Preprint ANL/MCS-P1539-0908

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2009; **33**:1–34 *Prepared using* nmeauth.cls *[Version: 2002/09/18 v2.02]*

Efficient sampling for spatial uncertainty quantification in multibody system dynamics applications

Kyle Schmitt [1] Mihai Anitescu [2,*] and Dan Negrut [1]

[1] *Department of Mechanical Engineering, University of Wisconsin, Madison, WI 53562* [2] *Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439* anitescu@mcs.anl.gov

SUMMARY

We present two methods for efficiently sampling the response (trajectory space) of multibody systems operating under spatial uncertainty, when the latter is assumed to be representable with Gaussian processes. In this case, the dynamics (time evolution) of the multibody systems depends on spatially indexed uncertain parameters that span infinite dimensional spaces. This places a heavy computational burden on existing methodologies, an issue addressed herein with two new conditional sampling approaches. When a single instance of the uncertainty is needed in the entire domain, we use a Fast Fourier Transform technique. When the initial conditions are fixed and the path distribution of the dynamical system is relatively narrow, we use an incremental sampling approach that is fast and has a small memory footprint. Both methods produce the same distributions as the widely used Cholesky-based approaches. We illustrate this convergence at a smaller computational effort and memory cost for a simple nonlinear vehicle model. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: uncertainty quantification, Gaussian process, dynamical system, fast Fourier transform, conditional sampling

1. INTRODUCTION

Uncertainty quantification is relevant in a broad spectrum of science and engineering problems. It has been studied for decades with a wide spectrum of methods and techniques that are gradually improving in accuracy, efficiency, and robustness. This paper focuses on the efficiency and robustness attributes of algorithms used to investigate the dynamics of multibody system operating under spatial uncertainty. The equations that govern the time evolution of a multibody system can be expressed in the form (see, for instance, [1])

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{L}(\mathbf{q})\mathbf{v} \\ \mathbf{M}(\mathbf{q})\dot{\mathbf{v}} &= \mathbf{f}^A\left(t, \mathbf{q}, \mathbf{v}\right), \end{aligned} \tag{1}$$

*Correspondence to: Mathematics and Computer Science Div., Argonne National Laboratory, Argonne, IL 60439 anitescu@mcs.anl.gov

Received Sep 11, 2008
Revised Feb 12, 2009
Accepted Apr 21, 2009

where $\mathbf{q} = \left[\mathbf{r}_1^T, \epsilon_1^T, \ldots, \mathbf{r}_{n_b}^T, \epsilon_{n_b}^T\right]^T \in \mathbb{R}^{6n_b}$ are generalized positions, $\mathbf{v} = \left[\dot{\mathbf{r}}_1^T, \bar{\omega}_1^T, \ldots, \dot{\mathbf{r}}_{n_b}^T, \bar{\omega}_{n_b}^T\right]^T \in \mathbb{R}^{6n_b}$ are generalized velocities, and n_b represents the number of bodies in the system. The matrix $\mathbf{M}(\mathbf{q})$ is the generalized mass matrix, and $\mathbf{f}^A(t, \mathbf{q}, \mathbf{v})$ represents the vector of generalized applied forces. Often times, additional nonlinear algebraic constraints $\mathbf{\Phi}(t, \mathbf{q}) = \mathbf{0}$ must be satisfied by the generalized coordinates $\mathbf{q}$ leading to a set of differential algebraic equations (see, for instance, [2]). For each body i, its orientation is then described by a set of three Euler angles, $\epsilon_i \in \mathbb{R}^3$, following the 3-1-3 local rotation sequence (see, for instance, [1]). The rate at which each body changes its orientation is captured by the local angular velocity $\bar{\omega}_i \in \mathbb{R}^3$. The location of each body is uniquely determined by a position vector $\mathbf{r}_i = [x_i, y_i, z_i]^T$ that specifies where the body-fixed centroidal reference frame is located. The translational velocity of the body is simply $\dot{\mathbf{r}}_i$, where an overdot represents time differentiation. Since for each body i there is a locally nonsingular matrix $\mathbf{B}(\epsilon_i)$ such that $\bar{\omega}_i = \mathbf{B}(\epsilon_i)\dot{\epsilon}_i$, the operator $\mathbf{L}(\mathbf{q})$ that relates the time derivative of the level-zero generalized coordinates to the level-one generalized coordinates is generally not the identity matrix. Although many sets of generalized coordinates can be used to capture the dynamics of a multibody system, note that the equations of motion in (1) draw on the so-called absolute, or Cartesian, representation of the attitude of each rigid body in the system. Also note that for brevity, this summary of the multibody dynamics problem focused on the rigid body case. Equation (1) also captures the dynamics associated with multibody systems comprising flexible bodies, with the caveat that the set of generalized coordinates is richer to capture the deformation modes associated with each flexible body (see, for instance, [3]).

Ground and air vehicles are two of the most common examples of systems leading to a multibody dynamics problem. For instance, the uncertainty in the response of a ground vehicle might stem from a friction coefficient that is insufficiently known at the tire/road interface, or might be the result of uncertainty in the road profile (elevation). In other words, the uncertainty permeates the equations of motion through the force term $\mathbf{f}^A(t, \mathbf{q}, \mathbf{v})$. The force $\mathbf{f}^A$ depends on the location $\mathbf{q}$, and at this location uncertainty enters the problems through an uncertain friction coefficient, road profile, etc.

In an abstract framework, we are interested in characterizing the spatial uncertainty effects on a dynamical system

$$\dot{x} = f(x, t, \eta(x)), \tag{2}$$

subject to a given set of initial conditions. Note that an initial value problem that draws on (1) can always be recast as in (2). The function $f(x, t, u)$ is the intrinsic function that dictates the dynamics of the system; it is a known function obtained as the outcome of a mathematical modeling stage that is not of interest here. The quantity $\eta(x)$ is a stationary random variable indexed by the space variable x. The following problem is of interest:

$$\begin{array}{c}\text{Characterize the distribution of the trajectory (random variable) } x(t) \text{ at} \\ \text{a given time } t \text{ or for a collection of times } t_1 < t_2 < \ldots < t_N.\end{array} \tag{3}$$

If (2) represents the equations of motion of a vehicle, $\eta(x)$ can be the uncertain road elevation or friction coefficient. The same formalism can be applied to differential algebraic equations with virtually no change. For simplicity, the presentation is restricted to the case of ordinary differential equations (2), which in a stochastic framework is more rigorously reformulated as

$$\dot{x} = f(x, t, \eta(x, \omega)), \tag{4}$$

where ω is an element of the event space Ω. The solution of (4) leads to a trajectory $x(t, \omega)$, which is thus a random variable. An example of a quantity of interest is $E_\omega [x(T, \omega)]$, the expected state at a time T. The key to efficiently solving the stated problem is to sample $\eta(x, \omega)$ efficiently, that is, produce a sample function $\eta(x, \omega_1)$ for a given event $\omega_1 \in \Omega$ and for any x. To simplify the notation, we state merely that $\eta(x)$ and $x(t)$ are random variables, and we do not explicitly denote their dependence on ω. Note that (2) is not a stochastic differential equation; rather, it is an ordinary differential equation with state-dependent uncertain parameters.

An overview of uncertainty quantification methods commonly used can be found in [4]. Probably the most widespread technique draws on random sampling methods, such as Monte Carlo and Latin hypercube sampling [5, 6]. Reliability methods can be computationally less intensive than sampling methods [7]. Their increased efficiency is the result of the fact that the entire surface response is updated by any given sample information. This update can be carried out by mean value [7], global Gaussian process [8, 9], and other approaches [10]. Polynomial chaos expansion (PCE) methods construct a polynomial surface response approximation [11, 12, 13]. They use either a nonlinear projection Galerkin method [11] or a collocation approach [13] to obtain the polynomial approximation. PCE methods have generally been impaired, however, by a relatively low dimension of the uncertainty space that they can span [13]. The other methods are in principle applicable independent of dimension. Nonetheless, their efficiency tends to degrade with the increasing dimension of uncertainty spaces if one uses the implementations presented in the above references.

Kriging and its variations, co-kriging and kriging with regression, are example of the developing nature of spatial uncertainty [14, 15, 16]. These "hybrid methods" were introduced to address accuracy and robustness concerns, capable of extending to less-structured data (i.e. nonstationary, nonhomogeneous). However, the methods require the solution of small kriging systems at each sample point or the solution of a large system, using global dual kriging simulation. These methods are suitable for data interpolation of small sample sets from large data sets but cannot be extrapolated without encountering significant computational effort. Gaussian processes, as presented in [17] and applied in [18], provide an accurate and versatile methodology for interpolating spatial data. Unfortunately, the Gaussian processes framework traditionally suffers from numerical instability and excessive storage requirements because of the matrix operations it employs.

Some traditional methods for spatial uncertainty calculations, such as white-noise methods and spectral methods, fare much better than kriging with respect to runtime yet suffer from accuracy and robustness issues. Examples of the former, based on homogeneous random processes, are explained in [19, 20]. While these methods model a large class of problems and may be useful in design and simulation, they are nonetheless not appropriate for situations where the spatial variation has large areas of coherence that are inhomogeneous, a common occurrence in real-life applications. Another original approach is discussed in [21], where the use of traditional spectral methods is supplemented with covariance spectrum phase values conditioned from data. However, the methodology is plagued by accuracy limitations typically associated with spectral methods. The accuracy of these methods is dependent on the number of harmonics used and can quickly become hard to implement and manage. Likewise, spectral techniques cannot easily accommodate rapid variations in the properties of the surface, which is a well-known side effect of the Gibbs phenomenon, thus presenting robustness concerns.

Growing problem complexity and solution resolution have led to creative approaches for conditioned problems, that is, solution methods that gain efficiency in lieu of robustness,

and sometimes accuracy. These methods still provide informative results that can be used in industry applications. One common approach is regimenting the data space to a lattice and sampling from the prior. In this case, the samples are not conditioned on the local deviation from the prior model. The benefit of this simplification is that the covariance matrix is Toeplitz and can be sampled from by leveraging fast Fourier transforms (FFT) [22]. Another perspective gaining wider acceptance is the sparse-grid assumption introduced by Stein in [14]. Stein leverages compact support kernels to limit the training data set size required for statistically accurate and consistent interpolation. This is especially useful in conjunction with high-frequency data sets, where correlation decays rapidly to nearly zero. Stein, however, does not couple sparse grid theory with conditional sampling to allow for truly dynamic interpolation, as we do in Section 4.

The first method proposed herein, periodic fold sampling (PFS), imposes one condition and one assumption in order to conduct the entire Gaussian process based methodology in a *matrix-free* fashion. The condition is that the data is provided on a lattice. The assumption is that the space is periodic, which reflects in the autocorrelation function. This assumption was tried and found to introduce few problems away from boundaries when working on large grid spaces. We present the specifics of each PFS step conducted in the frequency domain: data organization and data augmentation, estimation of hyper-parameters, computation of posterior distribution characteristics, and multivariate Gaussian sampling. We explain how FFTs are used to map between the time and frequency domains. We note that FFT approaches have been used to sample from prior distributions [23] but not, as far as we know, from the posterior distribution. We show in conjunction with PFS the convergence of the resulting sample mean and covariance to the respective values computed with traditional Gaussian processes, as in [18].

