LCD Sampling Strategies for Trajectory-Based Bi-Fidelity Stochastic Collocation

Julian Schreiber, Daniel Frisch, Brandon A. Jones, and Uwe D. Hanebeck

***Abstract*— This paper studies bi-fidelity uncertainty propagation for dynamic systems under uncertain parameters. Low-fidelity trajectories are generated for many samples, representative samples are selected by QR-based column subset selection, and high-fidelity simulations are performed only at these selected samples. We compare random sampling with deterministic strategies based on localized cumulative distributions (LCD) and a reduced Karhunen–Loève representation of the noise process. The method is easy to implement, since it relies only on existing solvers and standard linear algebra.**

We then demonstrate the method on a controlled vehicle model with uncertain initial conditions and correlated steering perturbations. The example is used as a simple computational test case for comparing sampling strategies in a trajectory-based bi-fidelity framework. An explicit Euler solver is used as the low-fidelity model, while a fourth-order Runge–Kutta solver serves as the high-fidelity reference. The results show that deterministic sampling can improve the bi-fidelity estimate in several test cases.

I. INTRODUCTION

Uncertainty propagation for dynamical systems often requires the repeated solution of an initial value problem with random inputs. In this work, the uncertain input consists of random initial conditions and a time-dependent steering perturbation. For each realization of these inputs, the controlled ordinary differential equation becomes deterministic and produces one trajectory. A typical quantity of interest is the expected final position after propagation. A direct Monte Carlo estimate is straightforward [1], but it requires a large number of model evaluations. If each trajectory is computed with an accurate high-fidelity solver, this can become computationally expensive.

Several approaches have been developed to reduce this cost. Intrusive stochastic Galerkin methods based on polynomial chaos expansions represent the random solution in an orthogonal polynomial basis and project the governing equations onto this basis [2]. These methods can converge rapidly for suitable random inputs, but they often lead to coupled deterministic systems and may require substantial modifications of existing solvers. As a non-intrusive alternative, stochastic collocation methods approximate the random solution from deterministic model evaluations at selected parameter points. In the classical form, one writes

$$\hat{u}(z) = \sum_{k=1}^{M} u(z_k) L_k(z) \ ,$$

where z_k are collocation points and L_k are Lagrange basis functions [3]. This keeps the deterministic model evaluations independent, but high accuracy may still require many expensive high-fidelity simulations.

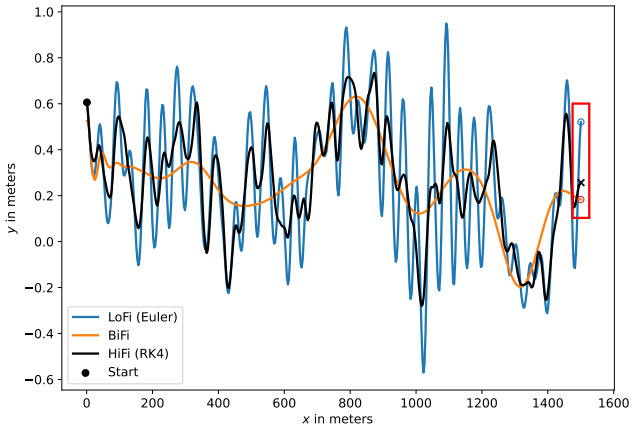

Fig. 1. Trajectories $\underline{u}(\underline{z})$ of the low-fidelity (blue), high-fidelity (black), and bi-fidelity (orange) models for one test realization. The final-time state (red box) is the quantity of interest used for the Monte Carlo mean estimate.

Multifidelity methods address this difficulty by combining models of different cost and accuracy. A cheap low-fidelity model is used to explore the parameter space, while the high-fidelity model is evaluated only at a smaller number of informative samples [4]. In the bifidelity stochastic collocation approach of Narayan, Gittelson, and Xiu [5], many low-fidelity samples are first computed and used to identify representative parameter points. The high-fidelity model is then evaluated only at these selected points. For a new input z, the low-fidelity trajectory is projected onto the span of the selected low-fidelity trajectories, and the same coefficients are used to reconstruct a high-fidelity approximation. Related trajectory-based ideas have been used in orbit uncertainty propagation, where low-fidelity propagated samples define a basis and a small number of high-fidelity propagations correct this basis [6].

The contribution of this paper is to study LCD-based deterministic sampling strategies [7] for trajectory-based bi-fidelity stochastic collocation [5]. The goal is to investigate how the construction of the low-fidelity candidate set affects

Julian Schreiber, Daniel Frisch, and Uwe D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany (e-mail: julian.schreiber2.edu; daniel.frisch@kit.edu; uwe.hanebeck@kit.edu).

Brandon A. Jones is with the Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, C0600, Austin, TX, USA (e-mail: brandon.jones@utexas.edu).

the final bi-fidelity estimate, and whether deterministic sampling can improve the result compared with purely random sampling.

We evaluate this idea on a controlled vehicle model with uncertain initial conditions and correlated steering noise. The noise process is represented by a reduced Karhunen–Loève expansion [8], and the representative samples are identified using QR-based column subset selection [9]. The resulting bi-fidelity approximation is used to estimate the final-time mean position.

II. Bi-Fidelity Framework for Uncertainty Propagation

In this section, we follow the general stochastic collocation setup presented in [5]. Here, we consider the special case in which the differential system consists of an ODE, and the random parameter input is composed of uncertain initial conditions together with a discrete-time trajectory of noise. As the corresponding low- and high-fidelity models, we consider discretized solutions of this system at different fidelity levels.

