IRTG 1792 Discussion Paper 2021-001

# Surrogate Models for Optimization of Dynamical Systems

Kainat Khowaja [*] [*2] [*3]
Mykhaylo Shcherbatyy [*2]
Wolfgang Karl Härdle [*] [*4] [*5] [*6] [*7]

[*] Humboldt-Universität zu Berlin, Germany
[*2] Ivan Franko National University of Lviv, Ukraine
[*3] University of L'Aquila, Italy
[*4] Singapore Management University, Singapore
[*5] Xiamen University, China
[*6] National Chiao Tung University, Taiwan
[*7] Charles University, Czech Republic

This research was supported by the Deutsche Forschungsgesellschaft through the International Research Training Group 1792 "High Dimensional Nonstationary Time Series".

http://irtg1792.hu-berlin.de
ISSN 2568-5619

# Surrogate Models for Optimization of Dynamical Systems *

Kainat Khowaja, Mykhaylo Shcherbatyy, and Wolfgang Karl Härdle

**Abstract** Surrogate models using a suitable orthogonal decomposition and radial basis functions have been proposed by many researchers to reduce the computational complexity of numerical solutions to optimization problems. However, these reduced-order models result in low accuracy, sometimes due to inappropriate initial sampling or the occurrence of optima at vertices. This paper provides an improved intelligent data-driven mechanism for constructing low-dimensional surrogate models using alternative memory-based sampling strategies in an iterative algorithm. Furthermore, the application of surrogate models to optimal control problems is extended. It is shown that surrogate models with Latin hypercube sampling dominate variable-order methods in optimization computation time while maintaining accuracy. They are also shown to be robust to nonlinearities in the model. Therefore, these computationally efficient predictive surrogate models are applicable in various fields, especially for solving inverse problems and optimal control problems, some examples of which are shown in this paper.

**Keywords:** Proper Orthogonal Decomposition, SVD, Radial Basis Functions, Optimization, Surrogate Models, Smart Data Analytics, Parameter Estimation

---

Kainat Khowaja
International Research Training Group 1792 "High Dimensional Nonstationary Time Series", Humboldt-Universität zu Berlin, Berlin, Germany; Ivan Franko National University of Lviv, Ukraine; University of L'Aquila, Italy. e-mail: kainat.khowaja@hu-berlin.de

Mykhaylo Shcherbatyy
Ivan Franko National University of Lviv, Ukraine. e-mail: mykhaylo.shcherbatyy@lnu.edu.ua

Wolfgang Karl Härdle
BRC Blockchain Research Center, Humboldt-Universität zu Berlin, Germany; Sim Kee Boon Institute, Singapore Management University, Singapore; WISE Wang Yanan Institute for Studies in Economics, Xiamen University, China; Dept. Information Science and Finance, National Yang Ming Chiao Tung University, Taiwan, ROC; Dept. Mathematics and Physics, Charles University, Czech Republic. Grants–DFG IRTG 1792, CAS: XDA 23020303 and COST Action CA19130 gratefully acknowledged. e-mail: haerdle@hu-berlin.de

* This research was supported by Joint MSc in Applied and Interdisciplinary Mathematics, coordinated by the University of L'Aquila (UAQ) in Italy, Department of Information Engineering, Computer Science and Mathematics (DISIM) and the Deutsche Forschungsgemeinschaft through the International Research Training Group 1792 "High Dimensional Nonstationary Time Series", and European Union's Horizon 2020 training and innovation programme "FIN-TECH", under the grant No. 825215 (Topic ICT-35-2018, Type of actions: CSA).

## 1 Introduction

Over the years, mathematical modeling and optimization techniques have effectively described complex real-life dynamical structures using system of differential equations. More often, the dynamical behavior of such models, especially in optimization and inverse problems (the problems where some of the 'effects' (responses) are known but not some of the 'causes' (parameters) leading to them are unknown), cause necessity of repetitive solution of these model equations with a slight change in system parameters. This parameter exploration process can be computationally intense, specially in complex non-linear systems. While numerical models replaced experimental methods due to their robustness, accuracy, and rapidness, their increasing complexity, high cost, and long simulation time have limited their application in domains where multiple evaluations of the model differential equations are demanded.

To prevent this trade-off between computational cost and accuracy, one needs to focus on reduced order models (ROMs) which provide compact, accurate and computationally efficient representations of ODEs and PDEs to solve these multi-query problems. These approximation models, also commonly recognized as a surrogate models or meta-models [26], allow the determination of solution of model equations for any arbitrary combination of input parameters at a cost that is independent of the dimension of the original problem. They reduce the computational time for solution of the complex optimization problems by using training instances derived from the evaluations of the true objective functions. Accordingly, they meet the most essential criteria of every analysis problem: the criteria of highest fidelity at lowest possible computational cost, where high fidelity is defined by the efficacy of theoretical methods to replicate the physical phenomenons with least possible error [19].

In this work, we use Proper Orthogonal Decomposition (POD), a model reduction technique that originated in statistical analysis and is known for its optimality, capturing the most dominant components of data in an efficient way [15]. POD serves the purpose of dimension reduction by extracting hidden structures from high dimensional data and projecting it on lower dimensional space consisting of basis elements that contain characteristics of the expected solution [20]. We use POD to derive low order models of dynamical system by reducing a large number of interdependent variables to a much smaller number of uncorrelated variables, while preserving as much as possible of the variation in the original variables, in an iterative fashion. We hypothesize that the system responses of dynamical models can be obtained with a very high accuracy, but lower computational cost using this model reduction technique.

The novelty of this research is three-fold: 1) POD is combined with interpolation methods in an iterative procedure, which is an improved methodology for construction of highly accurate surrogate models; 2) the methodology has been extended to optimal control problems which has rarely been explored in the literature; 3) the combination of POD and radial basis functions (RBF) for surrogate models is quite

under utilized for the the economic models, some of which are discussed in this paper.

The computational procedure used in this research is decomposed between offline and online phases. The offline phase (training of the model) entails utilization of sampling techniques to generate data, computation of snapshot matrix of model solutions using variable order methods for solving ODEs (associated with the dynamical systems), obtainment of proper orthogonal modes via Singular Value Decomposition (SVD) and estimation of POD expansion coefficients that approximate the POD basis (via interpolation techniques radial basis functions).