The second method proposed, the incremental Gaussian process approach (IGPA), works on subdomains to localize the data interpolation. Data does not need to be interpolated all at once on the entire problem space but rather can be interpolated incrementally only in the proximity of the region of importance. In this approach the computational effort is invariant to problem size. For IGPA, we describe the incremental approach justification in detail, including a discussion of the implementation process, formal proofs, and numerical results.

1.1. Application Example

To illustrate the performance of the proposed methods, they will be used to quantify the uncertainty in the behavior of a simple vehicle model running over a flat road that has a given distribution of the tire/road friction coefficient at a set of learning points provided on a two dimensional grid. The specifics of this simulation are detailed in [18], but the essentials are discussed here. First, a model for the friction coefficient distribution is required; studies in geostatistics suggest that the squared exponential is a representative correlation function for Gaussian random processes [23] that is suitable to be used. In order to address the natural bounds of ice (between dry friction μ_d and ice friction μ_s), a phase parameter ϕ is used:

$$\phi = -\ln\left(\frac{\mu_d - \mu}{\mu - \mu_s}\right) \in (-\infty, \infty). \tag{5}$$

Note that the phase parameter is an infinite-dimensional random variable indexed by a spatial variable. Next, given a set of points $x_1, x_2, \ldots x_M$ in space, denote by $\phi_1, \phi_2, \ldots, \phi_M$

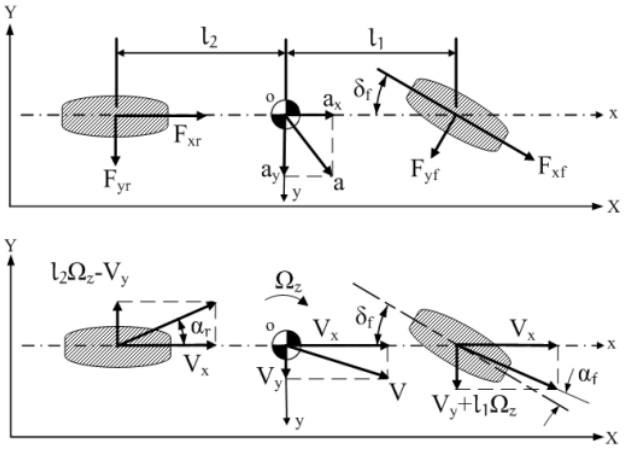


Figure 1. Bicycle model used in numerical validation [24].

the random variables that represent the phase parameters at those locations. Modeling the phase parameter as a Gaussian process means that the vector $\Phi = (\phi_1, \phi_2, \ldots, \phi_M)$ has both a prior and a posterior multivariate normal distribution. The posterior distribution can be determined from well-established statistical considerations applicable for any multivariate normal distribution. The key attribute of Gaussian random processes is that the prior distribution can be generated from a bivariate function, the covariance function, and a univariate function, the mean function. Herein we consider an exponential covariance function with hyperparameters $\theta_2 = \{\gamma, \alpha_{x1}, \alpha_{x2}\}$, and a linear mean function with hyperparameters $\theta_1 = \{a_0, a_1, a_2\}$ [17]:

$$m(x;\theta_1) = a_0 + a_1 x_1 + a_2 x_2, \tag{6}$$

$$k(x, x';\theta_2) = \exp\left(- \left[\frac{(x_1 - x_1')}{\alpha_{x_1}} \right]^{2/\gamma} - \left[\frac{(x_2 - x_2')}{\alpha_{x_2}} \right]^{2/\gamma} \right). \tag{7}$$

The vehicle model used, shown in Figure 1, has an open-loop steering system set to execute a constant radius turn without any axial forces applied. The model has three degrees of freedom: those associated with the longitudinal motion V_x, lateral motion V_y, and yaw Ω_z. Three input functions determine the behavior of the model: steer angle δ_f and the front and rear wheel road adhesion coefficients μ_f and μ_r, respectively. The governing equations are as follows:

$$\begin{aligned} m(\dot{V}_x - V_y\Omega_z) &= -F_{yf}\sin\delta_f \\ m(\dot{V}_y + V_x\Omega_z) &= F_{yr} + F_{yf}\cos\delta_f \\ I_z\dot{\Omega}_z &= l_1 F_{yf}\cos\delta_f - l_2 F_{yr} \\ \dot{X} &= V_x\cos\Theta_z - V_y\sin\Theta_z \\ \dot{Y} &= V_x\sin\Theta_z + V_y\cos\Theta_z \\ \dot{\Theta}_z &= \Omega_z. \end{aligned} \tag{8}$$

The constitutive equations for the forces acting on the tires, F_{yr} and F_{yf}, are of form

$F = \psi(x, d_f, \phi, \nu)$ where $x = (X, Y, \Theta_z, V_x, V_y, \Omega_z)$, d_f is the steering angle input, ϕ is the uncertain parameter in this study and ν contains tire/vehicle material and geometry parameters [24]. Note that after dividing with the inertial terms m, I and solving for $\dot{V}_x, \dot{V}_y$, and $\dot{\Omega}_z$ in (8) the problem is cast in the standard form defined by (2). The function f is determined from the function ψ above that defines the tire forces F. The stochastic parameter η in (2) represents the friction phase parameter ϕ.

2. GAUSSIAN PROCESS DESCRIPTION OF SPATIAL UNCERTAINTY

Gaussian processes represent a versatile approach for simulating infinite-dimensional uncertainty. By definition, a spatially distributed random variable $\eta(x)$ is a Gaussian process with mean function $m(x;\theta_1)$ and correlation function $k(x, x';\theta_2)$ if, for any set of space points $X = \{x_1, x_2, \ldots, x_M\}$

$$\eta(X) = \begin{pmatrix} \eta(x_1) \\ \eta(x_2) \\ \vdots \\ \eta(x_M) \end{pmatrix} \sim \mathcal{N}\left(m(X;\theta_1), K(X, X;\theta_2)\right). \tag{9}$$

For $m \in \mathbb{R}^M$ and $K \in \mathbb{R}^{M\times M}$, $\mathcal{N}(m, K)$ in (9) is the M–variate normal distribution with mean m and variance K, with

$$m(X;\theta_1) = \begin{pmatrix} m(x_1,\theta_1) \\ m(x_2,\theta_1) \\ \vdots \\ m(x_M,\theta_1) \end{pmatrix}, \quad K(X, X';\theta_2) = \begin{pmatrix} k(x_1,x'_1;\theta_2) & k(x_1,x'_2;\theta_2) & \cdots & k(x_1,x'_N;\theta_2) \\ k(x_2,x'_1;\theta_2) & k(x_2,x'_2;\theta_2) & \cdots & k(x_2,x'_N;\theta_2) \\ \vdots & \vdots & \vdots & \vdots \\ k(x_M,x'_1;\theta_2) & k(x_M,x'_2;\theta_2) & \cdots & k(x_M,x'_N;\theta_2) \end{pmatrix},$$

where $X' = \{x'_1, x'_2, \ldots, x'_N\}$, and θ_1 and θ_2 are the hyperparameters of the mean and covariance functions.

As in reference [17], we employ a Bayesian point of view in dealing with uncertainty. The hyperparameters θ_1 and θ_2 are obtained from a data set $\eta(D)$ at nodes $D = \{d_1, d_2, \ldots, d_M\}$. The posterior distribution of the variable $\eta(S)$ at node points $S = \{S_1, S_2, \ldots, S_N\}$, consistent with $\eta(D)$, is $\mathcal{N}(f^*, K^*)$ [17], where

$$f^* = K(S, D;\theta_2)\left[K(D, D;\theta_2) + \sigma_N^2 I_M\right]^{-1}\left(\eta(D) - m(D;\theta_1)\right) + m(S;\theta_1) \tag{10}$$

$$K^* = K(S, S;\theta_2) - K(S, D;\theta_2)\left[K(D, D;\theta_2) + \sigma_N^2 I_M\right]^{-1} K(D, S;\theta_2). \tag{11}$$

Here I_M represents the identity matrix of dimension M, and we have included the modification to the posterior distribution that is brought about by the noise in the data with variance σ_N^2.

The key issues in simulating from this posterior model are (a) how to obtain the hyperparameters from data and, (b) how to sample from $\mathcal{N}(f^*, K^*)$, especially in the case where M is very large. The classical approach is to do a Cholesky factorization of K^*, a costly order $O(M^3)$ operation.

To simplify the notation, we will use the same symbols for a vector of random variables as for the vector of locations to which these variables are attached: for example, $\eta(D) \to D$, and $\eta(S) \to S$, the use of which will become clear from the context. In addition, we will not explicitly represent the dependence of $m(\cdot)$ and $k(\cdot,\cdot)$ on the hyperparameters θ.

2.1. Hyperparameter Estimation

Before training the posterior and sampling from it, the hyperparameters must be estimated to evaluate the prior. A commonly used method for the estimation of the hyperparameters from data is maximum likelihood estimation [17]. The method relies on the maximization of the log-likelihood function. In the multivariate Gaussian with mean $m(\theta)$ and covariance matrix $K(\theta)$ case, the log-likelihood function assumes the form

$$\log p(y|\theta) = -\frac{1}{2}W^T K(\theta)^{-1} W - \frac{1}{2}\log|K(\theta)| - \frac{M}{2}\log 2\pi \tag{12}$$

where $W = y - m(\theta)$ and y is the observed data. Here $|A|$ denotes the determinant of a matrix A. In the case of spatial uncertainty, the dependence on the hyperparameters θ appears by means of the spatial coordinates x.

In the example in Section 1.1, we have that $\theta = \{\theta_1, \theta_2, \sigma_n\}$. The gradients of the likelihood function can be computed analytically [17]:

$$\frac{\partial}{\partial\theta_{1j}}\log p(y|\theta) = \frac{1}{2}\mathrm{tr}\left((K(\theta)^{-1}W(K(\theta)^{-1}W)^T - K(\theta)^{-1})\frac{\partial K(\theta)}{\partial\theta_{1j}}\right) \tag{13a}$$

$$\frac{\partial}{\partial\theta_{2j}}\log p(y|\theta) = -\left(\frac{\partial}{\partial\theta_{2j}}m(\theta)\right)^T K(\theta)^{-1}W \tag{13b}$$

$$\frac{\partial}{\partial\sigma_n}\log p(y|\theta) = \frac{1}{2}\sigma_n\mathrm{tr}\left(K(\theta)^{-1}W(K(\theta)^{-1}W)^T - K(\theta)^{-1}\right). \tag{13c}$$

Here, $\mathrm{tr}(A)$ denotes the trace of a square matrix A, the sum of its diagonal entries.

It can be proven that the gradient of $K(\theta)$ with respect to any hyperparameter maintains a periodic matrix structure. This fact suggests that all evaluation inside the trace operator can be done in the frequency domain after K and $\frac{\partial K}{\partial\theta_{1j}}$ are diagonalized with FFT. Nonetheless, since the conditional sampling part of our procedure is far more time consuming, we use the FFT-based direct diagonalization only for its matrices.