A. Parameter-Dependent Controlled Dynamics

In the general setup, we model the uncertain input as a finite-dimensional random variable $\underline{Z}$. With a slight abuse of notation we write

$$\underline{Z} \in I_{\underline{Z}} \subset \mathbb{R}^d \ . \tag{1}$$

We assume that $\underline{Z}$ consists of two components: an uncertain initial condition and a time-dependent perturbation trajectory. More precisely, we write

$$\underline{Z} = (\underline{s}_0, \underline{A}) \ ,$$

where

$$\underline{s}_0 \in I_0 \subset \mathbb{R}^{d_0}$$

denotes the random initial condition and

$$\underline{A} \in I_{\underline{\alpha}} \subset \mathbb{R}^{d_{\underline{\alpha}}}$$

denotes the random perturbation trajectory. Hence, the full parameter domain is given by

$$I_{\underline{Z}} = I_0 \times I_{\underline{\alpha}} \subset \mathbb{R}^{d_0 + d_{\underline{\alpha}}}.$$

A realization of Z is denoted by

$$\underline{z} = (\underline{s}_0, \alpha) \in I_{\underline{Z}} \ .$$

For each fixed realization z, the controlled system becomes deterministic. The general formulation in [5] is motivated by parameter-dependent PDEs. In this work, we use the corresponding finite-dimensional ODE setting and write the controlled initial value problem as

$$\frac{d}{dt}\underline{s}(t,z) = f\big(\underline{s}(t,\underline{z}),\, a(t,\underline{z}),\, \underline{\alpha}(t)\big) \ , \qquad \underline{s}(0,\underline{z}) = \underline{s}_0 \ .$$

We assume that for each fixed realization z, this initial value problem admits a unique solution on the time interval $[0,T]$. Here, $\underline{s}(t,z)$ denotes the system state, $a(t,z)$ denotes the control input, and $\underline{s}_0$ is the initial condition contained in the realization $\underline{z} = (\underline{s}_0, \underline{\alpha})$. See Fig. 1 for a visualization of one such trajectory realization. The perturbation trajectory $\underline{\alpha}$ is also part of this realization. It enters the dynamics through the time-dependent term $\underline{\alpha}(t)$.

Instead of considering the continuous-time solution of this ODE, we follow the ideas presented in [6] and look at the corresponding discrete-time trajectories. We thus consider a discrete-time grid with n time steps t_i

$$0 = t_0 < t_1 < \cdots < t_n = T$$

and the system outputs $\underline{s}(t_i, \underline{z})$ at these time steps, which we concatenate to a long column vector $\underline{u}(\underline{z})$

$$\underline{u}(\underline{z}) = \big(\underline{s}(t_0,\underline{z}), \underline{s}(t_1,\underline{z}), \ldots, \underline{s}(t_n,\underline{z})\big)^\top \ .$$

Using the full discrete trajectory $\underline{u}(z)$ rather than only the final state provides a higher-dimensional output vector and therefore a richer basis for the subsequent bi-fidelity approximation. Thus, each realization $\underline{z}$ is mapped to one trajectory sample, which is used in the sample-based uncertainty propagation and in the bi-fidelity approximation below.

B. Sample-Based Viewpoint

As discussed above, the random input $\underline{Z}$ takes values in the parameter domain $I_{\underline{Z}}$. In the setting of stochastic collocation, the uncertainty in $\underline{Z}$ is represented by a finite set of M realizations $\underline{z}_k$

$$\Gamma = \{\underline{z}_1, \ldots, \underline{z}_M\} \subset I_{\underline{Z}} \ .$$

Each point $\underline{z}_k \in \Gamma$ defines one deterministic model evaluation and therefore one discrete trajectory $\underline{u}(\underline{z}_k)$, as introduced above. We write the corresponding set of trajectory samples as

$$\underline{u}(\Gamma) = \{\underline{u}(\underline{z}_1), \ldots, \underline{u}(\underline{z}_M)\} \ .$$

The choice of Γ is relevant for uncertainty quantification, since it determines how the distribution of $\underline{Z}$ is represented by finitely many model evaluations. In this work, we consider both standard Monte Carlo samples and deterministic sampling strategies.

C. Bi-Fidelity Approximation

We now introduce the bi-fidelity approximation used to reduce the number of expensive model evaluations. Following the general idea of Narayan, Gittelson, and Xiu [5], we assume that two models of different fidelity are available. The low-fidelity model is denoted by

$$u^L : I_{\underline{Z}} \to V^L \ ,$$

and the high-fidelity model by

$$u^H : I_{\underline{Z}} \to V^H \ .$$

Here, V^L and V^H denote the discrete trajectory spaces associated with the low- and high-fidelity models, respectively. In the simplest case, both models return trajectories on the same time grid and these spaces can be identified. More generally, they are kept distinct to allow for different discretizations or resolutions.

The high-fidelity model u^H is assumed to be more accurate but more expensive to evaluate, while the low-fidelity model u^L is cheaper but less accurate. A direct high-fidelity evaluation on all samples in

$$\Gamma = \{\underline{z}_1, \dots, \underline{z}_M\} \subset I_{\underline{Z}}$$

would require the computation of

$$u^H(\Gamma) = \{\underline{u}^H(\underline{z}_1), \dots, \underline{u}^H(\underline{z}_M)\} \ .$$

This can be computationally expensive when M is large.