Next, the model quality is evaluated by carrying out error analysis on various experimental designs. These experimental designs are created by varying sampling strategies, interpolation techniques and the size of training set. Using the optimal experimental design, the online phase of algorithm starts. The online phase (testing of the model) involves redefinition of model equations in terms of surrogate models and computation of system responses corresponding to any arbitrary set of input parameters in given domain [26].

Finally, using the optimal experimental design, we solve optimal control problems using both models to evaluate accuracy of the surrogate model. If the error tolerance is not met, an iterative algorithm is implemented to enhance the performance of the surrogate model. All codes used for the analysis in this paper are available on www.quantlet.de

The remainder of this paper is organized as follows. In the next section, we summarize the research on the surrogate models, and identify the limitations of existing literature. In section 3, we explain in detail theoretical concepts related to POD, SVD and RBF, and how these are used to construct surrogate models. In section 4, we discuss the newly proposed iterative algorithm. The perspective of our methodology is demonstrated with practical examples of dynamical systems in section 5. Finally, we conclude the main results and provide a summary of current research, limitations, as well as the future prospects of this research in the last section.

## 2 Literature Review

Over a century ago, Pearson proposed the idea of representing the statistical data in high dimensional space using a low dimensional straight line or plane, hence discovering a finite dimensional equivalence of POD as a tool for graphical analysis [24]. In the years following Pearson's paper, the technique has been independently rediscovered by several other scientists including Kosambi, Hotelling and Van Loan under different names in the literature such as principle component analysis (PCA), Hotelling Transformation and Loeve-Karhunen Expansion, depending on the domain in which it is used. Despite their early discovery, the computational resources required to compute POD modes were limited, and the technique remained virtually unused until the 1950s. The technological advancements subsequently took off with the massive increase in computing power, leading to the popularity of POD [20].

Since then, the development and applications of POD have been widely investigated in various disciplines such as structural mechanics [20], Aerodynamics [19], Signal and Image Processing [4], etc. Due to its strong theoretical foundations, the technique has been used in many applications, such as for damage detection [21], human face recognition [28], detection of signals in multi-channel time-series [30], exploration of peak clustering [5] and many more.

In general, a non-equivalent variant of POD, known as factor analysis, has been renowned and has been used for various applications [1, 2, 3, 23], etc. Unlike POD, factor analysis assumes that the data have a strict factor structure and it looks for the factors that amount for common variance in the data. On contrary, PCA the finite counterpart of POD, allows the accountability of maximal amount of variance for observed variables. The PCA analysis consists of identifying the set of variables, also known as principle components, from the system that retain as much variation from the original set of variables as possible. Similarly, principal expectile analysis (PEC), which generalizes PCA for expectiles was recently developed as a dimension reduction tool for extreme value theory [29]. These POD equivalent tools have also been adopted in analysis on several instances such as [1, 11, 22, 29]. Yet, most of these sources exploit only the real life data for dimension reduction.

Even though the real life data can be very useful in analysis, complete data on relevant dynamical systems is rarely available. This creates an urgent need for the introduction of tools that utilize simulated data. The simulated data can be easily generated by repetitive evaluations of the original set of differential equations using so-called 'method of snapshots'. Bujlak [8] explains how a simulation based matrix is obtained by evaluating the original set of equations with various parameter combinations. The resulting snapshot matrix allows easy implementation of data analytical and smoothing tools for reducing the dimensionality of the dynamical systems.

Very recently, researchers have tried to couple various smoothing methods with POD to improve the performance of surrogate models. In [19], the authors construct a POD+cubic spline surrogate model for an aerodynamic design problem. Similarly, the POD+neural network framework for solving finite element models is proposed in [16]. Some of these sources also recognize problem with these surrogate models, such as [9] in which the authors remedy the common problem of high training time in smoothed POD models using Hadamard product.

Another well known issue with the surrogate models is that some times they result in low accuracy due to occurrence of solution on the boundaries. One way to deal with this issue is to examine the efficacy of POD+RBF surrogate models using various pre-processing methods for the snapshot matrix [14]. One can then examine the errors due to truncation through POD and due to interpolation of the data. Another solution is presented in [31] where surrogate modelling techniques are enhanced by incorporating derivatives of snapshots for the training.

An alternative method for dealing with issue is to change the way snapshots are obtained. In this paper, we propose to deal with this problem by using various sampling methods and an iterative training procedure that truncates the domain of sampling at each step, which provides an improved method for construction of the

surrogate models. We will discuss this technique in detail in section 4, but first, we present the mathematical framework for the POD+RBF surrogate modelling technique.

## 3 Mathematical Framework

Model reduction techniques have been known for their ability to reduce the computational complexity of mathematical models involving numerical simulations. The main reason for increasing applications of ROMs in various disciplines is due to its strong theoretical foundations. Also, computational complexities of high dimensional physical system is ever-so-rising, which has created demand for the model reduction techniques. ROMs address these issues effectively by providing low dimensional approximations of the high dimensional systems.

Although a variety of dimensionality-reduction techniques exist, for example operational based reduction methods [25], reduced basis methods [7], the ROMs are often based upon POD. Analogous to PCA, the POD theory find components of the systems, known as Proper Orthogonal Modes (POMs), that are ordered in a way that each subsequent mode holds less energy than previous one. As stated earlier, POD is ubiquitous in the dimensionality reduction of physical systems. It presents the optimal technique for capturing the system modes in least square sense. That is, for constructing ROM for any system, incorporating $k$ POMs will give the best $k$ component approximation of that system. This assures that any approximation obtained using POD will be the best possible approximation: there is no other method that can reduce the dimensionality of the given system in lower number of components or modes.

In this section, we discuss the mathematical concepts associated with POD and its correspondence with SVD and RBF for construction of surrogate models. The computational procedure presented in the sections 4 and 5 is strictly based on the theory formulated in this section.