To determine the hyperparameters, we use the MATLAB *fsolve* function. This function implements a quasi-Newton approach for nonlinear equations. We apply it to the nonlinear equations (13) obtained from the optimality conditions of maximizing the likelihood function. Using an optimization approach instead would have resulted in the need to compute the objective function at multiple points. This involves the evaluation of the determinant $|K(\theta)|$, which is currently not known to be possible in a matrix-free setting. Note that (13) can be evaluated in matrix-free fashion by using conjugate gradient techniques for linear systems that are equivalent to the application of K^{-1} by using an FFT approach. Although a matrix-free approach for MLE has not been implemented in this paper, this is a major roadblock for using optimization approaches for large problems of the type we intend to solve in the future. We have thus decided to carry out the MLE procedure by a nonlinear equations approach in order to test its feasibility.

2.2. Conditional Probability Decomposition

The two methodologies presented in this paper hinge on conditional sampling with Gaussian processes; that is, both data and previous samples are used when training new data sets. In statistical notation, this is captured by the following condition:

$$\mathbb{P}(S_1, S_2|D) = \mathbb{P}(S_2|S_1, D) * \mathbb{P}(S_1|D), \tag{14}$$

where D is the observed data space, S_1 is the data space for the first iteration, and S_2 is the data space for the second iteration. The motivation behind this "divide and conquer" approach is twofold. First, the aggregate of all conditional simulations is sometimes more efficient to carry out than one comprehensive simulation. Second, the memory requirements of a successive conditioning approach can be much reduced compared to the all-at-once approach. Note that the conditioning procedure can be applied recursively to result in as small a dimension of the variable to be sampled as needed, provided that the parameters of the conditional distribution can be efficiently determined.

The equality (14) can be expressed with probability densities by using the multivariate Gaussian distribution joint probability density function [17]:

$$p\left(\begin{pmatrix} S_1 \\ S_2 \end{pmatrix} |m_1, \Sigma_1\right) = p(S_2|m_2, \Sigma_2) * p(S_1|m_3, \Sigma_3) \tag{15}$$

$$p(x|m, \Sigma) = (2 * \pi)^{-n/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(x - m)^T \Sigma^{-1}(x - m)\right), \tag{16}$$

where $x \in \mathbb{R}^n$. The aggregate covariance matrix Σ_1 and the conditional covariance matrices Σ_2 and Σ_3 are defined in equation (34) of the Appendix. This result will be taken advantage of in PFS and IGPA.

3. A SPECTRAL APPROACH: PERIODIC FOLD SAMPLING

The question that motivates PFS is as follows: how can one conduct high-fidelity interpolation and avoid the bottleneck of large matrix storage and computation without compromising accuracy and robustness? A satisfactory answer is produced if two assumptions regarding (a) the spatial distribution of the training input data, and (b) input data periodicity are made. Sampling can then be carried out efficiently following a regimented strategy called *folding*.

In terms of (a) above, PFS is applied strictly to learning data initially provided on a lattice. This assumption certainly reduces the applicability of the methodology. Still, countless applications exist where data is compiled this way. Small-scale data collection in custom lab experiments, for instance, is one area where lattice data collection is readily achievable. Another important application of PFS is in satellite-collected data because the range of satellites makes their data collection flexible and adaptive. Note that although the paper focuses on the two-dimensional methodology, little alteration is required to extend the methodology to higher dimensional spaces.

Assumption (b) above reflects in the expression of the covariance function. Unlike in (7), the covariance function is selected to be periodic:

$$k_p(x, x^*) = \begin{cases} \prod_{s=1}^{l} \exp\left(-\left[\frac{|x_s - x_s^*|}{\alpha_s}\right]^{2/\gamma}\right) & |x_s - x_s^*| < L_s/2 \\ \prod_{s=1}^{l} \exp\left(-\left[\frac{L_s - |x_s - x_s^*|}{\alpha_s}\right]^{2/\gamma}\right) & |x_s - x_s^*| > L_s/2, \end{cases} \tag{17}$$

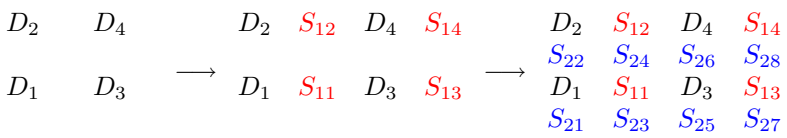


Figure 2. Pictorial representation of first two iterations of PFS.

where l is the number of dimensions, x is a point in the first data set, x^* is a point in the second data set, L_s is the dimension respective domain length, and α_s is the dimension respective characteristic length-scale. Because this class of periodic matrices is an algebra, the periodic assumption ensures that the Gaussian processes-computed posterior covariance matrix in (11) will also be periodic. This periodic nature will allow us to diagonalize the matrix with FFT. It should be pointed out, however, that any Gaussian process can be approached in a periodic framework on a domain that is extended twice in each direction [22]. The approach proposed and based on (17) does not need a larger domain, but runs the risk of resulting in non-positive definite covariance matrices for fine grids [14]. We have not run into this scenario for any of the grid sizes that we have considered. In the end, the salient feature of PFS is that it can be applied in both cases; i.e., when condition (b) really holds, or when one forces this assumption and accepts the risk of corrupting the accuracy of the results, a risk which decreases exponentially fast away from the domain boundary.

The periodic assumption has been used in a broad range of research and applications fields to achieve significant efficiency gains. The aspect that adversely impacts this approach is the Gibbs phenomenon: the correlative misrepresentation of sample points near the boundary, especially those within two characteristic length-scales of the boundary. Points near the boundary are predicted to have large correlation with boundary points on opposite sides of the space, which is certainly false unless the space is innately periodic (spherical or cylindrical, for instance). In natural distributions, this drawback is critical only if the inspected space is on the order of the characteristic length-scale, which is rarely the case. In this situation, less sophisticated regression methods can be used to predict data. It is important to note the erroneous nature of our method near the boundaries, however, so that information is drawn only from accurate samples and simulations are conducted away from the boundaries.

The regimented sampling strategy required by PFS, introduced above as *folding*, is done iteratively. Each iteration folds or shifts the sample points from the existing data points by half a resolution length; thus, each iteration doubles the sampled space. Subsequent iterations then use the previously interpolated data to learn from. This strategy is illustrated in Figure 2. Figure 3 shows in detail the direction and magnitude of each consecutive shift; each time the direction of shift is rotated by $-\pi/2$, and the shift magnitude is reduced 50 percent. The conditional nature of sampling was discussed in detail in Section 2.2. The fold method of sampling ensures that the K_{12} covariance matrix will be square and FFT diagonalizable as proven and leveraged in Section 3.1.

Formally, a series of increasing grids of data, Γ_D, Γ_1, Γ_2, ..., Γ_n, is considered. The data vector D is provided on Γ_D, and obtaining sample vectors S_1 on Γ_1, S_2 on Γ_2, ..., S_n on Γ_n is becomes the problem of interest. The approach proposed relies on an iterative use of the

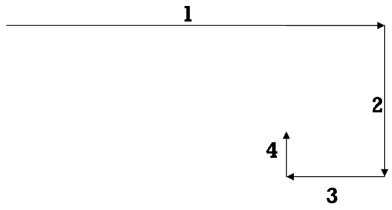


Figure 3. Direction and magnitude of the iterative shift.

conditional probability density formula (14):

$$p\left(S_1, S_2, \ldots, S_n|D\right) = \prod_{i=1}^{n} p\left(S_i|D, S_1, S_2, \ldots S_{i-1}\right). \tag{18}$$

Due to the spiral-like spatial dependence described in Figure 3, a Markov chain-like simplification of $p((S_i|D, S_1, S_2, \ldots S_{i-1}))$ to $p((S_i|D, S_{i-1}))$ or other variants is generally not possible, except for very particular cases of processes, such as the white noise process. The product rule allows for incremental sampling. Starting from D, sampling occurs for sets S_i of increasing index i conditionally on the preceding sample values. A key observation is that, for Gaussian processes, the conditional distribution is algebraically easy to obtain based on equation (34) in the Appendix.

We note that

$$\dim(S_i) = \dim(D) + \sum_{j=1}^{i-1} \dim(S_j), \quad i = 1, 2, \ldots, n, \tag{19}$$

which results in the vectors to the left and right of the conditional symbol having the same size. Consequently, a conditional sample can be easily calculated by using FFT techniques. It is quite possible that the FFT approach can work in some form even when (19) does not hold, but this remains to be investigated. Finally, note that this sampling approach is not suitable for computing realizations at ad hoc locations because only coordinates $(N_1 s_x/2^n, N_2 s_y/2^n)$ are interpolated, where n is the number of fold iterations, s_x and s_y are the respective dimension observed data resolutions, and N_1 and N_2 are positive integers.

3.1. Matrix-Free Computation of Posterior Distribution Characteristics

Designing an efficient method for sampling the conditional distribution $p\left(S_i|D, S_1, S_2, \ldots, S_{i-1}\right)$ requires an effective recursive conditioning method in (18), which becomes the objective of this section. Herein, subscript 1 refers to the coordinates of the observed and simulated data set, $\Gamma_1 = \{D, S_1, S_2, \ldots, S_{i-1}\}$, and subscript 2 refers to the half-resolution shift of those coordinates, $\Gamma_2 = S_i$. With this block notation, the prior distribution is

$$(\Gamma_1, \Gamma_2) \sim p(D, S_1, S_2, \ldots, S_i) \sim \mathcal{N}\left(\begin{pmatrix} m_1 \\ m_2 \end{pmatrix}, \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_W \end{bmatrix}\right).$$

Reference [17] presents a time-domain framework for computing the Gaussian posterior distribution characteristics from the observed data y incorporating measurement noise σ_n,

using the prior mean, (6), and the prior covariance matrix, (7). This results in $p(\Gamma_2|\Gamma_1) \sim \mathcal{N}(f^*, K^*)$, where

$$f^* = m_2 + K_{21}{K_W}^{-1}(y - m_1) \in \mathbb{R}^N \tag{20a}$$
$$K_W = K_{11} + \sigma_n^2 I_D \in \mathbb{R}^{M\times M} \tag{20b}$$
$$K^* = K_{22} - K_{21}{K_W}^{-1}K_{12} \in \mathbb{R}^{N\times N}. \tag{20c}$$

Here M is the number of observed and currently simulated data, and N is the dimension of data vector to be sampled. Note that, in this notation, I_D is the identity operator when reduced to the D subblock of Γ_1, and it is 0 otherwise; it is not the identity matrix of dimension M. The matrix blocks are obtained from the covariance function as $K_{11} = k(\Gamma_1, \Gamma_1)$, $K_{12} = k(\Gamma_1, \Gamma_2)$ and $K_{22} = k(\Gamma_2, \Gamma_2)$. Note that (20) is a condensed version of (10) and (11), where the subscripts qualify which variate is on which covariance matrix axis.

The operations involved above, matrix inverses and matrix products, lead to excessive runtimes. Furthermore, storage capabilities begin to falter when the number of sample points N is in the neighborhood of 10^4 because of the N^2 entries in the covariance matrices.