The bi-fidelity approach first evaluates the low-fidelity model on the full sample set

$$u^L(\Gamma) = \{\underline{u}^L(\underline{z}_1), \dots, \underline{u}^L(\underline{z}_M)\} \ .$$

These low-fidelity trajectories are used to select a smaller set of informative samples. One possible selection criterion is the greedy rule

$$\underline{z}_n = \arg\max_{\underline{z}\in\Gamma} d^L\big(\underline{u}^L(\underline{z}), U^L(\gamma_{n-1})\big) \ , \quad \gamma_n = \gamma_{n-1} \cup \{\underline{z}_n\} \ . \tag{2}$$

where

$$U^L(\gamma_{n-1}) = \mathrm{span}\{\underline{u}^L(\underline{z}) : \underline{z} \in \gamma_{n-1}\} \ ,$$

Here, d^L denotes the distance induced by the norm on the low-fidelity trajectory space. In the present numerical setting, the trajectories are represented as vectors, so this distance is taken as the Euclidean distance to the span of the already selected low-fidelity trajectories

$$\gamma = \{\underline{z}_{i_1}, \dots, \underline{z}_{i_r}\} \subset \Gamma \ , \qquad r \ll M \ .$$

The high-fidelity model is then evaluated only at the selected samples

$$u^H(\gamma) = \{\underline{u}^H(\underline{z}_{i_1}), \dots, \underline{u}^H(\underline{z}_{i_r})\} \ .$$

For a new realization $\underline{z} \in I_{\underline{Z}}$, the low-fidelity trajectory $\underline{u}^L(\underline{z})$ is approximated by a linear combination of the selected low-fidelity trajectories

$$\underline{u}^L(\underline{z}) \approx \sum_{j=1}^{r} c_j(\underline{z}) \underline{u}^L(\underline{z}_{i_j}) \ .$$

The coefficients $c_j(\underline{z})$ are obtained by projecting $u_L(\underline{z})$ onto the span of the selected low-fidelity trajectories via the pseudoinverse. The same coefficients are then used to construct a bi-fidelity approximation of the high-fidelity trajectory

$$\widehat{\underline{u}}^H(\underline{z}) = \sum_{j=1}^{r} c_j(\underline{z}) \underline{u}^H(\underline{z}_{i_j}) \ .$$

Thus, the low-fidelity model is used to determine the reconstruction coefficients, while high-fidelity evaluations are required only at the selected samples in γ.

D. Sample-Based Estimation of Statistics

Bi-fidelity methods have also been used to estimate statistical moments from Monte Carlo or collocation samples [10]. Here, we apply

this idea in a trajectory-based setting.

The bi-fidelity approximation is constructed for complete discrete trajectories. For the statistical evaluation, however, we only consider selected components of the final state. Let P denote the projection onto these components of interest. We define the final-time output by

$$p_T(\underline{u}(\underline{z})) = P(\underline{s}(T, \underline{z})) \ .$$

These components could, for example, be selected position coordinates of the final state.

The expected high-fidelity final position is

$$\underline{\mu}_T^H = \mathbb{E}\left[p_T(\underline{u}^H(\underline{Z}))\right] \ .$$

The expectation is taken with respect to the probability distribution of the uncertain input $\underline{Z}$.

Given a sample set $\Gamma = \{\underline{z}_1, \dots, \underline{z}_M\}$, this expectation is approximated by the Monte Carlo estimator

$$\widehat{\underline{\mu}}_T^H = \frac{1}{M} \sum_{i=1}^{M} p_T(\underline{u}^H(\underline{z}_i)) \ .$$

This direct estimator requires M high-fidelity model evaluations.

In the bi-fidelity approach, the high-fidelity trajectories $\underline{u}^H(\underline{z}_i)$ are replaced by their reconstructed approximations $\widehat{\underline{u}}^H(\underline{z}_i)$. The corresponding bi-fidelity estimator is

$$\widehat{\underline{\mu}}_T^{BF} = \frac{1}{M} \sum_{i=1}^{M} p_T(\widehat{\underline{u}}^H(\underline{z}_i)) \ .$$

Thus, the bi-fidelity approximation is used to estimate the expected high-fidelity final position while requiring high-fidelity evaluations only at the selected samples. See the orange line in Fig. 1 for such a bi-fidelity approximation: its final point is much closer to the high-fidelity than to the low-fidelity.

In this work, we are particularly interested in reducing the error between the corresponding Monte Carlo estimators,

$$e_{\mathrm{MC}} = \left\| \widehat{\underline{\mu}}_T^H - \widehat{\underline{\mu}}_T^{BF} \right\|_2 \ . \tag{3}$$

III. Candidate Set Construction and Sample Selection

This section describes how the informative sample set $\gamma \subset \Gamma$ is computed in practice. We first construct a finite candidate set Γ of input samples, which may be generated either by random Monte Carlo sampling or by deterministic sampling strategies. The low-fidelity model is then evaluated on all samples in Γ, and the resulting trajectories are collected in a snapshot matrix. A column subset selection method is applied to this matrix in order to identify representative samples for high-fidelity evaluation.

A. Sampling of the Candidate Set

The candidate set

$$\Gamma = \{\underline{z}_1, \ldots, \underline{z}_M\} \subset I_{\underline{Z}}$$

contains the input samples used for the low-fidelity evaluations. Each sample $\underline{z}_i$ represents one realization of the uncertain input and therefore defines one deterministic trajectory.

