac{d \log[p_x(x|d)p(d)]}{dx}$ then we can improve on the variance of a sample-based approximation of the expectation of $f_x\left(x\right)$ by using the following estimator:

$$\int_{x=l}^{u} p_x\left(x|d\right) f_x\left(x\right) dx \approx \frac{1}{N} \sum_{i=1}^{N} \left[f_x\left(x^{(i)}\right) + \alpha z_i^x \right] \quad (16)$$

where we refer to the variance of the summand as the standard error for this estimator and where, e.g. as a result of using MCMC, $x^{(i)} \sim p_x\left(x|d\right)$, z_i^x has an expectation of zero (as explained in Section II-C) such that adding αz_i^x does not introduce a bias. Note that $\alpha = 0$ is not always the value that minimises the estimation variance. Indeed, we can show that the optimal (with respect to a least-squares criterion: regularisation can be beneficial[16]) value for α is:

$$\alpha^\star = -\sigma_{fz} {\sigma_{zz}}^{-1} \quad (17)$$

where σ_{fz} is the cross-correlation between $f_x\left(x\right)$ and z, and σ_{zz} is the corresponding covariance of z (both of which need to be estimated from the samples). Note that this approach will reduce the variance from σ_{xx} to $\sigma_{xx} - \sigma_{fz} {\sigma_{zz}}^{-1} {\sigma_{fz}}^T$.

III. Control Variates for Constrained Variables

The approach described in Section II-D assumes, for example, that $p_x\left(u|d\right) = p_x\left(l|d\right) = 0$. In theory, one could apply the approach if $p_x\left(u|d\right)$ and $p_x\left(u|d\right)$ were non-zero and available as analytic functions. However, this case is rarely encountered. The core contribution of this paper is to consider a mapping to another variable, $y = y\left(x\right)$ (such that $x = y^{-1}\left(y\right)$), which is unconstrained, ie $-\infty \leq y \leq \infty$ and chosen such that there is a mapping from each value of x to a unique value of y. We note that such mappings are used in, for example, the probabilistic programming language, Stan[9], when considering constrained variables[4]. In this context for example:

$$\int_{y=-\infty}^{\infty} p_y\left(y|d\right) \frac{d \log p_y\left(y,d\right)}{dy} dy = 0. \quad (18)$$

[3]We note that the right-hand side of (15) is actually the sum of two terms each having a mean of zero such that we could consider each of the two terms (separately) as control variates. We choose not to do so to conform with the specific approaches used in previous work.

[4]Stan considers a wide range of constraints including: upper and lower bounds; ordered vectors; simplexes; correlation matrices; covariance matrices.

We can then write a similar expression to (16) as follows:

$$\int_{y=\infty}^{\infty} p_y\left(y|d\right) f_y\left(y\right) dy \approx \frac{1}{N}\sum_{i=1}^{N} f_y\left(y^{(i)}\right) + \alpha z_i^y \tag{19}$$

where (similarly to (12)) z_i^y is defined as

$$z_i^y = \left[\begin{array}{c} z_{i,1}^y \cdots z_{i,D}^y, z_{i,(1,1)}^y \cdots z_{i,(1,D)}^y, z_{i,(2,2)}^y \cdots z_{i,(D,D)}^y \end{array}\right]^T \tag{20}$$

with

$$z_{i,j}^y = \frac{d\log\left[p_y\left(y^{(i)}|d\right)\right]}{dy_j} \tag{21}$$

$$z_{i,(j,j)}^y = y_j^{(i)} z_{i,j}^y - 1 \tag{22}$$

$$z_{i,(j,k)}^y = y_k^{(i)} z_{i,j}^y + y_j^{(i)} z_{i,k}^y. \tag{23}$$

We now make the following choices:

$$p_y\left(y|d\right) = p_x\left(y^{-1}\left(y\right)|d\right)\left|\frac{dx\left(y\right)}{dy}\right| \tag{24}$$

$$f_y\left(y\right) = f_x\left(y^{-1}\left(y\right)\right) \tag{25}$$

which are chosen such that:

$$\int_{x=l}^{u} p_x\left(x|d\right) f_x\left(x\right) dx = \int_{y=\infty}^{\infty} p_y\left(y|d\right) f_y\left(y\right) dy \tag{26}$$

$$f_y\left(y\left(x^{(i)}\right)\right) = f_x\left(x^{(i)}\right) \tag{27}$$

and we note that the gradients used in z_i^y are already calculated by some probabilistic programming languages, including Stan[9], when applying the No-U-Turn-Sampler (NUTS)[13].