### 3.1 Optimal Control Problem for Dynamical Systems

Many problems of optimal control are focused on the minimization and maximization problems. In order to find an optimal set of parameters, optimization models are usually defined in which the problems are summarized by the objective function. These optimization parameters are called control parameters and they affect the choice of allocation. In optimal control problems, these parameters are time paths which are chosen within certain constraints so as to minimize or maximize the objective functional. The applications presented in section 5 are optimization problems, the general structure of which has been discussed in the next paragraph.

Let us consider optimization problem which consists of finding a vector of optimization parameters $u^* \in U_S$ and proper state function $y^* \subset Y_S$ that minimizes the optimization criterion (objective function)

$$\psi_0 = \tilde{\psi}_0(u^*, y^*) = \min_{(u,y)\in U_S \times Y_S} \tilde{\psi}_0(u, y) \tag{1}$$

subject to ODEs (state equation)

$$c(y, u) = 0 \sim \begin{cases} y_i' - f(t, u, y) = 0, \ t \in [t_0, T], \\ y(t_0) - y_0 = 0, \end{cases} \tag{2}$$

box constrains on the control variable

$$U = \{u \in U_S : u^- \leq u \leq u^+, u^- \in U_S, u^+ \in U_S\} \tag{3}$$

and possibly additional equality and non-equality constraints on state and control

$$\begin{array}{c} \tilde{\psi}_j(u, y) = 0, j = 1, \ldots, m_1, \\ \tilde{\psi}_j(u, y) \leq 0, j = m_1 + 1, \ldots, m. \end{array} \tag{4}$$

where $U_S$ and $Y_S$ are real Banach spaces, $u = u(t) = [u_1(t), \ldots, u_{n_u}(t)]^\top \in U_S, y = y(t) = [y_1(t), \ldots, y_{n_y}(t)]^\top \in Y_S, \tilde{\psi}_j : U_S \times Y_S \to \mathbb{R}, j = 0, 1, \ldots, m$

We assume that for each $u \in U$, there exists a unique solution $y(u)$ of state equation $c(y, u) = 0$. For rest of the paper, we will use the compact notation of the optimization problem (1- 4) in its reduced form: find a function $u^*$ such that

$$\begin{array}{c} u^* \in U_{\partial_u}, \psi_0(u_*) = \min_{u\in U_{\partial_u}} \psi_0(u) \\ U_{\partial_u} = \left\{u : u \in U; \psi_j(u) = 0, j = 1, \ldots, m_1; \psi_j(u) \leq 0, j = m_1 + 1, \ldots, m\right\} \\ c(y(u), u) = 0 \\ \psi_j(u) = \tilde{\psi}_j(u, y(u)), j = 0, 1, \ldots, m \end{array} \tag{5}$$

The optimal control problems in this research are solved using direct method. In the direct method, each problem is transformed to nonlinear programming problem, i.e., it is first discretized and then the resulting nonlinear programming problem is optimized. The optimality conditions of undiscretized optimal control problems need to be re-established for each new problem. They also often require partial a-priori knowledge of the mathematical structure of the solution which in general is not available for many practical problems. Therefore, direct methods are preferred because optimality conditions are generic for the discretized optimal control problems.

The first step of the direct method is to approximate each component of the control vector by a function of finite parameters $u_i(t) = u_i(t, b^{(i)}), b^{(i)} = [b_1^{(i)}, ..., b_{n_i}^{(i)}]^\top, i = 1, \ldots, n_u$. As a result, we write control function $u(t)$ as a function of vector of optimization parameters $b$: $u(t) = u(t, b)$. In this paper we use a piecewise-linear or piecewise-constant approximation for each function $u_i(t), i = 1, \ldots, n_u$.

The optimization problem can be written as nonlinear programming problem as following: find a vector $b^*$ such that

$$\begin{gathered} b^* \in U_\partial, \psi_0\left(b^*\right) = \min_{b \in U_\partial} \psi_0(b) \\ U_\partial = \left\{b : b \in U_b, \psi_j(b) = 0, j = 1, \ldots, m_1; \psi_j(b) \leq 0, j = m_1 + 1, \ldots, m\right\} \\ U_b = \{b : b \in R^n, b^- \leq b \leq b^+, b^- \in \mathbb{R}^n, b^+ \in \mathbb{R}^n\} \\ c(y(b), b) = 0 \\ \psi_j(b) = \tilde{\psi}_j(u(b), y(b)), j = 0, 1, \ldots, m \end{gathered} \tag{6}$$

## 3.2 Surrogate Models for Optimization Problems

The optimization problem formulated in equation (6) used for estimation of parameter values is often computationally expensive. It requires repetitive solutions of the state equation $c(y(b), b) = 0$ and the objective function $\tilde{\psi}_0$, subject to the constraints $\tilde{\psi}_j, j = 1, \ldots, m$ for different values of optimization parameters $b$. In order to solve multi-query problems with limited computational resources, often approximation models (also known as surrogates models, meta-models or ROMs) are used. Surrogate models replace the high-fidelity models and tend to have lower numerical complexity, and hence less computational cost.

The first step for construction of the surrogate models is to select an appropriate sampling strategy. Once $n_s$ sampling points are generated, the state equation (6) (ODEs) is solved for each sample point $b^{(i)}$. The resulting $n_s$ vectors of solutions (snapshots) $Y_i = \left[y\left(t_1, b^{(i)}\right)^\top, \ldots, y\left(t_{n_t}, b^{(i)}\right)^\top\right]^\top \in \mathbb{R}^m, m = n_y \times n_t$ at different time instances, $t_0 < t_1 < t_2 < \ldots < t_{n_t} = T$ are called snapshots vectors $Y_i$. The snapshot vectors collectively create the snapshot matrix $Y = [Y_1, Y_2, \ldots, Y_n] \in \mathbb{R}^{m \times n_s}$.

### 3.2.1 Initial sampling and method of snapshots

The method of snapshots for POD was first introduced by Sirovich [27] in 1987. Generally, it comprises of evaluating the model equations for the number of sampling points at various time instances. Each model response is called snapshot and is recorded in a matrix which is collectively called snapshot matrix.