The set Γ can be generated either randomly or deterministically. Random sampling corresponds to standard Monte Carlo sampling. As a deterministic alternative, we use localized cumulative distribution (LCD) sampling, based on the LCD framework for comparing multivariate random vectors [7]. In this approach, the target density f of the uncertain input is approximated by a finite Dirac mixture $\widetilde{f}$

$$\widetilde{f}(\underline{z}) = \sum_{i=1}^{M} w_i \, \delta(\underline{z} - \underline{z}_i) \ .$$

Using a kernel $K(\cdot; \underline{m}, b)$, centered at $\underline{m}$ with positive scale parameter b, the LCD of f is

$$F(m, b) = \int_{I_{\underline{Z}}} f(\underline{z}) \, K(\underline{z}; \underline{m}, b) \, d\underline{z} \ ,$$

where the integral is taken over the possibly multidimensional parameter domain $I_{\underline{Z}}$. For the Dirac mixture, the corresponding LCD is

$$\widetilde{F}(m, b) = \sum_{i=1}^{M} w_i \, K(\underline{z}_i; \underline{m}, b) \ .$$

The deterministic sample points $\underline{z}_i$ are then obtained by minimizing a discrepancy between F and $\widetilde{F}$ over relevant locations $\underline{m}$ and scales b. In this sense, LCD sampling can be interpreted as an optimization-based construction of a finite sample set that represents the distribution of $\underline{Z}$. Compared to random sampling, the systematic placement of LCD samples typically requires fewer samples to represent the target density accurately.

B. Sampling the Noise Process

With the limited number of deterministic samples used in this study, direct sampling of the full noise vector did not give reliable results in preliminary tests. We therefore reduce the stochastic dimension of the noise vector $\underline{\alpha}$. In this paper, we use the discrete Karhunen–Loève expansion (KLE) as presented in [8]. We consider the covariance matrix

$$\mathbf{C}_\alpha \in \mathbb{R}^{n \times n}$$

of the discretized correlated noise process.

Since $\mathbf{C}_\alpha$ is symmetric and positive semidefinite, it admits the eigenvalue decomposition

$$\mathbf{C}_\alpha = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^\top,$$

where

$$\mathbf{Q} = \begin{bmatrix} \underline{q}_1 & \cdots & \underline{q}_n \end{bmatrix}$$

contains the orthonormal eigenvectors and

$$\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$$

contains the corresponding eigenvalues, ordered as

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0 \ .$$

A realization of the Gaussian noise vector can then be written as

$$\underline{\alpha}_k = \mathbf{Q} \mathbf{\Lambda}^{1/2} \underline{\xi}_k, \qquad \underline{\xi}_k \sim \mathcal{N}(\underline{0}, \mathbf{I}_n),$$

where

$$\mathbf{\Lambda}^{1/2} = \operatorname{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n}) \ .$$

To reduce the stochastic dimension, only the R dominant eigenmodes are retained. With

$$\mathbf{Q}_R = \begin{bmatrix} \underline{q}_1 & \cdots & \underline{q}_R \end{bmatrix} \ , \qquad \mathbf{\Lambda}_R = \operatorname{diag}(\lambda_1, \ldots, \lambda_R) \ ,$$

the truncated representation is

$$\underline{\alpha}_k \approx \mathbf{Q}_R \mathbf{\Lambda}_R^{1/2} \underline{\xi}_k \ , \qquad \underline{\xi}_k \sim \mathcal{N}(\underline{0}, I_R) \ .$$

Since the noise process is Gaussian, the KLE coefficients $\underline{\xi}_k$ are independent standard normal variables. Thus, deterministic sampling is only applied to the reduced coefficient vector $\underline{\xi}_k \in \mathbb{R}^R$, rather than to the full noise vector $\underline{\alpha}_k \in \mathbb{R}^n$.

C. Column Subset Selection

To identify the important directions in the low-fidelity trajectory ensemble, we consider the low-fidelity snapshot matrix

$$\mathbf{S}^L = \begin{bmatrix} \underline{u}^L(z_1) & \underline{u}^L(z_2) & \cdots & \underline{u}^L(z_M) \end{bmatrix} \ .$$

Analogously, the high-fidelity snapshot matrix $\mathbf{S}^H$ is defined. The singular values of this matrix indicate how well the snapshot space can be represented by a low-dimensional subspace. By the Eckart–Young–Mirsky theorem [11], the truncated singular value decomposition gives an optimal rank-k approximation in the Frobenius norm. The relative amount of captured energy is given by

$$\frac{\sum_{i=1}^{k} \sigma_i^2(\mathbf{S}^L)}{\sum_{j=1}^{\rho} \sigma_j^2(\mathbf{S}^L)} \ ,$$

where $\rho = \operatorname{rank}(\mathbf{S}^L)$. This criterion can be used to assess whether a low-dimensional approximation of the snapshot matrix is sufficient.

However, the basis vectors obtained from the SVD are not actual snapshot columns and therefore do not correspond to individual trajectories $\underline{u}^L(\underline{z}_i)$. Since the bi-fidelity procedure requires selected parameter samples $\underline{z}_i$ for high-fidelity evaluation, we use column subset selection instead. Here the goal is to select representative columns from $\mathbf{S}^L$. In the following, we use the basic ideas of rank-revealing QR factorizations, following Gu and Eisenstat [9], to compute such a subset.