Our method emulates the framework outlined above, maintaining its robustness and accuracy, but draws on two assumptions introduced in Section 3 to conduct all expensive operations in the frequency domain. It is based on the key observation that all covariance matrices, K_{11}, K_{12}, and K_{22} (herein called the shift covariance matrix), are FFT diagonalizable; that is, $K_a = Q * D_a * Q'$, where D_a is a diagonal matrix and a is one of the pairs 11, 12, 22, and W. Q is a unitary matrix such that $Q = \mathcal{F}(I_N)$, where $\mathcal{F}$ is FFT over the space bearing the uncertainty. In a two-dimensional space, FFT is defined as (using the array indexing from 0)

$$\{x_{lq}\} \xrightarrow{\mathcal{F}} \left\{\hat{x}_{l_f q_f}\right\}; \quad \hat{x}_{l_f q_f} \triangleq \frac{1}{N^2}\left(\sum_{l,q=0}^{N-1} x_{lq} e^{-i\left(\frac{2\pi}{N} l l_f + \frac{2\pi}{N} q q_f\right)}\right), \ \forall l_f, q_f = 0, 1, \ldots, N-1. \tag{21}$$

In the following, we denote by D the set of either data grid points or data random variables and by D_a a diagonal matrix identified by its subscript a. Note that N need not be the same in both the l and q variable, but the proofs are carried out under this assumption for algebraic convenience.

The matrices K_{11}, K_{12}, and K_{22} are periodic matrices. If the spatial dimension is one, they are circulant matrices. In two dimensions, they satisfy the following property, described by a generic matrix K, indexed by the grid index pairs

$$K_{lq,l'q'} = u\left((l - l')\mathrm{mod}N, (q - q')\mathrm{mod}N\right), \quad l, l', q, q' = 0, 1, \ldots, N-1. \tag{22}$$

Here u is a function defined on $\{0, 1, \ldots, N-1\} \times \{0, 1, \ldots, N-1\}$. In particular, for $l, q = 0, 1, \ldots, N-1$, K_{11}, K_{12}, and K_{22} satisfy (22) for

$$u_{11}(l, q) = u_{22}(l, q) = k\left(\frac{lL_x}{N}, \frac{qL_y}{N}\right), \ \ u_{12}(l, q) = k\left(\frac{(l + 0.5)L_x}{N}, \frac{qL_y}{N}\right).$$

Recall that we take the covariance function k to be periodic over a rectangle of dimension $L_x \times L_y$. We consider only an x-shift, but the y-shift conclusions follow as well by the ensuing argument. The matrix K_W can also be represented as in (22) using

$$u_W(l, q) = u_{11}(l, q) + \sigma_n^2 \delta((l)\mathrm{mod}N_D, (q)\mathrm{mod}N_D), \quad l, q = 0, 1, \ldots, N-1,$$

where $\delta(l,q) = 1$ if $l = 0, q = 0$, and it is 0 otherwise.
It remains to prove that matrices with the structure described by (22) can be diagonalized by FFT. While this result is straightforward for covariance matrices such as K_{11} and K_{22} [23], we are not aware of a similar proof in the Gaussian process literature based on the representation (22). The latter is central to PFS since it shows that K_{12} is also diagonalizable by FFT.

The proof proceeds by considering next the FFT basis vectors; i.e., the FFT transform of the columns of the identity matrix. They are, up to a scaling parameter,

$$V^{l_f,q_f} = \left\{ v_{l,q}^{l_f q_f} \right\}_{l,q}; \quad v_{l,q}^{l_f q_f} = e^{-i\left(\frac{2\pi}{N} l l_f + \frac{2\pi}{N} q q_f\right)}, \ \ \forall l, l_f, q, q_f = 0, 1, \ldots, N-1.$$

Here the variables l, q index the entries in the FFT basis vector whereas l_f, q_f index the vectors themselves. We then have that

$$\begin{aligned}
\left[KV^{l_f q_f}\right]_{lq} &= \sum_{l',q'=0}^{N-1} K_{lq,l'q'} v_{l'q'}^{l_f q_f} \\
&\overset{(22)}{=} \sum_{l',q'=0}^{N-1} u\left((l-l')\mathrm{mod}N, (q-q')\mathrm{mod}N\right) e^{-i\left(\frac{2\pi}{N} l' l_f + \frac{2\pi}{N} q' q_f\right)} \\
&= e^{-i\left(\frac{2\pi}{N} l l_f + \frac{2\pi}{N} q q_f\right)} \sum_{l',q'=0}^{N-1} u\left((l-l')\mathrm{mod}N, (q-q')\mathrm{mod}N\right) e^{-i\left(\frac{2\pi}{N}(l-l') l_f + \frac{2\pi}{N}(q-q') q_f\right)} \\
&= e^{-i\left(\frac{2\pi}{N} l l_f + \frac{2\pi}{N} q q_f\right)} \psi(l_f, q_f) = \psi(l_f, q_f) v_{l,q}^{l_f q_f},
\end{aligned}$$

for all $l_f, q_f, l, q = 0, 1, \ldots, N-1$. Here,

$$\psi(l_f, q_f) = \sum_{l,q=0}^{N-1} u(l,q) e^{-i\left(\frac{2\pi}{N} l l_f + \frac{2\pi}{N} q q_f\right)}. \tag{23}$$

Therefore,

$$KV^{l_f q_f} = \psi(l_f, q_f) V^{l_f q_f}, \quad \forall l_f, q_f = 0, 1, \ldots, N-1.$$

This shows that the matrix K is indeed diagonalizable by the FFT basis vectors and that $\psi(l_f, q_f)$ are its eigenvalues and, thus, its diagonal entries in that basis. Note that the diagonal elements will be real for K_{11} and K_{22} because of their symmetry but they will likely be complex for K_{12} because of the $\frac{L_x}{2N}$ shift.

At this point, a framework for direct diagonalization follows easily. The elements of the FFT diagonal of the covariance matrix, D_a, where a is either $12/22/W$, can be generated directly by using (23), as opposed to constructing Q first. This direct diagonalization is necessary because the generation of the unitary matrix Q and the computation of the product $Q'KQ$ represents a substantial bottleneck for large values of N. The subscript notation used for the time-domain covariance matrices is recycled for the directly diagonalized matrices D_{22}, D_W, and D_{12}. With this, all of the expensive operations in (20a) can be conducted in the frequency domain after direct diagonalization with (23). The adjustment to the posterior mean $\hat{f}$ is computed in the frequency domain and then mapped back to the time domain with a two-dimensional inverse FFT and added to the posterior mean:

$$\begin{aligned}
f^* &= m_2 + \hat{f} \in \mathbb{R}^N \\
\hat{f} &= \mathcal{F}^{-1}(D_{21} {D_W}^{-1} \hat{W}) \in \mathbb{R}^N \\
D_W &= D_{11} + \sigma_n^2 I_M \in \mathbb{R}^{N \times N},
\end{aligned} \tag{24}$$

where $\hat{W} = \mathcal{F}(y - m(x))$.

The FFT diagonal of the covariance matrix can be used for sampling and thus does not have to be mapped into the time domain. This fact has significant efficiency implications because large matrix products and inverses in posterior computation are avoided. Likewise, there is no need for taking large matrix square roots when sampling from $\mathcal{N}(f^*, K^*)$ (20):

$$D^* = D_{22} - D_{21}^* {D_W}^{-1} D_{12} \in \mathbb{R}^{N \times N}. \tag{25}$$

3.2. Matrix-Free Multivariate Gaussian Sampling

Sampling from a multivariate Gaussian distribution matrix such as $\mathcal{N}(f^*, K^*)$ is a mathematically simple but computationally expensive operation as it requires a large matrix square root of K^*. In the time domain, a Cholesky decomposition of the covariance matrix is taken, and the upper triangular matrix $L^{1/2}$ is multiplied by a vector of standard normals u. This procedure brings in the $O(N^3)$ computational effort for kernels that are not compactly supported.

The posterior distribution sampling can be done more efficiently in the frequency domain by first creating $\tilde{u} = \mathcal{F}(u)$ where $u \sim \mathcal{N}(0_N, I_N)$ and simulating

$$\begin{aligned} S^* &= f^* + \hat{f}^* \in \mathbb{R}^N \\ \hat{f}^* &= \mathcal{F}^{-1}((D^*)^{1/2}\tilde{u}) \in \mathbb{R}^N. \end{aligned} \tag{26}$$

The Cholesky decomposition is avoided entirely, leading to substantial efficiency gains, especially when Monte Carlo simulation is employed to compute trends and errors in dynamic simulations. Numerical results reported in Section 5 demonstrate the accuracy of the proposed approach. Since this methodology is based entirely on FFT operations, it has a theoretical computational effort bound of $O(NlogN)$, compared to the $O(N^3)$ bound of the conventional Cholesky-based approaches.

4. INCREMENTAL GAUSSIAN PROCESS APPROACH

The conditional sampling methodology discussed in the Appendix, in combination with (18), allows for an approach where small grid spaces (subspaces) are interpolated dynamically *at runtime* throughout a simulation. This can replace the all-at-once (or everywhere) sampling at pre-simulation stage, allowing for constant computational effort regardless of problem size.

To simplify the discussion, it is assumed that the interest is in the distribution of states associated with the multibody system at equally spaced time intervals; i.e., the probability density function of the states $x_1(\Delta t)$, $x_1(2\Delta t)$, ..., $x_1(n\Delta t)$ (x_1 is used in this section to denote the state of the dynamical system to avoid confusion with the spatial variable x used in the Gaussian process description). The initial configuration x_0 is given, and the uncertain interaction of the system with the environment is captured by η in (2). The latter quantity has a Gaussian process prior distribution (9) and its simulation must be conditioned on measured data D at a small number of sites. Using the conditional probability rule (18) yields

$$p(x_1(\Delta t), x_1(2\Delta t), \ldots, x_1(n\Delta t)|x_0, \eta) = \prod_{i=1}^{n} p(x_1(i\Delta t)|x_1((i-1)\Delta t), x_1((i-2)\Delta t), \cdots, x_1(\Delta t), x_0, \eta).$$

Given the functional random variable η and $x_1((i-1)\Delta t)$, $x_1(i\Delta t)$ can be evaluated by integrating the differential equation (2) for one time step Δt. This leads to a Markov-chain behavior where the conditional distribution depends only on the previous state, that is,

$$p(x_1(i\Delta t)|x_1((i-1)\Delta t), x_1((i-2)\Delta t), \cdots, x_1(\Delta t), x_0, \eta) = p(x_1(i\Delta t)|x_1((i-1)\Delta t), \eta).$$

Therefore, the following holds:

$$p(x_1(\Delta t), x_1(2\Delta t), \ldots, x_1(n\Delta t)|x_0, \eta) = \prod_{i=1}^{n} p(x_1(i\Delta t)|x_1((i-1)\Delta t), \eta)$$

Assume that the function f defining the time derivative of x_1 in (2) is bounded in norm by F_B. Then, over the interval $[(i-1)\Delta t, i\Delta t]$ the dynamic system will stay in the ball $\mathcal{B}(x_1((i-1)\Delta t), F_B)$ with probability 1; here $\mathcal{B}(a, r)$ denotes the ball of radius r centered at a. This implies that only values of η inside $\mathcal{B}(x_1((i-1)\Delta t), F_B)$ are needed for computing $p(x_1(i\Delta t)|x_1((i-1)\Delta t), \eta)$. Therefore,

$$p(x_1(\Delta t), x_1(2\Delta t), \ldots, x_1(n\Delta t)|x_0, \eta) = \prod_{i=1}^{n} p(x_1(i\Delta t)|x_1((i-1)\Delta t), \{\eta(x)|x \in \mathcal{B}(x_1(i-1)\Delta t, F_B)\}). \tag{27}$$

This suggests the following strategy for sampling from the distribution of $x_1(\Delta t)$, $x_1(2\Delta t)$, ..., $x_1(n\Delta t)$ conditional on η and x_0. For each $i = 1, 2, \ldots, n$, simulate η in $\mathcal{B}(x_1(i-1)\Delta t, F_B)$ conditional on the data D. Numerically integrate (2) for one step Δt starting at $x_1((i-1)\Delta t)$, return $x_1(i\Delta t)$, and continue with the next i. For complex multibody systems, it might be difficult to estimate the bound F_B. We therefore advance the simulation until we arrive within ϵ of the edge of the currently simulated subspace. Then we resample and restart the simulation. The caveat is that the control points – the points where the simulation is restarted – are no longer equally distributed in time. We note, however, that the justification for (27) can be immediately adapted to accommodate that case.