The initial dimension of the problem is equal to the number of snapshots $n_s$ recorded at each time instance $t_i, i = 1, ..., n_t$. These points are selected from the parameter space using some sampling technique. In general, the sampled points should represent the dynamic behaviour of the system. Many researchers simply use the random or uniform sampling, however there is no standard method for generating the sampling points. Nevertheless, the choice of sampling method has direct effects on the accuracy of the model and therefore, it is regarded as an autonomous problem.

This research briefly explores the initial sampling problem by comparing various classical a-priori methods of sampling.

We propose to use Latin Hypercube Sampling (LHS) and its variant Symmetric Latin Hypercube Sampling (SLHS) for sampling. LHS is a memory-based, near-random sampling technique that aims at spreading the sample points evenly across the surface. In statistics, a square grid containing sample positions is a latin square if and only if there is only one sampling point in each row and each column. A latin hypercube is the generalization of this concept to an arbitrary number of dimensions, whereby each sample is the only one in each axis-aligned hyperplane containing it. Unlike random sampling (RS), which is frequently referred as Monte-Carlo method in finance, LHS uses a stratified sampling techniques that remembers the position of previous sampling point and shuffles the inputs before determining the next sampling points. It has been considered to be more efficient in a large range of conditions and proven to have faster speed and lower sampling error than RS [12].

SLHS is an extension of LHS that achieves the purpose of optimal design in a relatively more efficient way. SLHS also has higher minimum distance between randomly generated points than LHS. In a nutshell, both LHS and SLHS are hypothesized to perform better than RS. Nevertheless, sampling is performed using all three techniques in this work to determine which techniques provides optimal sampling of the underlying space and maximizes the system accuracy. A simple sampling distribution of each of the three techniques is illustrated in figure 1.

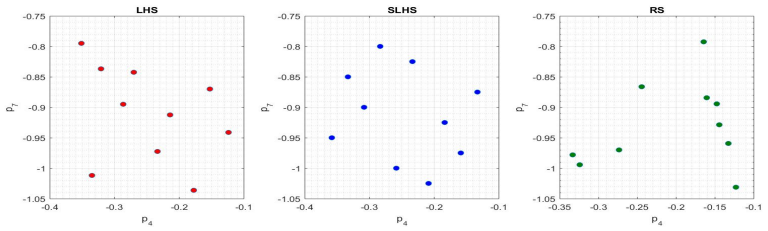


**Fig. 1** Comparison of various sampling techniques. SurrogateModel

We discuss the utility of these sampling techniques, and how they are used for surrogate modelling in the next sections. The deeper questions of sampling that relate to the choice of surrogate model, nature of the objective function and analysis are left for the reader to explore from recommended sources such as [19].

### 3.2.2 Model order reduction

The overarching goal of POD method is to provide a fit of the desired data by extracting interpolation functions from the information available in the data set. Geometrically, it derives proper orthogonal modes by projecting the original model

onto the reduced space spanned by the POD modes [19]. A simple mathematical formulation of POD technique is laid out in this subsection which closely follow the references [8, 10, 26].

Suppose that we wish to approximate the response of the system given by output parameters $y \in \mathbb{R}^m$, where $m = n_y \times n_t$, using the set of input parameters $b \subset \mathbb{R}^{n_u}$ over a certain domain $\Omega$. The ROMs approximate the state function y(t) in domain $\Omega$ using linear combination of some basis function $\phi^i(x)$ such that

$$y(t) \approx \sum_{i=1}^{M} a_i . \phi^i(t) \tag{7}$$

where, $a_i$ are unknown amplitudes of the expansions and $t$ is the temporal coordinate. The first step in this process would be to find the basis. Once the basis function is chosen, the amplitudes can be easily determined by a minimization process. It is ideal to take orthonormal set as the basis with the property

$$\int_{\Omega} \phi_{k_1}(t) \, . \, \phi_{k_2}(t) \, dx = \begin{cases} 1 & k_1 = k_2 \\ 0 & k_1 \neq k_2 \end{cases} \tag{8}$$

This way, the determination of the amplitudes $a_k$ only depends on function $\phi_k^i(t)$ and not on any other $\phi$. Along with being orthonormal, the basis should approximate the function in best possible way in terms of the least square error. These ordered orthogonal functions are called the POMS for the function $y(t)$ and the equation (7) is called the POD of $y(t)$.

In order to determine the number of POMs that should be used in approximation of lower dimensional space, we use the idea that POD inherently orders the basis elements by their relative importance. This idea is used very often in statistics with singular value decomposition of the matrices. Since the theory of SVD is so widespread, we only highlight the most general and relevant details of SVD that are helpful in derivation of POMs and POD basis.

There prevails a misconception amongst researchers about distinction between SVD and POD. As opposed to the common understanding, POD and SVD are not strictly the same: the former is a model reduction technique where as the latter is merely a method of calculating the orthogonal basis.

In general, SVD is a technique that is used to decompose any real rectangular matrix $Y$ into three matrices, $U$, $\Sigma$ and $V$, where $U$ and $V$ are orthogonal matrices, $\Sigma$ is a diagonal matrix that contains the singular values $\sigma_i$ of $Y$, sorted in a decreasing order such that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_d \geq 0$, and $d$ is the number of non-zero singular values of $Y$.

The singular values can then be used as a guide to determine the POD basis. If a $k$-dimensional approximation of original surface is required, where the rank $k<d$. The first $k$ columns of the matrix $U$ serve as the basis $\phi^i, i = 1, ..., k$. These set of columns, gathered in matrix $\Phi$, form an orthonormal set of basis for our new low-dimensional surface.

The relative magnitude of each singular value with respect to all the others give a measure of importance of the corresponding eigenfunction in representing elements of the input collection. Based on the same idea, a common approach for selection of number of POMs ($k$) is to set a desired error margin $\epsilon_{\text{POD}}$ for the problem under consideration and choose $k$ as a minimum integer such that the cumulative energy *E(k)* captured by first $k$ singular values (now POMs) is less than 1-$\epsilon_{\text{POD}}$, i.e.

$$E(k) = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{d} \sigma_i^2} \leq 1 - \epsilon_{\text{POD}}^2 \tag{9}$$

After collection of basis using SVD, it is easy to calculate the matrix of amplitudes $A_k$. Let $\Sigma_k = [\sigma_1, \sigma_2, ..., \sigma_k]$ be the set of $k$ largest singular values of our initial matrix $Y$, then, the matrix of amplitudes is given by $Y_k = \Sigma_k A_k,\ A_k = \Sigma_k^\top Y_k$.