We then see that the greedy rule in (2) can be implemented by QR factorization with column pivoting applied to the low-fidelity snapshot matrix $\mathbf{S}^L$. It computes a column permutation Π such that

$$\mathbf{S}^L \Pi = \mathbf{Q}\mathbf{R}, \qquad \mathbf{R} = \begin{pmatrix} \mathbf{A}_k & \mathbf{B}_k \\ 0 & \mathbf{C}_k \end{pmatrix} \ .$$

Algorithm 1 QR with column pivoting for fixed number of samples

$k := 0, \quad \mathbf{R} := \mathbf{S}^L, \quad \mathbf{\Pi} := \mathbf{I}$
while $k < r$ **do**
  $j_{\max} := \arg\max_{1 \le j \le M-k} \gamma_j(\mathbf{C}_k)$
  $k := k + 1$
  Move column $k + j_{\max} - 1$ to position k
  Update the QR factorization and the permutation $\mathbf{\Pi}$
end while

Here, the first k permuted columns correspond to the currently selected snapshots. The block $\mathbf{A}_k$ is associated with these selected columns, while $\mathbf{C}_k$ denotes the residual block of the remaining columns after projection onto the span of the selected columns. Here, $\mathbf{C}_k$ is only a QR block and should not be confused with the covariance matrix $\mathbf{C}_\alpha$.

Following the notation of [9], let

$$\gamma_j(\mathbf{C}_k) = \|(\mathbf{C}_k)_{:,j}\|_2$$

denote the Euclidean norm of the j-th residual column. Thus, $\gamma_j(\mathbf{C}_k)$ measures how much of the corresponding remaining snapshot is not yet represented by the span of the already selected snapshots. QR with column pivoting selects the remaining column with the largest residual norm. The corresponding QR-pivoting procedure is summarized in Algorithm 1.

After termination, the first r columns of the permuted snapshot matrix $\mathbf{S}^L\mathbf{\Pi}$ define the selected low-fidelity trajectories and the corresponding sample set

$$\gamma = \{\underline{z}_{i_1}, \dots, \underline{z}_{i_r}\} \subset \Gamma \ .$$

Although QR with column pivoting provides an efficient greedy strategy, the resulting column subset is not guaranteed to be optimal. In particular, the selected columns may lead to a numerically unstable representation of the remaining columns. This is relevant in the bi-fidelity setting, since the remaining low-fidelity snapshots are obtained from the selected ones by projection and the resulting coefficients are later transferred to the high-fidelity trajectories.

Algorithm 2 is not expected to directly minimize the final Monte Carlo error. Its purpose is to improve the numerical quality of the selected subset, for example by avoiding nearly linearly dependent selected columns. Therefore, a more stable column subset does not necessarily lead to a smaller error in the final-time statistics.

To obtain a more robust column subset, Gu and Eisenstat [9] introduce the Strong QR strategy based on column exchanges. The idea is to improve the selected block $\mathbf{A}_k$ after an initial column ordering has been computed. The quality of this block is related to its singular values: very small singular values of $\mathbf{A}_k$ indicate that the selected columns are close to being linearly dependent. At the same time, the residual block $\mathbf{C}_k$ should be small, since it represents the part of the remaining columns not captured by the selected columns. For the leading block $\mathbf{A}_k$, the product of its singular values

Algorithm 2 Strong RRQR column exchange for fixed number of samples

Compute an initial QR factorization with column permutation and r selected columns
Partition

$$R = \begin{pmatrix} \mathbf{A}_r & \mathbf{B}_r \\ 0 & \mathbf{C}_r \end{pmatrix}$$

while there exists a selected column i and a non-selected column j such that

$$\frac{\det(\widetilde{\mathbf{A}}_r)}{\det(\mathbf{A}_r)} > g$$

do
  Exchange selected column i with non-selected column j
  Update the QR factorization, the permutation $\mathbf{\Pi}$, and the block $\mathbf{A}_r$
end while

can be expressed by $\mathbf{C}_k$ through the identity

$$\det(\mathbf{A}_k) = \prod_{i=1}^{k} \sigma_i(\mathbf{A}_k) = \frac{\sqrt{\det((\mathbf{S}^L)^\top \mathbf{S}^L)}}{\prod_{j=1}^{M-k} \sigma_j(\mathbf{C}_k)} \ .$$

This motivates an exchange criterion based on $\det(\mathbf{A}_k)$. Selected and non-selected columns are interchanged whenever the exchange sufficiently increases the determinant of the leading block. A swap is accepted if the new leading block $\widetilde{\mathbf{A}}_r$ satisfies

$$\frac{\det(\widetilde{\mathbf{A}}_r)}{\det(\mathbf{A}_r)} > g, \qquad g \ge 1 \ .$$

The process is repeated until no such improving exchange exists.

IV. Vehicle Control Test Case

In this section, we apply the bi-fidelity framework introduced above to a controlled vehicle test problem. The example is intentionally simple and is used to study the effect of solver fidelity and sampling strategy on the bi-fidelity approximation. We first describe the kinematic vehicle model, then define the Euler and Runge–Kutta based fidelity models, and finally present numerical results for different sampling strategies.

A. Controlled Vehicle Model

We now introduce a simplified planar kinematic bicycle model without leaning dynamics, where the front and rear axles are each represented by a single wheel. The model is based on the formulation in [12]. While that formulation uses the center of mass as the reference point, we use the rear-wheel contact point instead, see Fig. 2.