Based on the previous discussion, a particularly attractive option is the use of a compact kernel. This results in a sparse prior Gaussian process covariance matrix, which has favorable consequences in the computation of the gradient in the training phase (13). It can also be leveraged in the conditional simulation phase, but this has not been exploited here, since, pursuant to (27), the sample size in the incremental simulation is much smaller than in Section 3.

A family of piecewise polynomial covariance functions with compact support that guarantee positive definiteness in $\mathbb{R}^D$ is suggested in [17, Ch. 4.2]. In the $\mathbb{R}^2$ case relevant herein,

$$k(r) = \begin{cases} (1 - \frac{r}{\alpha})^{j+1} & r < \alpha \\ 0 & r > \alpha, \end{cases} \tag{28}$$

where $r = \sqrt{x_1^2 + x_2^2}$, α defines the compact threshold (and is the direct source of ϵ), and j quantifies the function smoothness. For the convergence study carried out in section 5 a compact kernel was needed that approximated the baseline γ-exponential correlation function well. Some trial and error was used to select the hyperparameters for both covariance functions to achieve likeness. If $\gamma = 2$ in (7), and $j = 3$ in (28), then likeness is achieved when $\alpha = 5 * \alpha_{x_1} = 5 * \alpha_{x_2}$. Two like kernels are shown in Figure 4, which displays the scenario $\alpha_{x_1} = \alpha_{x_2} = 2.5$ and $\alpha = 12.5$. Note that the compact kernel decays to zero at $r = \alpha$.

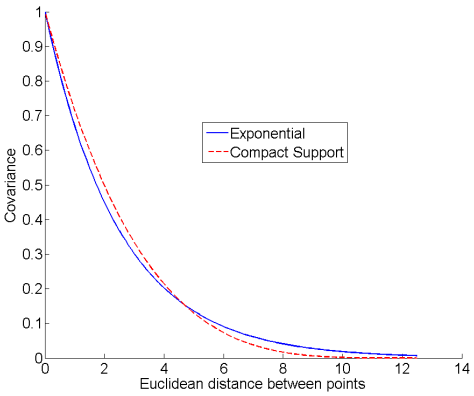


Figure 4. Exponential vs. compact kernels.

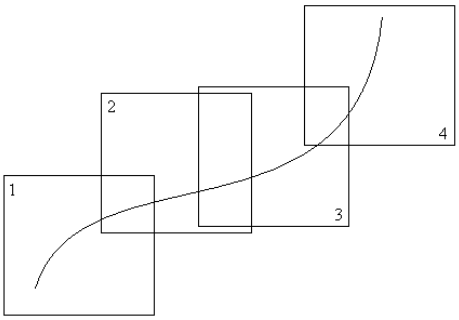


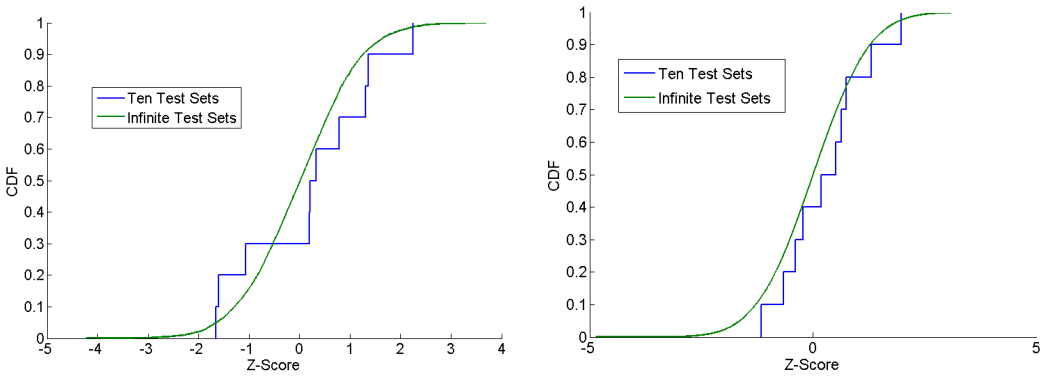
Figure 5. Pictorial representation of IGPA.

Finally, IGPA, the second sampling procedure proposed, is implemented as follows. Imagine a vehicle traveling in uncertain space. For the vehicle model considered, values of η are needed only at the tire/terrain points of contact; thus, only data values as stipulated by (27) within F_B need to be used for training the interpolation. Since it would be computationally expensive to interpolate with Gaussian processes for each integration step, we instead interpolate on subdomains large enough to accommodate the simulation for several time steps; this strategy is presented pictorially in Figure 5.

As previously pointed out, an a priori bound on F_B in (27) is hard to obtain. We thus resample as soon as the trajectory comes within ϵ of the edge of the simulated domain. An ϵ value of about 1/8 of the diameter of the kernel support was found to provide a good bound. Note also that the proof (27) was carried out with the end trajectory point at the center of the new sampling domain. The only change needed for the proof to hold in general is that the trajectory stay for Δt inside of the newly sampled domain. A heuristic was used whereby the new sampling domain had the starting state at about ϵ of the boundary with the velocity vector pointing inside, which was found to still capture the distribution correctly for the considered vehicle simulation.

5. NUMERICAL RESULTS

The numerical results presented validate the methodologies introduced in this paper in conjunction with the nonlinear vehicle model discussed in Section 1.1. In principle, any statistics that draws on states associated with a system described by an initial value problem such as (3) can be solved using these methods. The statistics reported herein are for the final time t_N as well as whole trajectory statistics, $x(t)$; $t \in [t_0, t_N]$. The convergence studies illustrate the accuracy of the methods, while the effort vs. problem-size plots demonstrate the gains obtained in terms of runtime and storage. Some validation results are carried on the Gaussian process $\phi(x)$ alone, without dynamics simulation, in order to validate the equivalence of the sampling methods.



(a) Stationary data: $\gamma = 4$, $\alpha_x = \alpha_y = 1.5$, $a_0 = 1$, $a_1 = a_2 = 0$.

(b) Nonstationary data: $\gamma = 4$, $\alpha_x = \alpha_y = 1.5$, $a_0 = 3$, $a_1 = a_2 = 0.1$.

Figure 6. Cross validation CDF comparison.

5.1. Validation of Hyperparameter Estimation

A leave-one-out cross-validation procedure was used to verify the hyperparameter estimation method. To start, data was generated from known hyperparameters and then divided into ten distinct training and testing sets. The training sets were used to estimate the hyperparameters, and the resulting posteriors were compared to the test set data. Since the test data $\phi \sim \mathcal{N}(\hat{\phi}, \Sigma)$, where $\hat{\phi}$ and Σ are the mean and covariance matrix predicted from the training set, for each test, the expected behavior of the prediction method was demonstrated by confirming that

$$(\phi - \hat{\phi})^T \Sigma^{-1} (\phi - \hat{\phi}) \sim \mathcal{N}(n, \sqrt{2n}) \tag{29}$$

This is the case since $v = \Sigma^{-1/2}(\phi - \hat{\phi}) \sim \mathcal{N}(0, I) \in \mathbb{R}^n$, and therefore $v^T v \sim \chi^2_n$. Equation (29) then follows based on the central limit theorem, where n is the sample size.

The estimation stage was verified using two different sets of hyperparameters. For each hyperparameter test there were ten cross-validations, generating ten Z-scores. The resulting CDFs for the Z-scores are plotted against the expected CDF (given infinite cross-validation tests) in Figure 6 to qualitatively validate the results of the estimation stage.

5.2. Validation of the Spectral Approach

For the spectral approach leading to the PFS method, we investigate only the sampling of $\eta(x)$ itself and not the multibody dynamics simulation. Once sample surfaces consistent with the Gaussian process prior are obtained, simulations can be done by virtually any engineering package followed by postprocessing of the sampling trajectories. Such an approach was followed in [18, 25], where three dimensional nonlinear dynamics of two vehicle models was simulated in the commercial multibody dynamics package ADAMS [26]. The contribution of this work is in the sampling efficiency accompanied by the proof of the fact that the method is consistent with the state prior. This part will be validated by numerical examples below.

5.2.1. Validation of Diagonalization and Diagonal Equivalence To demonstrate the diagonalization property (23) numerically, a diagonalization-by-FFT error operator Ψ is defined by using the MATLAB *diag* function convention:

$$\Psi(K) = \|Q'KQ - \text{diag}\,(\text{diag}(Q'KQ))\,\|. \tag{30}$$

As a first check, at each iteration the priors K_{11} and K_{22} should be equal because of the PFS sampling strategy; the nonhomogeneity of the data is not factored in until the posterior. Results in Table I confirm this (see the $\|K_{11} - K_{22}\|$ column). Second, the diagonalizable-by-FFT property of K_{11}, K_{22}, and K_{12} is confirmed by results reported in last two columns.

Table I. Verification of Covariance Matrix Structure

Iteration	$\|K_{11} - K_{22}\|$	$\Psi(K_{22})$	$\Psi(K_{12})$
1	0	.0003e-10	.0001e-10
2	0	.0027e-10	.0010e-10
3	0	.0079e-10	.0036e-10
4	0	.0275e-10	.0466e-10
5	0	.1943e-10	.1351e-10
6	0	.4960e-10	.7803e-10

To leverage the operations in the frequency domain, the FFT diagonal must be mapped without producing the unitary matrix Q. This is accomplished by using the explicit formula for the diagonal entries in the FFT basis, (23). An operator Γ is used, where

$$\Gamma(K_a) = \|Q'K_aQ - D_a\|, \quad a = 22, 12, W, \tag{31}$$

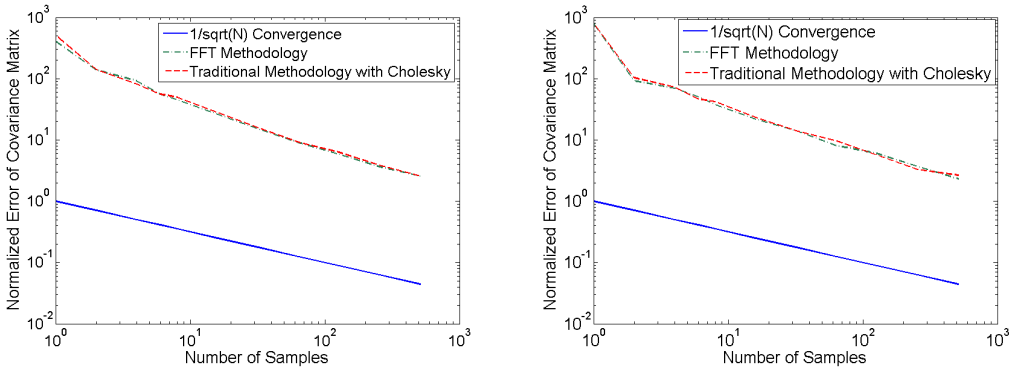
to show in Table II the equivalence of the directly and indirectly derived diagonals for different iterations. Specifically, the table compares the directly and indirectly diagonalized matrices, using (31), to demonstrate their equivalence for seven fold iterations.