With the basis vectors and amplitude matrix, using POD discrete theory, low dimensional approximation of our problem has been constructed. However, the formulation is not very useful since our new model can only give the responses of the system for a discrete number of parameter combinations (those that were previously used to generate the snapshot matrix). Since, in many practical applications (for optimization and inverse analysis), even though the values of input parameters may sometime fall in a particular range, the parameter values are not known a-priori and can assume any arbitrary value between those ranges. Therefore, we take a step further to approximate the newly constructed model. We combine POD with RBF interpolation to create low-order parameterization of high-order systems for accurate prediction of system responses.

RBF is a unique interpolation technique that determines one continuous function defined over the whole domain. It is a widely used for smoothing and multidimensional approximation. Let $Y_k$ be the reduced dimensional matrix. For better approximation of the surrogate model, we want to find a continuous function $f(b) = y$, where $b$ is the vectors of some parameters and $y$ is the system response. It can be achieved easily by applying RBF to reduced dimensional space where system responses are expressed as amplitudes in the matrix $A_k$. Hence,

$$f(b) = y = \Sigma_k A_k = \Sigma_k f_a(b) = \phi f_a(b) \tag{10}$$

When RBF is applied for the approximation of $f_a$, $f_a$ is written as linear combination of some basis functions $g_i$ such that

$$f_a(b) = \begin{bmatrix} a_1^i \\ a_2^i \\ . \\ . \\ . \\ a_K^i \end{bmatrix} = \begin{bmatrix} d_{11} \\ d_{21} \\ . \\ . \\ . \\ d_{K_1} \end{bmatrix} .g_1(b) + \begin{bmatrix} d_{12} \\ d_{22} \\ . \\ . \\ . \\ d_{K_2} \end{bmatrix} .g_2(b) + ... + \begin{bmatrix} d_{1N} \\ d_{2N} \\ . \\ . \\ . \\ d_{K_N} \end{bmatrix} .g_N(b) = D.g(b) \tag{11}$$

In this work, we use linear and cubic spline RBF for analysis, given by:

$$\text{linear spline}: \ g_j(b) = ||b - b_j||; \quad \text{cubic spline}: \ g_j(b) = ||b - b_j||^3; \tag{12}$$

Once the basis functions $g_i$ are known, the aim is to solve for the interpolation coefficients that are collectively stored in matrix $B$. Since we already have the value of amplitudes $A$ from last step, matrix $B$ can be easily obtained by using the equation $B = G^{-1}A$. Finally, using equation (10), our initial space $y$ can be approximated by:

$$y \approx \Phi.D.g(b) = \hat{y} \tag{13}$$

Since matrix $\Phi$ and $D$ are calculated once for all, one only needs to compute the vector $g(b)$ for any arbitrary combination of parameters to obtain system responses.

We have constructed surrogate model using POD and RBF to calculate the value of functionals $\hat{\psi}_j(b) = \tilde{\psi}_j(b, \hat{y}), j = 0, 1, \ldots, m$. The formulation of optimal control problem for surrogate models is to find a vector $\hat{b}^*$ such that:

$$\begin{gathered}
\hat{b}^* \in U_\partial, \hat{\psi}_0\left(\hat{b}^*\right) = \min_{b \in U_\partial} \hat{\psi}_0(b) \\
U_\partial = \left\{b : b \in U_b, \hat{\psi}_j(b) = 0, j = 1, \ldots, m_1; \hat{\psi}_j(b) \leq 0, j = m_1 + 1, \ldots, m\right\} \\
U_b = \{b : b \in \mathbb{R}^n, b^- \leq b \leq b^+, b^- \in \mathbb{R}^n, b^+ \in \mathbb{R}^n\} \\
\hat{y} = S(b) \\
\hat{\psi}_j(b) = \tilde{\psi}_j(u(b), \hat{y}), j = 0, 1, \ldots, m
\end{gathered} \tag{14}$$

Replacing the state equation (2) with surrogate model given in equation (13) can decrease the computational time by a significant amount. It is free of the complexity of initial problem and involves matrix multiplication that can be accomplished in a much smaller time than solving ordinary differential equations with high fidelity methods. We test this hypothesis by comparing the accuracy of system responses and time of calculation for both equation (2) and equation (13) for real life examples in section 5.

We evaluate the accuracy of responses by generating $n_g$ test points for the set of parameters P, using the same sampling technique that had been used for generation of training test. For these new points, we calculate the system responses $Y_g = [y_1, y_2, ..., y_{n_g}] \in \mathbb{R}^{m \times n_g}$ using initial numerical method (that solves entire system), and newly constructed surrogate model $\hat{Y}_g = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_{n_g}] \in \mathbb{R}^{m \times n_g}$. We use relative maximum absolute error (RMAE) to determine the accuracy of the optimization results given by:

$$\text{RMAE} = \max_{1 \leq i \leq m} \max_{1 \leq j \leq n_g} \frac{|y_{ji} - \hat{y}_{ji}|}{y_{ji}} \tag{15}$$

## 4 Enhanced Surrogate Models

While POD-RBF surrogate models presented in 3 are usually accurate, sometimes the desired accuracy of the model is not achieved. This usually occurs because either the initial sampling does not truly represent the true behaviour of the dynamical system, or system's optimal values occur at the corner points and the predictive models in general tend to perform poorly on extreme ends. One common approach to overcome this issue is to use adaptive sampling, a method that has been used by many researchers such as [19] to find optimal design space points. In [13], the authors compare the adaptive sampling technique with uniform and random sampling and show the effectiveness of this approach in generating enough data points in the most plausible regions. However, this technique involves solving a series of point placement non-linear programming problems for optimization and the repetitive nature of this procedure can still have a high computational cost.