The vehicle state $\underline{s}$ with position (r_x, r_y) and orientation ϕ is given by

$$\underline{s}(t, z) = \big(r_x(t, \underline{z}),\, r_y(t, \underline{z}),\, \phi(t, \underline{z})\big)^\top.$$

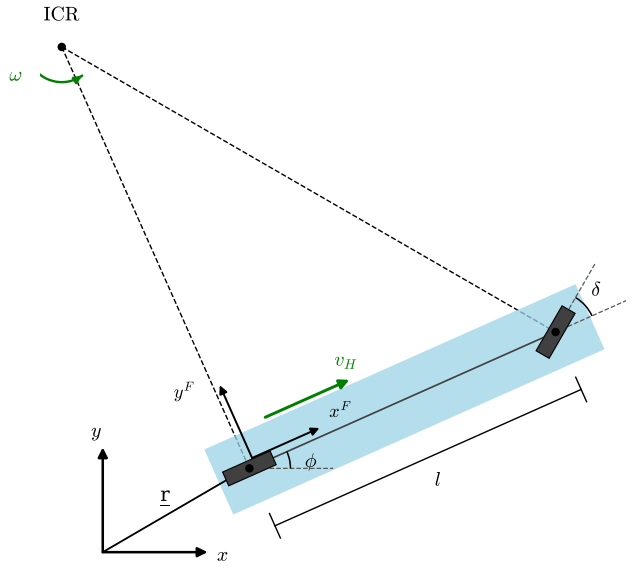

Fig. 2. Rear-wheel reference formulation of the kinematic bicycle model. The state position $r = (r_x, r_y)$ refers to the rear-wheel contact point, v_H denotes the rear-wheel speed, ϕ is the vehicle orientation, and δ is the steering angle.

Let $\delta(t, \underline{z})$ denote the steering command generated by the controller and let $\alpha(t)$ denote a time-dependent steering perturbation. The kinematic bicycle dynamics are given by

$$\begin{aligned} \dot{r}_x(t, \underline{z}) &= v_H \cos\phi(t, \underline{z}) \ , \\ \dot{r}_y(t, \underline{z}) &= v_H \sin\phi(t, \underline{z}) \ , \\ \dot{\phi}(t, \underline{z}) &= \frac{v_H}{l} \tan(\delta(t, \underline{z}) + \alpha(t)) \ . \end{aligned}$$

Because the rear wheel is the reference point, v_H is aligned with the vehicle orientation ϕ. Thus, the position dynamics do not require an additional slip angle.

To keep the vehicle close to the straight reference path $r_y = 0$ and $\phi = 0$, we use a proportional feedback law for the desired yaw rate,

$$\omega_{\text{des}}(t, z) = -k_y r_y(t, \underline{z}) - k_\phi \phi(t, \underline{z}) \ ,$$

where $k_y > 0$ and $k_\phi > 0$ are feedback gains. The corresponding steering command is obtained from the bicycle relation

$$\omega = \frac{v_H}{l} \tan\delta \ ,$$

which gives

$$\delta(t, \underline{z}) = \arctan\left(\frac{l}{v_H} \omega_{\text{des}}(t, \underline{z})\right) \ .$$

The steering perturbation is modeled as a correlated random trajectory on the time grid. We write

$$\alpha = (\alpha_0, \ldots, \alpha_{n_T - 1})^T \sim \mathcal{N}(0, \Sigma_\alpha) \ ,$$

with covariance matrix

$$(\Sigma_\alpha)_{ij} = \sigma_\alpha^2 \exp\left(-\frac{|i-j|}{\ell}\right), \qquad i, j = 0, \ldots, n_T - 1 \ .$$

Here, σ_α controls the perturbation magnitude and ℓ is the correlation length measured in time steps.

B. Euler/RK4 Fidelity Pair

The purpose of this test case is to study the bi-fidelity construction in a controlled numerical setting. Therefore, the fidelity hierarchy is not introduced through different physical models, but through different numerical solvers applied to the same controlled vehicle dynamics. The explicit Euler method is used as a cheap low-fidelity solver, while the classical fourth-order Runge–Kutta method is used as a more accurate reference solver. Both methods are standard one-step methods for ordinary differential equations; see, for example, [13]. In this sense, RK4 serves as the high-fidelity model relative to Euler.

Both solvers are evaluated for the same uncertain input $\underline{z} = (\underline{s}_0, \alpha)$ and return discrete trajectories on the same output time grid. We write

$$\underline{u}^L(\underline{z}) = \underline{u}_{\text{Euler}}(\underline{z}) \ , \qquad \underline{u}^H(\underline{z}) = \underline{u}_{\text{RK4}}(\underline{z}) \ .$$

This intentionally simple setup is used to demonstrate that the bi-fidelity approach can achieve reasonable approximation accuracy with a substantially reduced number of high-fidelity evaluations. It is not intended as a detailed computational cost benchmark against direct high-fidelity Monte Carlo.

C. Experimental Setup

For all experiments, we use $M = 3000$ training samples to construct the low-fidelity snapshot matrix and select $N = 15$ samples for high-fidelity evaluation. The choice of N is based on preliminary numerical tests: although the singular-value decay indicates an effectively low-dimensional trajectory ensemble, the bi-fidelity approximation still improves up to about $N = 15$, while larger values did not lead to a significant improvement of the final-time quantities considered here. The approximation is evaluated on an independent test set of 1000 samples, and all reported values are averaged over several random seeds. The test samples are drawn from the full correlated noise model and are identical for all candidate-set construction strategies.

The trajectories are computed on the time interval $[0, 100]$ using 500 equidistant time points. This relatively long time horizon is chosen such that solver discrepancies between the Euler low-fidelity model and the RK4 high-fidelity model can accumulate and become visible in the final-time quantities of interest.

We vary the vehicle speed in the range

$$v_H \in \{20, 25, 30, 40\}\,\text{m/s} \ .$$

Smaller velocities are not used as main test cases, since preliminary experiments showed that the low- and high-fidelity solutions are already close in this regime and the bi-fidelity correction gives no substantial improvement.