IV. Results

A. Toy Example

We begin with a simple example involving a Beta distribution with parameters θ_1 and θ_2: $p_x\left(x|d\right) = \beta\left(x;\theta_1,\theta_2\right)$ for $0 \leq x \leq 1$. We specifically consider the case where $\theta_1 = 1$ and $\theta_2 = 5$ (ie d comprises 4 positive outcomes) such that $p\left(0|d\right) = \infty$. We also consider the following mapping:

$$y = \log\frac{x}{1-x} \tag{28}$$

or equivalently

$$x = y^{-1}\left(y\right) = \frac{1}{1+\exp\left(-y\right)}. \tag{29}$$

We know that the mean of a Beta distribution is analytically given by $\frac{\theta_1}{\theta_1+\theta_2}$, which is equal to $\frac{1}{6}$ for the parameter values considered. We consider two estimators, one that considers $\alpha = 0$ and another that considers $\alpha = \alpha^\star$ with linear control variates. We compare the estimation performance of the estimators across 100 Monte-Carlo runs, each of which involves estimating the mean from (just) 10 samples, which we draw from $p_x\left(x|d\right)$. The results from the 100 Monte-Carlo runs are shown in Figure 1. It is evident that, relative to the Monte-Carlo errors provided by a conventional estimator, the Monte-Carlo errors when using control variates are negligible[5].

[5]The mapping chosen is such that y is well approximated as Gaussian, for which linear control variates can reduce the Monte-Carlo variance to zero.

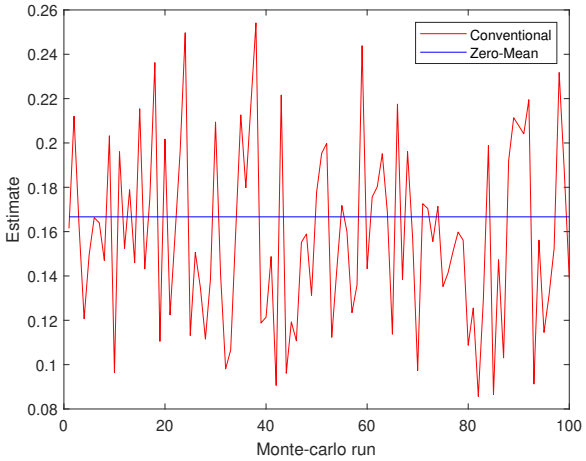

Fig. 1. Results from considering a Beta Distribution.

B. Performance on Stan Benchmarks

To assess the performance of the approach, we considered the 12 'benchmark models' (as used by Stan developers when assessing performance). Table I summarises some pertinent features of these posteriors. We implemented the control variates as a post-processing component that integrates with PyStan2[6]. For each parameter (and 'generated quantity') and for each of linear and quadratic control variates, the software outputs estimates of the: mean; standard deviation; standard error; the effective sample size. Results in this paper were generated using that code and run in Ubuntu 21.04 through Virtualbox, the CPU is i7-7700K with 4 cores enabled.

The "ir_2pl" model has a large number of parameters. So, in the context of this model alone, only D quadratic terms (in (14)) were used and the terms in (15) were omitted.

We then assessed the variance reduction (relative to the variance of an estimator that does not use control variates and in the context of estimating the mean of each parameter, whether it is constrained or not) resulting from using either linear or quadratic control variates. We average performance over all parameters for each model and show the results in Table I along with results for the constrained parameters only.

It is evident that control variates can reduce the variances of estimators in all the cases considered, including those involving constrained parameters, and that using quadratic control variates results in larger reductions in variance than using only linear control variates. Note that for the model "low_dim_gauss_mix_collapse", the advantage of using the two versions of control variates is modest. In contrast, control variates reduce the Monte-Carlo error to within machine precision of zero for the model "low_dim_corr_gauss".

To assess the implicit assertion that calculating control variates is fast, we quantified the computational time spent on each of: building the model; drawing 2000 samples (including time spent in warm-up as well as time spent generating the samples that are output); evaluating the gradients used to

[6]The code (with `readme` and data, d) is available here: https://github.com/zhouyifan233/pystan2 in the branch ControlVariates.