In this paper, we deal with the sampling issue in two ways— we first propose to use non-random, memory based sampling techniques that assist in diversifying the training sample, as discussed in the previous section. Secondly, we develop an iterative algorithm which recursively shifts the domain of training to direct the surrogate model towards finding the true optimal solutions. The combination of these approaches result in highly accurate solutions as demonstrated in the next section.

The choice of sampling techniques obviously affects the accuracy of the surrogate model. Therefore, we would like to compare the various sampling methodologies discussed in section 3. We combine each sampling technique with varying number of training points and various interpolation techniques, and treat it as a separate experimental designs in our research. We compare these experimental designs to determine which setup results in the highest accuracy while satisfying the time constraints for generation of the snapshots.

The algorithm for constructing surrogate models can be divided into three parts: experimental design, offline phase and online phase. The offline phase (training of the model) entails utilization of sampling techniques to generate data, computation of snapshot matrix of model solutions, obtainment of proper orthogonal modes via singular value decomposition and estimation of POD expansion coefficients that approximate the POD basis via RBFs. Then comes a pseudo-testing phase (testing to find the best experimental design) where the surrogate models from the offline phase are used to solve the dynamical system and the overall error of approximation is used to select the best experimental design.

Note that until this point, only the system of ODEs is solved, and the accuracy of the original and surrogate system responses is compared to decide the best experimental design. Once the best sampling strategy and corresponding number of training points, and interpolation technique are decided, the algorithm enters in its online phase (testing phase) in which the surrogate model responses are used to solve the optimal control problem and the error between true and surrogate optimal values is calculated. If the error exceeds the given threshold, the iterative algorithm is activated.

The iterations caters to the aforementioned low-accuracy issue in two ways. Firstly, it trains the initial model with the sampling points from a slightly wider domain than the domain in which the optimization is performed. This way, the corner points are incorporated into the sampling space and surrogate model tends to provide better approximation for the optimal points. Secondly, in order to minimize the error of approximation, the algorithm allows to decrease the width of domain of control parameters at each iteration. By decreasing the size of sampling space, the sampling points move closer to each other. Even if the corner points are not accounted for in the sampling design, the smallest distance between the corner and the neighboring points is lower in smaller domain, hence resulting in better approximation and higher accuracy. The iterative algorithm becomes active every time the error of approximation is higher than the tolerance level.

## 4.1 Iterative Algorithm

The iterative process can be summarized in four steps:

1. Initialization: In this step, the parameters of algorithm are initialized. This includes width (the length of domains of control parameters), desired tolerance level and $b^{(0)}$ = initial guess for $b$ (the optimization parameters)
2. Setting up the bounds: In this step, upper and lower bounds of domain are defined for each control parameter. It is done by taking $b^{(0)}$, interpolating it and substituting it as the value of control variables in our problem. Next, the new bounds are created centered at $b^{(0)}$. The width of domain for each subsequent iteration is lower than the previous iteration. The value of $b^{(0)}$ is replaced with optimal value of $b$ obtained using surrogate model ($\hat{b}^*$) in the previous iteration. Finally, it is checked if the new bounds are within the bounds that were defined at the beginning of the problem. If not, the algorithm restricts them from exceeding the initial bounds. This step of the iterative process is depicted for two optimization parameters in figure 2.
3. Optimization: This is the main step of the algorithm which includes training and testing phases. In summary, we create sampling set and snapshots, construct the surrogate model, solve optimization problem and calculate the error.
4. Updating parameters: This step prepares the parameters for the next iteration in the case when the tolerance level falls below the error of approximation. In general, the algorithm replaces $b^{(0)}$ with the optimized value of $\hat{b}^*$ from the surrogate response of current iteration, shortens the length by using a predefined multiplier. If the tolerance criteria is met, the iterative process stops. Else the algorithm resumes from step 2.

The iterative algorithm discussed throughout this section is summarized in flowchart presented in the figure 3.

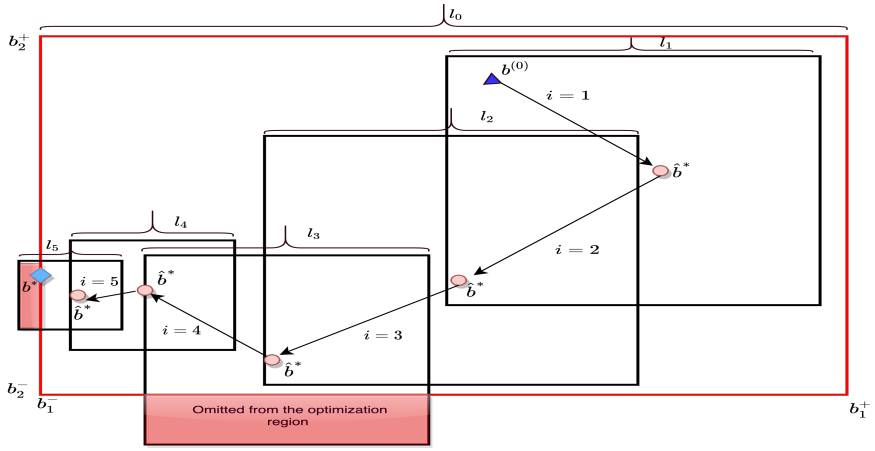


**Fig. 2** Example of iterative algorithm of two optimization parameters $b_1$ and $b_2$ with iterations $i = 1, \ldots, 5$ and recursively decreasing lengths $l_i, i = 1, \ldots, 5$