The uncertain initial condition is sampled according to

$$x_0 \sim \mathcal{N}(0, 1) \ , \quad y_0 \sim \mathcal{N}(0, 1) \ , \quad \phi_0 \sim \mathcal{N}\left(0, (1^\circ)^2\right) \ .$$

For the deterministic variants, samples from these Gaussian distributions are generated using the Dirac mixture

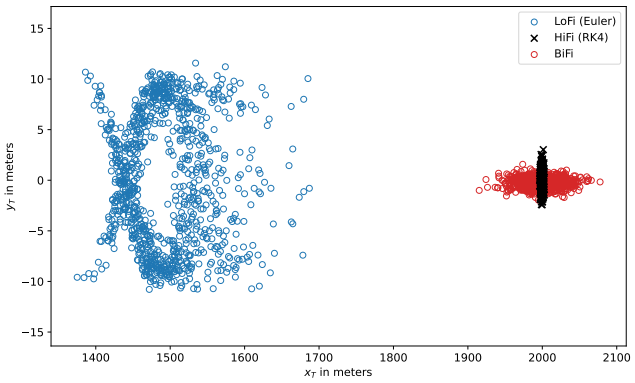

Fig. 3. Final-time endpoint samples for $v_H = 20\,\mathrm{m/s}$. The bi-fidelity reconstruction $p_T(\widehat{u}^H(\Gamma))$ (red) (construction with D-$I \times \alpha$) is much closer to the high-fidelity reference $p_T(u^H(\Gamma))$ (black) than the low-fidelity model $p_T(u^L(\Gamma))$ (blue).

approximation method of [14]. The initial heading angle is sampled from a normal distribution because its standard deviation is small. For larger angular uncertainties, a circular distribution such as the von Mises distribution would be more appropriate.

The steering perturbation has standard deviation $\sigma_\alpha = 1°$ and temporal correlation factor $c_\alpha = 0.1$, corresponding to a correlation length of approximately 50 time steps. For the KLE-based perturbation models, we use $R = 10$ retained modes. Larger numbers of retained modes were also tested, but did not lead to an improvement in the final-time quantities considered here. In the repeated-product construction, 300 deterministic initial-condition samples are combined with 10 deterministic perturbation trajectories, yielding again 3000 training samples. This split was chosen because it gave the best performance among the tested product decompositions while keeping the total training size fixed.

See Fig. 1 for a visualization of the low-, high-, and bi-fidelity trajectories originating from one exemplary input $\underline{z}$ and furthermore Fig. 3 for the end points $p_T(u(\Gamma))$ from all trajectories $u(\Gamma)$.

D. Results

We compare the error in the Monte Carlo estimate of the final-time mean, as defined in (3), for different sampling strategies of the low-fidelity candidate set Γ. The results are reported for different vehicle speeds in Table I.

TABLE I

MONTE CARLO MEAN ERROR FOR DIFFERENT VEHICLE SPEEDS.

v_H	LoFi	D-I	D-I/R-α	D-I/D-α	D-$I \times \alpha$	Rand
20	520.17	3.31	4.99	1.73	**0.93**	8.78
25	1466.38	**1.49**	19.10	1.63	1.55	20.92
30	2436.60	18.17	95.86	**16.46**	18.82	62.12
40	4522.81	101.04	**24.82**	92.35	95.68	32.16

Vehicle speed v_H is given in $\mathrm{m/s}$. All error values are in meters. The best result for each v_H is shown in bold.

The abbreviations in Table I are defined as follows. LoFi denotes the direct low-fidelity estimator. D-I uses deterministic initial-condition samples $\underline{s}_0$ and zero steering perturbation. D-I/R-α combines deterministic initial-condition samples with randomly sampled perturbation trajectories $\underline{\alpha}$. D-I/D-α pairs deterministic initial-condition samples directly with deterministic KLE perturbation samples. D-$I \times \alpha$ denotes the repeated-product construction, where each deterministic initial-condition sample is combined with all deterministic KLE perturbation trajectories, see also Fig. 3. For this construction, the total number of candidate samples is the product of the number of deterministic initial-condition samples and deterministic perturbation samples, so the split between both sample sets can affect the result. Rand denotes fully random sampling of both initial conditions and perturbations.

The results show that the low-fidelity estimator alone has a large bias for the final-time mean, especially as the vehicle speed increases. The bi-fidelity approximation strongly reduces this error for all tested velocities. For $v_H = 20, 25,$ and $30\,\mathrm{m/s}$, at least one deterministic construction clearly outperforms the fully random baseline. The strongest improvement is observed for $v_H = 20\,\mathrm{m/s}$, where the repeated-product construction reduces the MC mean error from $8.78\,\mathrm{m}$ for random sampling to $0.93\,\mathrm{m}$. For $v_H = 25\,\mathrm{m/s}$, the deterministic variants based on initial-condition sampling and KLE perturbations perform similarly well and are much more accurate than the random baseline. At $v_H = 30\,\mathrm{m/s}$, the deterministic KLE construction remains the best method among the tested strategies.

At $v_H = 40\,\mathrm{m/s}$, the behavior changes. The deterministic KLE-based constructions no longer outperform the random baseline, and the best result is obtained by deterministic initial-condition sampling combined with random perturbation trajectories. This indicates that the reduced performance at $v_H = 40\,\mathrm{m/s}$ is not only caused by the sampling strategy. At this velocity, the low- and high-fidelity trajectory spaces appear to be less compatible, so the coefficient transfer becomes more difficult.