Table II. Verification of Equivalence of Directly Diagonalized Matrices

Iteration	$\Gamma(K_{11})$	$\Gamma(K_{22})$	$\Gamma(K_{12})$
1	.0133e-13	.0133e-13	.0022e-13
2	.0178e-13	.0178e-13	.0266e-13
3	.0155e-13	.0155e-13	.0355e-13
4	.0355e-13	.0355e-13	.0375e-13
5	.2842e-13	.2842e-13	.0711e-13
6	.3553e-13	.3553e-13	.4263e-13
7	.5684e-12	.5684e-12	.5969e-12

5.2.2. Convergence Study of Spectral Approach To verify accuracy, the mean vector and covariance matrix computed from the sampled data generated by PFS are compared with

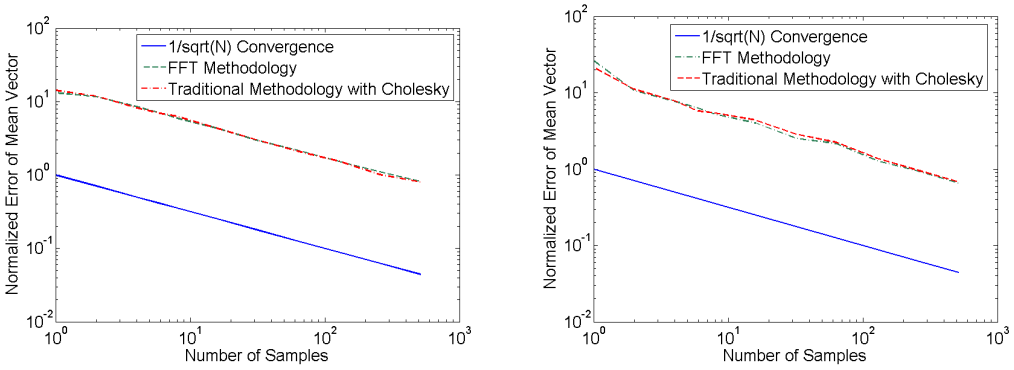
the mean and covariance posteriors computed with traditional methods. In the latter case, all samples are computed at once. The plots in Figure 7 show the normalized error of computed posterior covariance matrices (compared to a Cholesky-derived covariance matrix derived with 1000 samples) as a function of samples, k. Mathmatically, the dependent quantity is $\|\Sigma_{1000} - \Sigma_k\|$ where the subscript represents sample size. Both methods converge at an approximate rate of $1/\sqrt{N}$. The plots in Figure 8 show the normalized error of computed mean vectors (compared to a Cholesky-derived mean vector derived with 1000 samples) as a function of samples, k. Mathematically, the dependent quantity is $\|m_{1000} - m_k\|$. Again, both methods converge at an approximate rate of $1/\sqrt{N}$.



(a) Stationary, high-frequency data: $\gamma = 4, \alpha_x = \alpha_y = 1, a_0 = a_1 = a_2 = 0$.

(b) Non-stationary, low-frequency data $\gamma = 4, \alpha_x = \alpha_y = 3, a_0 = 0.5, a_1 = a_2 = 0.1$.

Figure 7. Comparison of PFS derived covariance matrix convergence and Cholesky derived covariance matrix convergence.



(a) Non-stationary, low-frequency data: $\gamma = 4, \alpha_x = \alpha_y = 3, a_0 = 0, a_1 = a_2 = .5$

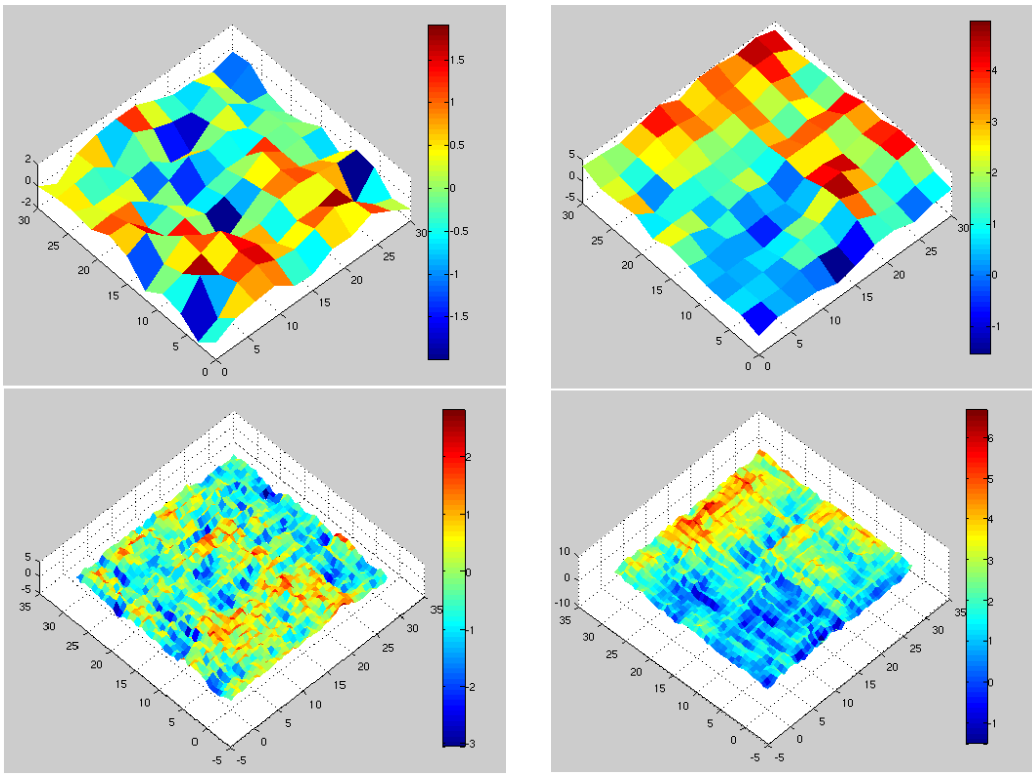
(b) Stationary, high-frequency data: $\gamma = 4, \alpha_x = \alpha_y = 1, a_0 = 1, a_1 = a_2 = 0$

Figure 8. Comparison of PFS derived mean vector convergence and Cholesky derived mean vector convergence.

5.3. Dynamic Simulation

In vehicle simulation, as the tire rolls over terrain, tire/road interface information is needed not only at a discrete set of locations (the nodes of a lattice, for instance), but at an infinite number of locations. Starting with information at a very limited number of points, sample road realizations can be generated on a fine grid. An additional interpolation step is needed to generate information at offgrid locations. In this context, spline interpolation has been used in this work to provide tire/road friction coefficient information at offgrid locations.

Two friction coefficient realizations are shown in Figure 9. In both, Gaussian processes with PFS sampling are used to interpolate the data, refining the space 2^4-fold; one realization is shown for both data sets. The first data set is stationary with high frequency, and the second is nonstationary with low frequency. The resulting refinements were computed in approximately 0.4 seconds.



(a) $\gamma = 4, \alpha_x = \alpha_y = 3, a_0 = a_1 = a_2 = 0.$ (b) $\gamma = 4, \alpha_x = \alpha_y = 5, a_0 = 0, a_1 = a_2 = 0.06.$

Figure 9. Two scenarios and corresponding grid refinements.

For each posterior sample, a cubic spline χ is used to make predictions outside the N-node grid. During the simulation, η at offgrid locations is interpolated allowing for the computation of trajectory at all points at any time. Specifically, a Monte Carlo approach is used to produce the multibody dynamics statistics through simulation.

At points away from the evaluation grid it is evident that the field function $\chi(\eta)$ approximated by splines no longer obeys the Gaussian process model; it is only an approximation of it. One can show, however, that, in the limit of the evaluation grid spacing going to zero, the results of the simulator converge to those that would be obtained if proper Gaussian process sampling had been employed at the points required by the integration procedure. This convergence is due to the fact that almost any sample χ surface is smooth [17]. This convergence is illustrated numerically in [18].

5.3.1. Convergence Study of IGPA The incremental approach is validated numerically with a convergence study of the vehicle trajectory $x(t)$. It is expected that as the number of Monte Carlo samples, N, increases, the trajectory statistics will converge regardless of which method is used for interpolation. Figure 10 shows the $[N = 500]$ mean rotational velocity of the vehicle predicted with full grid interpolation and with IGPA interpolation. Figure 11 shows the relative error of the IGPA predicted mean trajectory as compared with a $[N = 1000]$ mean rotational velocity predicted with full grid interpolation, with N varying for IGPA. Figure 12(a) displays the norm of the deviation vector; the error decays at a rate of $1/\sqrt{N}$. Figure 12(b) shows the Kolmogorov-Smirnov statistic (KSS) comparing the distribution of $x(t_N)$ predicted by $[N = 1000]$ full grid interpolation, with the trajectory distribution predicted by different N IGPA interpolation. With F as the cumulative distribution function, $KSS = \max(F(x_1) - F(x_2))$ is a minimum distance estimation used to compare empirically derived cumulative distribution functions. For N sufficiently large, the convergence rate for the KSS is $1/\sqrt{N}$, which validates IGPA as a method that draws on (27).

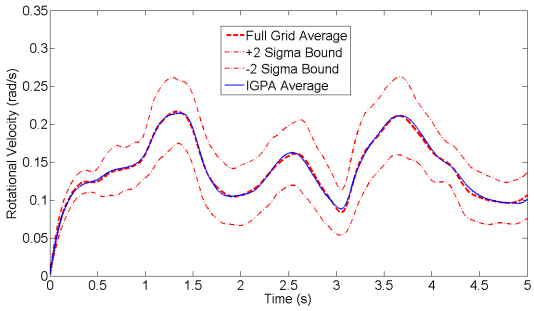


Figure 10. Mean rotational velocity predicted by full grid interpolation and IGPA interpolation: $N = 500$.