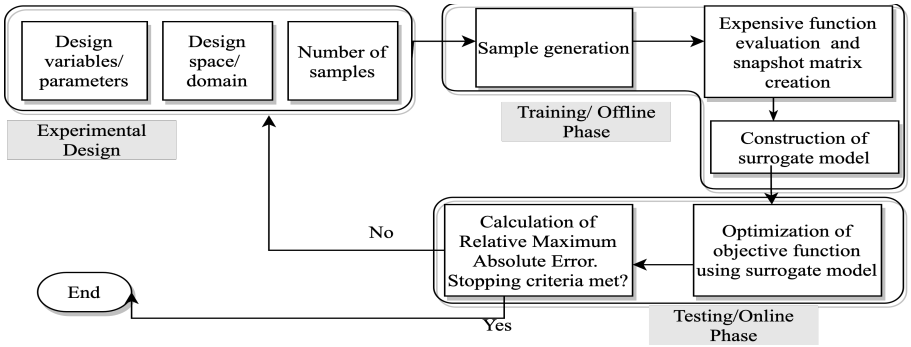


**Fig. 3** POD-RBF algorithm flowchart

## 5 Application of POD-RBF Procedure on Dynamical Systems

In this section, the POD-RBF procedure is used to construct the surrogate models for real-life dynamical systems and associated optimal control problems are solved. Three dynamical systems with various complexity are presented, with model 1 being the simple non-linear ODE problem, and model 2 and 3 featuring a non-linear system of equations with complex optimization criteria. For each model, a description of the problem is presented and the values of initial parameters used in numerical experiments are defined. Next, the numerical experiments are performed to first decide the combination of sampling technique, interpolation method and sampling

points optimal for that model and then the optimization problem is solved to evaluate the accuracy of surrogate responses and the difference in computational time of optimization with original and POD-RBF methods.

As a convention for this section, the variables with the hat operator $\hat{}$ represent the results obtained using surrogate model and without hat stand for the results from original model. The description of common variable names are summarized in table 1.

**Table 1** Details of notations used in preceding analysis

| Notation | Description |
|---|---|
| $b^{(0)}$ | Initial value of optimization parameter |
| $\hat{b}^*$ | Optimal value of optimization parameter, surrogate model |
| $b^*$ | Optimal value of optimization parameter, original model |
| $\psi_0(b^{(0)})$ | Value of optimization criteria for $b^{(0)}$, original model |
| $\psi_0(\hat{b}^*)$ | Value of optimization criteria for $\hat{b}^*$, original model |
| $\widehat{\psi_0}(\hat{b}^*)$ | Value of optimization criteria for $\hat{b}^*$, surrogate model |
| $\psi_0(b^*)$ | Value of optimization criteria for $b^*$, original model |
| $\psi_i(b^{(0)})$ | Value of $i^{th}$ optimization constraint for $b^{(0)}$, original model |
| $\psi_i(\hat{b}^*)$ | Value of $i^{th}$ optimization constraint for $\hat{b}^*$, original model |
| $\widehat{\psi_i}(\hat{b}^*)$ | Value of $i^{th}$ optimization constraint for $\hat{b}^*$, surrogate model |
| $\psi_i(b^*)$ | Value of $i^{th}$ optimization constraint for $b^*$, original model |

## 5.1 Model 1: Science Policy

### 5.1.1 Description of the Model

This subsection features a very interesting application of optimal control theory in economics. The problem is one of the oldest optimal control problem in economics known as science policy and was originally introduced in 1966 by M.D. Intriligator and B.L.R. Smith in their paper "Some Aspects of the Allocation of Scientific Effort between Teaching and Research" [18]. Science policy addresses the important issue of allocation of new scientists between teaching and research staff, in order to maintain the strength of educational processes or alternatively, avoiding any other dangers caused by inappropriate allocation between scientific careers [17]. In order to find the optimal allocation, the optimal control problem was formulated as following:

$$\max_{(u,y)\in U\times Y} \tilde{\psi_0} = \int_{t_0}^{\top} [0.5y_1(t) + 0.5y_2(t)]dt, \tag{16}$$

subject to

$$c(y,u) = 0 \sim \begin{cases} y_1'(t) - u(t)gy_1(t) + \delta y_1(t) = 0, t \in [t_0, T] \\ y_2'(t) - (1-u(t))gy_1(t) + \delta y_2(t) = 0 \\ y_1(t_0) - y_{10} = 0, y_2(t_0) - y_{20} = 0 \end{cases}$$

$$\begin{bmatrix} \tilde{\psi_1} \\ \tilde{\psi_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \sim \begin{cases} y_1(T) - y_{1T} = 0 \\ y_2(T) - y_{2T} = 0 \end{cases}$$

$$u^- \leq u(t) \leq u^+$$

In this formulation, the state variable $y_1$ and $y_2$ represent the teaching scientists and research scientists respectively at any given time $t$. The detailed description of all the parameters and their values are summarized in table 2. The control variable $u$ represents the number of new scientists becoming teachers, correspondingly $(1-u)$ represents the number of new researchers. Hence, the differential equations determine the rate of change of number of teachers and researchers by subtracting the new proportion from the allocated proportion. The upper and lower limit of control function indicate the limits of the science policy in affecting the initial career choices, by government contracts, grants, incentive schemes, etc.

**Table 2** Description of parameters for Model 1

| Parameters | Definitions | Values |
|---|---|---|
| $u(t_0)$ | Proportion of new scientists becoming teachers at initial time | 0.5 |
| $g$ | Number of scientists annually produced by one scientist | 0.14 |
| $\delta$ | Rate of exit of scientists due to death, retirement or transfer | 0.02 |
| $y_{10}$ | Number of initial scientists working as teachers | 100 |
| $y_{20}$ | Number of initial scientists working as researchers | 80 |
| $T$ | Final time for the analysis in this policy | 15 |
| $y_{1T}$ | Number of final scientists working as teachers | 200 |
| $y_{2T}$ | Number of final scientists working as researchers | 240 |
| $u^-$ | Lower limit of control function | 0.1 |
| $u^+$ | Upper limit of control function | 0.6 |

The problem is the one of choosing a trajectory for the allocation of $u(t)$ such that the welfare is maximized, given by the objective function in equation (16). The terminal part $g_1(.,.)$ of welfare is not accounted for in the objective function, but the state constraints are added to compensate for it in the form of $y_1(T) - y_{1T} = 0$ and $y_2(T) - y_{2T} = 0$. The optimization process is focused at maximizing the intermediate value $g_2(.,.,.)$ of welfare. The welfare function is thought to be additive of individual utilities along the lines of utilitarian approach. The utilities are set as a linear function, with an assumption that the teachers and researchers are perfect substitutes, and the allocation of any scientist to one career will lead him to abandon the other career completely. This assumption, even though unrealistic, is granted for simplicity and can be complicated at the later stages.