Although the random perturbation has a much higher effective noise rank than the KLE-based deterministic perturbations, the resulting trajectory snapshot spaces remain low-dimensional. Thus, the dynamics strongly compress the stochastic input. However, the $40\,\mathrm{m/s}$ case shows that low rank alone is not sufficient; the low- and high-fidelity spaces must also be well aligned. Based on Björck and Golub [15], we therefore compare the principal angles between $\mathrm{span}(S^L)$ and $\mathrm{span}(S^H)$. The largest angle increases from about $3°$ at $20\,\mathrm{m/s}$ to about $13°$ at $30\,\mathrm{m/s}$, but reaches about $22°$–$26°$ at $40\,\mathrm{m/s}$. This indicates weaker subspace alignment at the highest velocity. Therefore, even if deterministic sampling finds a good basis in the low-fidelity space, this basis may not be equally suitable for the high-fidelity space, and the coefficient transfer can become less reliable.

As an additional robustness check, we compare Algorithm 1 and Algorithm 2 for the repeated-product sampling strategy. Algorithm 1 corresponds to QR with column pivoting, while Algorithm 2 applies the strong RRQR column-

exchange procedure. The results are shown in Table II. Both selectors lead to very similar MC mean errors. Algorithm 2 performs only a small number of additional swaps and does not consistently improve the final-time mean estimate.

This is plausible because the selector chooses columns based on the low-fidelity snapshot matrix, whereas the reported error measures the accuracy of the final-time mean after the full bi-fidelity reconstruction. Therefore, an improved column selection in the linear-algebraic sense does not necessarily imply a smaller MC mean error. Overall, the comparison indicates that the main results are not strongly dependent on the particular choice between Algorithm 1 and Algorithm 2. Together with the principal-angle results, this suggests that the poorer performance at 40m/s is mainly caused by weaker low-/high-fidelity subspace alignment rather than the column-selection method.

TABLE II

SELECTOR COMPARISON FOR REPEATED-PRODUCT SAMPLING.

v_H	Algorithm 1	Algorithm 2	swaps
20	$\mathbf{0.93}$	1.07	2
25	$\mathbf{1.55}$	1.72	2
30	18.82	$\mathbf{18.70}$	4
40	95.68	95.68	0

v_H is given in $\mathrm{m/s}$. Algorithm 1 and Algorithm 2 report e_{MC} in meters. The last column gives the number of additional swaps performed by Algorithm 2.

V. CONCLUSION

This paper studied a trajectory-based bi-fidelity approach for uncertainty propagation in a controlled vehicle test case. The results show that the bi-fidelity approximation strongly reduces the error of the low-fidelity estimator while requiring only a small number of high-fidelity evaluations.

Deterministic sampling improves the result in several cases and can provide a better low-fidelity trajectory basis than random sampling. However, this is not guaranteed for all parameter regimes. At $40\,\mathrm{m/s}$, the principal-angle diagnostic shows a weaker alignment between the low- and high-fidelity trajectory spaces. In this case, the main limitation is not only the sampling strategy, but the reduced compatibility between the two fidelity models. Thus, a good low-fidelity basis alone is not sufficient for reliable coefficient transfer.

Future work should therefore investigate more systematically under which conditions the low- and high-fidelity trajectory spaces are well aligned. This is important because the success of the bi-fidelity reconstruction depends on the transferability of low-fidelity coefficients to the high-fidelity space. Such an analysis could help to identify when the bi-fidelity assumption is valid and when a different or improved low-fidelity model is required.

REFERENCES

[1] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed. Springer, 2004.

[2] D. Xiu and G. E. Karniadakis, "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 619–644, 2002.

[3] D. Xiu and J. S. Hesthaven, "High-order collocation methods for differential equations with random inputs," *SIAM Journal on Scientific Computing*, vol. 27, no. 3, pp. 1118–1139, 2005.

[4] B. Peherstorfer, K. Willcox, and M. Gunzburger, "Survey of multifidelity methods in uncertainty propagation, inference, and optimization," *SIAM Review*, vol. 60, no. 3, pp. 550–591, 2018.

[5] A. Narayan, C. Gittelson, and D. Xiu, "A stochastic collocation algorithm with multifidelity models," *SIAM Journal on Scientific Computing*, vol. 36, no. 2, pp. A495–A521, 2014.

[6] B. A. Jones and R. Weisman, "Multi-fidelity orbit uncertainty propagation," *Acta Astronautica*, vol. 155, pp. 406–417, 2019.

[7] U. D. Hanebeck and V. Klumpp, "Localized cumulative distributions and a multivariate generalization of the Cramér–von Mises distance," in *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Republic of Korea, 2008, pp. 33–39.

[8] A. Alexanderian, "A primer on the Karhunen–Loève expansion," North Carolina State University, 2026.

[9] M. Gu and S. C. Eisenstat, "Efficient algorithms for computing a strong rank-revealing QR factorization," *SIAM Journal on Scientific Computing*, vol. 17, no. 4, pp. 848–869, 1996.

[10] X. Zhu, E. M. Linebarger, and D. Xiu, "Multi-fidelity stochastic collocation method for computation of statistical moments," *Journal of Computational Physics*, vol. 341, pp. 386–396, 2017.

[11] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[12] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Redondo Beach, CA, USA, 2017, pp. 812–818.

[13] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, 2nd ed. Springer, 1993.

[14] U. D. Hanebeck, M. F. Huber, and V. Klumpp, "Dirac mixture approximation of multivariate Gaussian densities," in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 3851–3858.

[15] Å. Björck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Mathematics of Computation*, vol. 27, no. 123, pp. 579–594, 1973.