5.4. Efficiency Comparisons

The plot shown in Figure 13 demonstrates the computational efficiency of PFS in terms of runtime and memory storage. PFS runtime starts lower and displays slower growth rate than that of traditional methods conducted in the time domain. Moreover, PFS demonstrates reduced storage requirements. Traditional methods implemented in MATLAB became impractical at eight iterations or approximately 9,000 data points; this requires the storage of matrices with over 81×10^6 entries. In the frequency domain, all information is

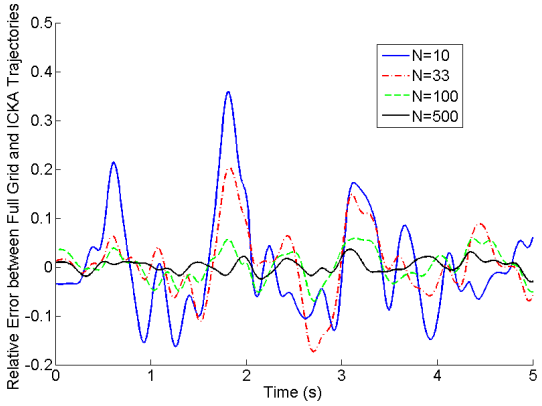
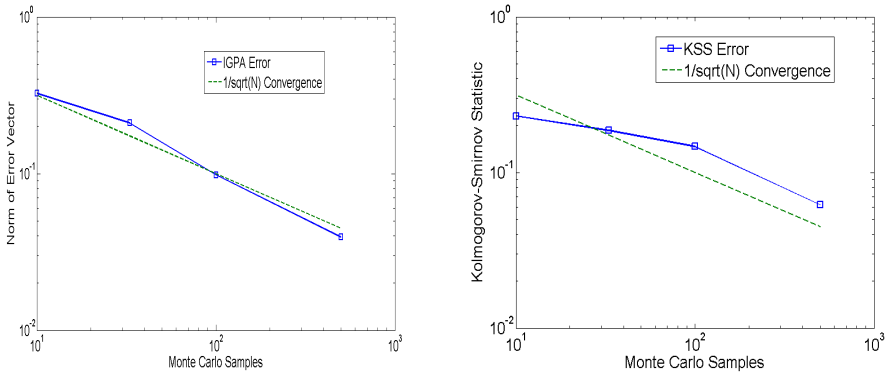


Figure 11. Relative error of IGPA predicted mean rotational velocities, for varying N, as compared with $[N = 1000]$ full grid interpolation prediction.



(a) Convergence of trajectory error $\hat{x}(t) - x(t)$. (b) Convergence of KSS at $x(t_N)$.

Figure 12. Convergence studies for IGPA predicted trajectories. $[N = 1000]$ full grid interpolation trajectory prediction is used for convergence target.

contained in the FFT diagonal values, so the largest storage is proportional to N instead of N^2.

The plot shown in Figure 14 demonstrates the significant computational gains achievable with IGPA for long simulations. The simulation shown is that of a vehicle executing a constant radius turn, with longer simulation times leading to larger problem spaces. Note that the times given in this plot are for a single realization; the incremental method is less useful when Monte Carlo simulation is applied because new realizations require completely fresh simulations as opposed to a single Cholesky decomposition for all realizations as with traditional methods. As shown before, the traditional method exhibits exponential growth in runtime and storage with increasing problem size. It is shown that IGPA has a nearly linear growth in runtime with simulation time because of the constant size of the posterior grids.

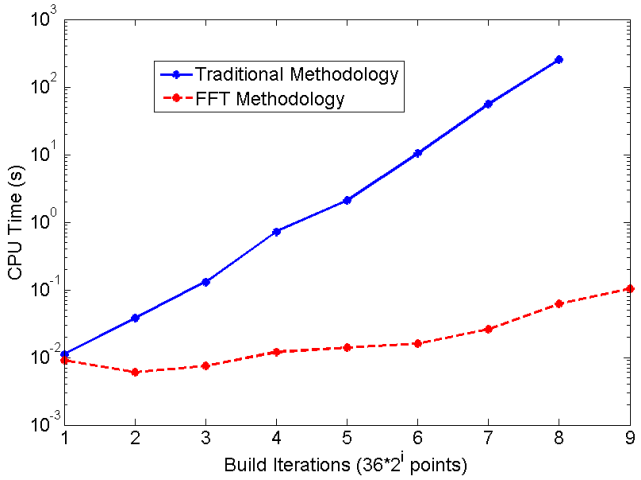


Figure 13. FFT methodology for periodic data shown to be several orders faster than traditional method. The traditional method reaches a memory limit after eight iterations.

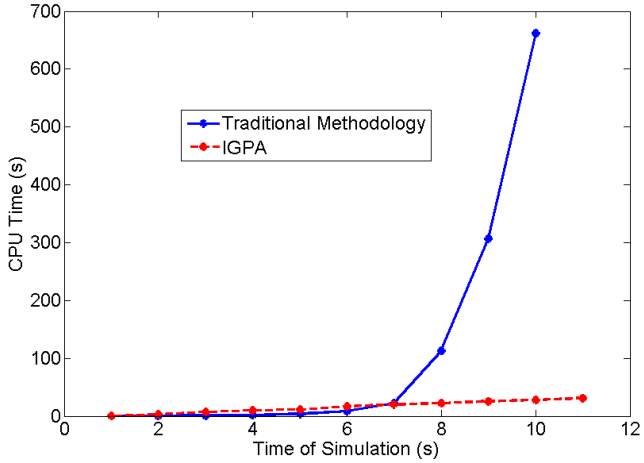


Figure 14. Runtime of traditional method and incremental approach shown as a function of simulation time. The traditional method reaches a memory limit after ten seconds in this simulation.

6. CONCLUSIONS AND FUTURE WORK

The paper introduces two sampling methods, the periodic fold sampling (PFS) and the incremental Gaussian process approach (IGPA), for handling spatial uncertainty in applied multibody dynamics analysis. We prove that the methods correctly sample from the underlying Gaussian process. The methods have both small computational costs and low memory requirements and as such are suitable for large-space and fine-grid spatial uncertainty sampling.

PFS (based on an FFT approach) is effective for interpolation and quantification of spatial data provided on a lattice. With only a periodic data assumption, reflected in the covariance matrix, and a conditional sampling regiment, PFS can do all the steps of traditional Gaussian processes in the frequency domain. Fast Fourier transforms are employed to map between the frequency domain and time domain when needed. Each step in this method – conditional sampling, hyperparameter estimation, computation of posterior with noise incorporated, and multivariate Gaussian sampling – is proven analytically and verified with numerical results. The computational effort to sample a vector of length n is $\mathcal{O}(n \log n)$ as opposed to $\mathcal{O}(n^3)$ effort of the Cholesky-based approach. PFS is limited by the fact that the data provided must reside on a regular grid. Nonetheless, the approach can benefit uncertainty quantification in many areas of mechanical engineering with infinite-dimensional uncertainty subspaces that have smooth realizations. These include contact mechanics and boundary roughness effects in continuum and fluid dynamics.

In IGPA, the method dynamically updates small interpolation spaces in proximity of the location of interest and thus avoids large grid interpolation completely. A Markov chain justification underlines the technique, whose implementation was discussed in detail and validated by means of a compact kernel Gaussian process simulation.

The most important metric presented in this paper is the plot of effort vs. number of samples. It demonstrates that PFS leads to substantial efficiency gains, especially when the sample set is large. Unless special precautions are taken, traditional methods implemented in MATLAB run out of memory between 5,000 and 10,000 sample points. PFS should be able handle up to 10^7 sample points. For IGPA, the Gaussian process interpolation effort is negligible compared to the integration. This leads to a linear relationship between considered length of analysis and actual simulation duration.

With only a few restrictions, the methods proposed combine the accuracy and reliability of Gaussian processes with the speed achievable with spectral or white-noise methods. Furthermore, the restrictions imposed are not severe, making PFS and/or IGPA usable in nearly all analysis scenarios. Their speed and accuracy should prove relevant in industry applications such as vehicle dynamics and nuclear reactor simulation. Finally, this work opens up several research opportunities. Since IGPA relies on a balance between grid size and the number of small grid interpolations required over an entire simulation, an optimal subdomain selection is anticipated to further improve the efficiency of IGPA. Research should also focus on the multi-variate spatial uncertainty problem both from "how" and "how faster" perspectives.

ACKNOWLEDGEMENTS

We are indebted to Michael Stein for insightful comments concerning conditional sampling. We are grateful to three anonymous reviewers whose comments have substantially improved the paper. Dan Negrut and Kyle Schmitt were supported by NSF under award Number CMMI-0700191. Kyle Schmitt and Mihai Anitescu were supported by the Department of Energy under Contract No. DE-AC02-06CH11357, the first under an Argonne Summer Internship Program.

APPENDIX

A proof is provided below that demonstrates that

$$p\left(\begin{pmatrix} S_1 \\ S_2 \end{pmatrix} | m_1, \Sigma_1\right) = p(S_2|m_2, \Sigma_2) * p(S_1|m_3, \Sigma_3), \tag{32}$$

for

$$p(x|m, \Sigma) = (2 * \pi)^{-n/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(x-m)^T \Sigma^{-1}(x-m)\right), \tag{33}$$

where m is the posterior mean vector, Σ is the posterior covariance matrix, x is the sample, and n is the number of elements in the sample.

The expressions involved in this multiplicative equality lead to three parts of the proof: (1) the scalar, (2) the determinant scalar, and (3) the exponential equalities. Separately, the proof below will demonstrate that each of these equalities holds.

1. Numeric Scalar Equality.

$$(2\pi)^{-n_1/2} = (2\pi)^{-n_2/2} * (2\pi)^{-n_3/2},$$

   which follows from $n_1 = n_2 + n_3$. QED
2. Determinant Scalar Equality.
   We must prove that $|\Sigma_1|^{-1/2} = |\Sigma_2|^{-1/2}|\Sigma_3|^{-1/2}$, which can be shown to be identical to the equality $|\Sigma_1| = |\Sigma_2||\Sigma_3|$. From [17], the posteriors are

$$\Sigma_1 = k(\Omega_1, \Omega_1) - k(\Omega_1, D)\left(k(D, D) + \sigma_n^2 I\right)^{-1} k(D, \Omega_1) \tag{34a}$$

$$\Sigma_2 = k(S_2, S_2) - k(S_2, \Omega_2)\left(k(\Omega_2, \Omega_2) + \sigma_n^2 I_D\right)^{-1} k(\Omega_2, S_2) \tag{34b}$$

$$\Sigma_3 = k(S_1, S_1) - k(S_1, D)\left(k(D, D) + \sigma_n^2 I\right)^{-1} k(D, S_1), \tag{34c}$$

   where $\Omega_1 = \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}$ and $\Omega_2 = \begin{pmatrix} D \\ S_1 \end{pmatrix}$. Next, define

$$A = \begin{bmatrix} k(D,D) + \sigma_n^2 I & k(D, S_1) & k(D, S_2) \\ k(S_1, D) & k(S_1, S_1) & k(S_1, S_2) \\ k(S_2, D) & k(S_2, S_1) & k(S_2, S_2) \end{bmatrix} \tag{35}$$

$$B = \begin{bmatrix} k(D,D) + \sigma_n^2 I & k(D, S_1) \\ k(S_1, D) & k(S_1, S_1) \end{bmatrix} \tag{36}$$

$$C = k(D, D) + \sigma_n^2 I. \tag{37}$$

   Schur's determinant formula states that

$$\det \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \det A_{11} \cdot \det\left(A_{22} - A_{21} {A_{11}}^{-1} A_{12}\right). \tag{38}$$

When using the Schur formula while solving for one of the right-hand side terms, we call it the inverse Schur formula. It then follows that

$$\Sigma_1 = \frac{|A|}{|C|}, \qquad \Sigma_2 = \frac{|A|}{|B|}, \qquad \Sigma_3 = \frac{|B|}{|C|}. \tag{39}$$