Model	Parameter Count		Variance Reduction				Root Mean-Squared Error		
	All	Constrained	All Parameters		Constrained Parameters		Nones	Linear	Quadratic
			Linear	Quadratic	Linear	Quadratic			
eight_schools	18	1	19.04	72.46	3.04	5.45	0.11	0.06	0.03
gp_pois_regr	24	2	8.82	31.36	1.06	4.00	2.09×10^{-2}	1.44×10^{-2}	1.05×10^{-2}
low_dim_gauss_mix	5	4	6.35×10^{3}	3.25×10^{5}	7.88×10^{3}	3.82×10^{5}	1.46×10^{-3}	1.96×10^{-4}	1.06×10^{-4}
low_dim_corr_gauss	5	0	2.15×10^{31}	4.70×10^{30}	N/A	N/A	0.11	0.09	0.01
low_dim_gauss_mix_collapse	5	3	1.08	1.46	1.11	1.63	0.21	0.21	0.21
arK	7	1	84.79	6.91×10^{3}	24.22	4.95×10^{3}	2.48×10^{-3}	3.43×10^{-4}	1.84×10^{-4}
garch	4	3	11.96	129.85	3.83	17.09	1.25×10^{-2}	6.20×10^{-3}	3.3×10^{-3}
gp_regr	3	3	96.97	332.57	96.97	332.57	2.67×10^{-2}	4.58×10^{-3}	2.23×10^{-3}
sir	84	4	83.41	2.42×10^{4}	58.26	6.43×10^{3}	6.26	1.44	0.32
arma	4	1	36.53	5.87×10^{3}	37.70	6.65×10^{3}	6.78×10^{-2}	6.73×10^{-2}	6.67×10^{-2}
irt_2pl	144	23	7.59	12.84	2.48	5.58	2.51×10^{-2}	1.96×10^{-2}	1.41×10^{-2}
one_comp_mm_elim_abs	44	4	3.59	32.13	3.39	31.86	0.23	0.19	0.10

TABLE I
OVERVIEW OF MODELS, VARIANCE REDUCTION (RELATIVE TO AN MCMC CHAIN THAT RUNS FOR 100 TIMES AS MANY ITERATIONS) FROM USING CONTROL VARIATES (FOR ALL PARAMETERS AND FOR THE CONSTRAINED PARAMETERS ONLY) AND ROOT MEAN-SQUARED ERROR.

Model	Stan		Control variates		
	Build	Sample	Evaluate gradients	Linear	Quadratic
eight_schools	91.80	0.14	9.26×10^{-3}	7.62×10^{-4}	1.66×10^{-3}
gp_pois_regr	94.35	4.76	2.02×10^{-2}	7.13×10^{-4}	2.69×10^{-3}
low_dim_gauss_mix	89.43	4.78	0.26	7.63×10^{-4}	9.30×10^{-4}
low_dim_corr_gauss	90.31	0.07	7.51×10^{-3}	4.77×10^{-4}	6.23×10^{-4}
low_dim_gauss_mix_collapse	89.75	13.16	0.26	5.49×10^{-4}	8.55×10^{-4}
arK	88.52	4.02	6.67×10^{-2}	6.24×10^{-4}	1.01×10^{-3}
garch	88.64	1.19	4.85×10^{-2}	4.56×10^{-4}	7.85×10^{-4}
gp_regr	94.69	0.28	2.14×10^{-2}	4.45×10^{-4}	7.66×10^{-4}
sir	94.02	211.53	1.33	9.73×10^{-3}	1.15×10^{-3}
arma	88.76	2.43	3.18×10^{-2}	4.48×10^{-4}	9.29×10^{-4}
irt_2pl	90.51	11.67	0.25	3.56×10^{-3}	7.28×10^{-3}
one_comp_mm_elim_abs	92.13	48.58	2.21	8.23×10^{-4}	9.66×10^{-4}

TABLE II
THE RUNTIMES (IN SECONDS) OF STAN COMPONENTS AND CONTROL VARIATE CALCULATIONS ON DIFFERENT BENCHMARK MODELS. 1000 SAMPLES WERE DRAWN FOR EACH MODEL.

calculate the control variates[7]; calculating the estimates. These times are shown in Table II. The run-time associated with calculating the control variates is small relative to the time spent generating the samples. Interestingly, while it is slightly more computationally expensive to calculate quadratic rather than linear control variates, the bulk of the time required to calculate the control variates' estimates relates to (re)-evaluation of the gradients. In some cases (e.g. "eight_schools"), this time is commensurate with the time spent generating the samples. We assume this occurs when the tree depth in NUTS is low (so each sample only requires very few gradient calculations).

While, as explained in Section I, control variates can only reduce the standard error and not the standard deviation, it is still of interest to understand the improvement in estimation accuracy that results from using control variates. To quantify this improvement, for each benchmark model, we obtained another set of samples by running Stan for 100 times longer and calculated estimates using these samples (and no control variates). We then calculated the root mean squared error between these estimates and those produced from the smaller set of samples with no control variates as well as with linear and quadratic control variates. 100 Monte-Carlo runs were used to generate all the experiment results. Table I tabulates the results and indicates that using control variates reduces the error. Note that there is a case, the "sir" model, where the use of quadratic control variates appears to increase the error relative to the use of linear control variates. The variance reduction is significantly improved as a result of using quadratic control variates in the context of this model. This implies that the quadratic control variates are providing reductions in variance that are greater than those achieved by using 100 times as many samples (which 'only' reduces the variance by a factor of 10) such that the use of quadratic control variates should, arguably, be the baseline against the other estimators are assessed. We note that these variance reductions are witnessed both in the full set of all parameters and when only the constrained parameters are considered.