Finally,

$$\begin{aligned} \frac{|A|}{|C|} &= \frac{|A|}{|B|} * \frac{|B|}{|C|} \\ \frac{|A|}{|C|} &= \frac{|A|}{|C|}. \end{aligned} \tag{40}$$

$$QED$$

3. Exponent Equality.
The following notation is introduced:

$$\mathbf{A} = x_1 - m_1 = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} - k(\Omega_1, D)k(D, D)^{-1}Y_D \tag{41a}$$

$$\mathbf{B} = \Sigma_1 \tag{41b}$$

$$\mathbf{C} = x_2 - m_2 = Y_2 - k(S_2, \Omega_2)k(\Omega_2, \Omega_2)^{-1} \begin{pmatrix} Y_D \\ Y_1 \end{pmatrix} \tag{41c}$$

$$\mathbf{D} = \Sigma_2 \tag{41d}$$

$$\mathbf{E} = x_3 - m_3 = Y_1 - k(S_1, D)k(S_D, S_D)^{-1}Y_D \tag{41e}$$

$$\mathbf{F} = \Sigma_3 \tag{41f}$$

Note that boldface is used to distinguish from other variables previously used in this proof; specifically, note the difference between D for data and $\mathbf{D}$, and the difference between A for the matrix in (35) and $\mathbf{A}$. For clarity, measurement noise is not included in this portion of the proof which can be shown to hold even when $\sigma_n \neq 0$. With the notation introduced, the hypothesis concerning the exponents in (32) is

$$\mathbf{A}^T\mathbf{B}^{-1}\mathbf{A} = \mathbf{C}^T\mathbf{D}^{-1}\mathbf{C} + \mathbf{E}^T\mathbf{F}^{-1}\mathbf{E}. \tag{42}$$

Result α. Taking the inverse Schur with respect to $\mathbf{B}$, yields

$$-\mathbf{A}^T\mathbf{B}^{-1}\mathbf{A} = \frac{\det \begin{bmatrix} \mathbf{B} & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix}}{\det \mathbf{B}}. \tag{43}$$

Taking the inverse Schur with respect to $k(D, D)$ yields

$$\det \mathbf{B} = \frac{\det A}{\det k(D, D)} \tag{44}$$

$$\det \begin{bmatrix} \mathbf{B} & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix} = \frac{\det \begin{bmatrix} \cdot & \cdot & \cdot & Y_D \\ \cdot & A & \cdot & Y_1 \\ \cdot & \cdot & \cdot & Y_2 \\ {Y_D}^T & {Y_1}^T & {Y_2}^T & {Y_D}^T k(D, D)^{-1} Y_D \end{bmatrix}}{\det k(D, D)}. \tag{45}$$

In what follows an empty matrix block will be denoted by $\cdot$. From (43-45),

$$-\mathbf{A}^T\mathbf{B}^{-1}\mathbf{A} = \frac{\det\begin{bmatrix} \cdot & \cdot & \cdot & Y_D \\ \cdot & A & \cdot & Y_1 \\ \cdot & \cdot & \cdot & Y_2 \\ {Y_D}^T & {Y_1}^T & {Y_2}^T & {Y_D}^T k(D,D)^{-1} Y_D \end{bmatrix}}{\det A}. \tag{46}$$

Result β. Using the Schur complement of $\begin{pmatrix} \mathbf{D} & 0 \\ 0 & \mathbf{F} \end{pmatrix}$

$$\mathbf{C}^T\mathbf{D}^{-1}\mathbf{C} + \mathbf{E}^T\mathbf{F}^{-1}\mathbf{E} = -\frac{\det\begin{bmatrix} \mathbf{D} & 0 & \mathbf{C} \\ 0 & \mathbf{F} & \mathbf{E} \\ \mathbf{C}^T & \mathbf{E}^T & 0 \end{bmatrix}}{\det\mathbf{D}\cdot\det\mathbf{F}}. \tag{47}$$

Result γ. Taking the inverse Schur with respect to $k(D,S_1)$ of D and inverse Schur with respect to $k(D,D)$ of $\mathbf{F}$ yields

$$\det\mathbf{D} = \frac{\det A}{\det k(\Omega_2,\Omega_2)}, \quad \det\mathbf{F} = \frac{\det k(\Omega_2,\Omega_2)}{\det k(D,D)}, \quad \det\mathbf{D}\cdot\det\mathbf{F} = \frac{\det A}{\det k(D,D)}. \tag{48}$$

Result δ. First, execute a row transformation, then an inverse Schur with respect to $k(D,D)$, then a Schur with respect to $k(\Omega_2,\Omega_2)$:

$$\begin{aligned} \det\begin{bmatrix} \mathbf{D} & 0 & \mathbf{C} \\ 0 & \mathbf{F} & \mathbf{E} \\ \mathbf{C}^T & \mathbf{E}^T & 0 \end{bmatrix} &= \det\begin{bmatrix} \mathbf{F} & 0 & \mathbf{E} \\ 0 & \mathbf{D} & \mathbf{C} \\ \mathbf{E}^T & \mathbf{C}^T & 0 \end{bmatrix} \\ &= \frac{1}{\det k(D,D)}\cdot\det\begin{bmatrix} k(\Omega_2,\Omega_2) & \cdot & 0 & Y_D \\ \cdot & \cdot & 0 & Y_1 \\ 0 & 0 & \mathbf{D} & \mathbf{C} \\ {Y_D}^T & {Y_1}^T & \mathbf{C}^T & {Y_D}^T k(D,D)^{-1} Y_D \end{bmatrix} \\ &= \frac{1}{\det k(D,D)}\cdot\det k(\Omega_2,\Omega_2)\cdot\begin{bmatrix} \mathbf{D} & \mathbf{C} \\ \mathbf{C}^T & \Xi \end{bmatrix}, \end{aligned} \tag{49}$$

where

$$\Xi = {Y_D}^T k(D,D)^{-1} Y_D - \begin{pmatrix} Y_D \\ Y_1 \end{pmatrix}^T k(\Omega_2,\Omega_2)^{-1}\begin{pmatrix} Y_D \\ Y_1 \end{pmatrix}. \tag{50}$$

Taking the inverse Schur with respect to $k(\Omega_2,\Omega_2)$, $k(\Omega_2,\Omega_2)$ cancels out, to yield

$$\det\begin{bmatrix} \mathbf{D} & 0 & \mathbf{C} \\ 0 & \mathbf{F} & \mathbf{E} \\ \mathbf{C}^T & \mathbf{E}^T & 0 \end{bmatrix} = -\frac{1}{\det k(D,D)}\cdot\det\begin{bmatrix} \cdot & \cdot & \cdot & Y_D \\ \cdot & A & \cdot & Y_1 \\ \cdot & \cdot & \cdot & Y_2 \\ {Y_D}^T & {Y_1}^T & {Y_2}^T & {Y_D}^T k(D,D)^{-1} Y_D \end{bmatrix}. \tag{51}$$

Combining Results α,β,γ, and δ proves (42). *QED*

REFERENCES

1. Shabana, A. A., 2005. *Dynamics of Multibody Systems*, third ed. Cambridge University Press.
2. Hairer, E., and Wanner, G., 1996. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems.* Springer.
3. Shabana, A. A., 2008. *Computational Continuum Mechanics*, first ed. Cambridge University Press.
4. Eldred, M. S., Giunta, A. A., van Bloemen Waanders, B. G., Wojtkiewicz, S. F., Hart, W. E., and Alleva, M. P., 2002. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis. SAND Report SAND2001-3796, Sandia National Laboratories, April.
5. Helton, J., and Davis, F., 2000. "Sampling-based methods". *Sensitivity Analysis*, pp. 101–153.
6. Helton, J., and Davis, F., 2003. "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems". *Reliability Engineering and System Safety,* **81**(1), pp. 23–69.
7. Haldar, A., and Mahadevan, S., 2000. *Probability, Reliability, and Statistical Methods in Engineering Design.* John Wiley, Chichester, England.
8. Huang, D., Allen, T., Notz, W., and Zeng, N., 2006. "Global optimization of stochastic black-box systems via sequential kriging meta-models". *Journal of Global Optimization,* **34**(3), pp. 441–466.
9. Jones, D., Schonlau, M., and Welch, W., 1998. "Efficient global optimization of expensive black-box functions". *Journal of Global Optimization,* **13**(4), pp. 455–492.
10. Tu, J., Choi, K., and Park, Y., 1999. "A new study on reliability-based design optimization". *Journal of Mechanical Design,* **121**, pp. 557–564.
11. Ghanem, R., and Spanos, P., 1991. *The Stochastic Finite Element Method: A Spectral Approach.* Springer, New York.
12. Ghanem, R., and Red-Horse, J., 1999. "Propagation of uncertainty in complex physical system using a stochastic finite element approach". *Physica D,* **133**, pp. 137–144.
13. Xiu, D., and Hesthaven, J., 2005. "High-order collocation methods for differential equations with random inputs". *SIAM Journal on Scientific Computing,* **27**, pp. 1118–1139.
14. Stein, M., 1999. *Interpolation of Spatial Data: Some Theory for Kriging.* Springer, Berlin.
15. Knotters, M., Brus, D., and Voshaar, J., 1995. "A comparison of kriging, co-kriging and kriging combined with regression for spatial interpolation of horizon depth with censored observations". *Geoderma,* **67**, pp. 227–246.
16. Hengl, T., Heuvelink, G., and Stein, A., 2003. Comparison of kriging with external drift and regression-kriging. Tech. rep., International Institute for Geoinformation Science and Earth Observation (ITC), Enschede, Netherlands.
17. Rasmussen, C., and Williams, C., 2006. *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, Massachusets.
18. Schmitt, K., Madsen, J., Negrut, D., and Anitescu, M., 2008. "A Gaussian process-based approach for handling uncertainty in vehicle dynamics simulation". In Proceedings of the International Mechanical Engineering Congress and Exposition, Boston, MA, ASME.
19. Narayanan, S., 1990. "Stochastic optimal-control of nonstationary response of a single degree-of freedom vehicle model". *Journal of Sound and Vibration,* **141**, pp. 449–463.
20. Hammond, J. K., and Harrison, R. F., 1984. *Covariance equivalent forms and evolutionary spectra for nonstationary random processes*, Vol. 62. Springer, Berlin.
21. Yao, T., 1998. "Conditional spectral simulation with phase identification". *Mathematical Geology,* **30**, pp. 285–308.
22. Dietrich, C. R., and Newsam, G. N., 1993. "A fast and exact method for multidimensional Gaussian stochastic simulations". *Water Resources Research,* **29**, pp. 2861–2870.
23. Chiles, J. P., and Delfiner, P., 1999. *Geostatistics: Modeling spatial uncertainty.* Wiley, New York.
24. Wong, J., 2001. *Theory of Ground Vehicles.* John Wiley & Sons, New York.
25. Negrut, D., Datar, M., Gorsich, D., and Lamb, D., 2008. "A framework for uncertainty quantification in nonlinear multi-body system dynamics". In Proceedings of the 26^{th} Army Science Conference, Orlando, Florida.
26. MSC.Software, 2005. ADAMS User Manual. Available online at http://www.mscsoftware.com.