[7]These gradients have already been calculated by Stan but are not exposed and so have to be re-evaluated.

V. CONCLUSIONS

By considering a mapping from an unconstrained space to the constrained space, control variates have been developed for constrained parameters. Results indicate that significant reductions in Monte-Carlo error result with negligible additional computational cost. Future work will consider other types of control variates, integrate the code more tightly with Stan, interface the approach into other probabilistic programming languages and apply the work to particle filters (extending an existing application to Sequential Monte Carlo samplers[16]).

REFERENCES

[1] D. Luengo, L. Martino, M. Bugallo, V. Elvira, and S. Särkkä, "A survey of monte carlo methods for parameter estimation," *EURASIP Journal on Advances in Signal Processing*, vol. 2020, no. 1, pp. 1–62, 2020.

[2] A. Doucet and P. Duvaut, "Fully bayesian analysis of hidden markov models," in *1996 8th European Signal Processing Conference (EUSIPCO 1996)*. IEEE, 1996, pp. 1–4.

[3] W. J. Fitzgerald, "Markov chain monte carlo methods with applications to signal processing," *Signal Processing*, vol. 81, no. 1, pp. 3–18, 2001.

[4] C. Vacar, J.-F. Giovannelli, and Y. Berthoumieu, "Bayesian texture and instrument parameter estimation from blurred and noisy images using mcmc," *IEEE Signal Processing Letters*, vol. 21, no. 6, pp. 707–711, 2014.

[5] I. Ram, I. Cohen, and S. Raz, "Multichannel deconvolution of seismic signals using statistical mcmc methods," *IEEE Transactions on signal processing*, vol. 58, no. 5, pp. 2757–2770, 2010.

[6] R.-H. Peng, R.-R. Chen, and B. Farhang-Boroujeny, "Markov chain monte carlo detectors for channels with intersymbol interference," *IEEE transactions on signal processing*, vol. 58, no. 4, pp. 2206–2217, 2009.

[7] S. Flaxman, S. Mishra, A. Gandy, H. J. T. Unwin, T. A. Mellan, H. Coupland, C. Whittaker, H. Zhu, T. Berah, J. W. Eaton *et al.*, "Estimating the effects of non-pharmaceutical interventions on covid-19 in europe," *Nature*, vol. 584, no. 7820, pp. 257–261, 2020.

[8] B. Eraker, "Mcmc analysis of diffusion models with application to finance," *Journal of Business & Economic Statistics*, vol. 19, no. 2, pp. 177–191, 2001.

[9] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *Journal of statistical software*, vol. 76, no. 1, 2017.

[10] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck, "Probabilistic programming in python using pymc3," *PeerJ Computer Science*, vol. 2, p. e55, 2016.

[11] A. Pfeffer, "Figaro: An object-oriented probabilistic programming language," *Charles River Analytics Technical Report*, vol. 137, p. 96, 2009.

[12] H. Ge, K. Xu, and Z. Ghahramani, "Turing: a language for flexible probabilistic inference," in *International conference on artificial intelligence and statistics*. PMLR, 2018, pp. 1682–1690.

[13] M. D. Hoffman, A. Gelman *et al.*, "The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo." *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1593–1623, 2014.

[14] R. Assaraf and M. Caffarel, "Zero-variance principle for monte carlo algorithms," *Physical review letters*, vol. 83, no. 23, p. 4682, 1999.

[15] A. Mira, R. Solgi, and D. Imparato, "Zero variance markov chain monte carlo for bayesian estimators," *Statistics and Computing*, vol. 23, no. 5, pp. 653–662, 2013.

[16] L. South, C. Oates, A. Mira, and C. Drovandi, "Regularized zero-variance control variates," *Bayesian Analysis*, vol. 1, no. 1, pp. 1–24, 2022.

[17] N. Friel, A. Mira, and C. J. Oates, "Exploiting multi-core architectures for reduced-variance estimation with intractable likelihoods," *Bayesian Analysis*, vol. 11, no. 1, pp. 215–245, 2016.

[18] T. Papamarkou, A. Mira, and M. Girolami, "Zero variance differential geometric markov chain monte carlo algorithms," *Bayesian Analysis*, vol. 9, no. 1, pp. 97–128, 2014.

[19] C. Oates and M. Girolami, "Control functionals for quasi-monte carlo integration," in *Artificial Intelligence and Statistics*. PMLR, 2016, pp. 56–65.