### 5.1.2 Simulation

This system of equations is solved for $n_s = 40, 60, 80$ training points, generated with LHS, SLHS and RS to create the snapshot matrix. The desired tolerance level is $\epsilon_{\text{POD}} = 0.01$. The singular value plot for one specific experimental design, SLHS and $n_s = 40$ is presented figure 4 and shows that the first 4 singular values explain almost 100% variance. The plots of singular values for other experimental designs depicted similar pattern. Given the criterion in equation (9), we choose the rank of $k = 4$. It can be clearly noticed that the magnitude of all the singular values is very small compared to first singular value; the relative commutative energy $E(i)$ of first singular value is more than 99%. This shows that that the responses of the system are fully correlated. Hence, rank 4 approximation is enough and adding more vectors in approximation (by increasing the rank) will not improve the precision a lot.

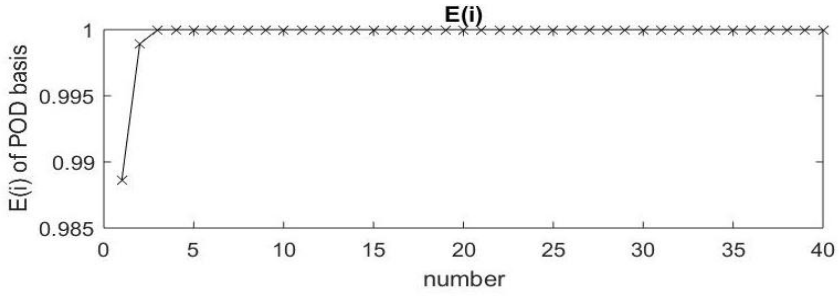


**Fig. 4** Cumulative energy plot to determine singular values for Model 1. Surrogate-Model_SciencePolicy

Next, the surrogate model is constructed for each of the variant with this rank and the RMAE are reported in table 3. The table shows that the lowest RMAE was obtained for LHS, followed by SLHS and the RS. As the theory suggests, RMAE is observed to decrease with increasing number of sampling points with an exception of cubic spline in random sampling. The anomalous behavior of RS can be associated with its randomness, which sometimes generates the sampling points which belong to only one region of the surface, leading to higher variance in the model and higher error of approximation, even with increasing number of training points. Another trend that can be consistently observed is that the linear spline RBF tend to perform better than the cubic spline in this model. Overall, the best experimental design for this model is to use a combination of LHS with linear spline RBF and $n_s = 80$. The surrogate model approximation for the initial control value $u = 0.5$ and the original system response are plotted in figure 5 and show that the approximated responses are very close to the actual responses.

**Table 3** RMAE for various experimental designs of Model 1

| **Sampling** | **Interpolation** | $n_s = 40$ | $n_s = 60$ | $n_s = 80$ |
|---|---|---|---|---|
| LHS | Linear | 0.02034 | 0.00293 | 0.00150 |
| | Cubic | 0.05316 | 0.00647 | 0.00641 |
| SLHS | Linear | 0.03825 | 0.00679 | 0.00437 |
| | Cubic | 0.05175 | 0.00897 | 0.00861 |
| RS | Linear | 0.01525 | 0.02410 | 0.02792 |
| | Cubic | 0.16457 | 0.26597 | 12.91601 |

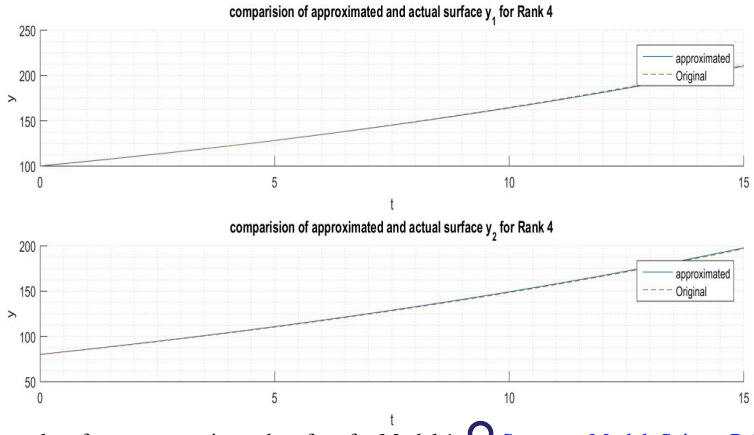


**Fig. 5** Actual surface vs approximated surface for Model 1. SurrogateModel_SciencePolicy

### 5.1.3 Optimization

For the final step of analysis, the surrogate model was constructed with 40 training points, LHS, and linear spline RBF. Here, we use $n_s = 40$ because given the simplicity of the problem, the accuracy required for optimization can be achieved by small number of training points. The optimization problem is solved with two optimization parameters for control function using both original and surrogate model. The results of optimization are given in table 4. The problem started with equal number of scientists allocated in both careers, with the initial value of state constraint $\psi_1(b^{(0)}) = [11.8001; 43.0163]$ representing that the number of teachers and researchers allocated at initial time were 11 and 43 units short of $y_{1T}$ and $y_{2T}$ respectively. The solution to the problem allocates around 52% of new scientists to teaching at the beginning of the time. This proportion decreases as the time passes with around 47% scientists allocated as teaching staff at the end of time (see figure 6(b)). The optimal surface in 6(a)) shows that the number of teaching staff is allocated to be higher than the number of researchers until the end time. The surrogate model gave consistent results, with error of approximation (the relative error of $\psi_0(\hat{b}^*)$ and $\widetilde{\psi_0}(\hat{b}^*)$) as low as 0.005 in the first iteration.

Even though the optimization using surrogate model is slightly quicker than the original model, the time taken for construction of surrogate model is higher. Hence, despite of highly accurate system responses through surrogate model, substituting original model with POD-RBF model might not be useful, as the time taken for optimization by surrogate model (training + optimization) takes much longer than the original model. This example give us insight into why surrogate modelling was avoided into applications earlier: the simple nature of optimization models for some applications do not require high computational resources, while the construction of surrogate models is much more computationally expensive and may not be desirable.

**Table 4** Optimization results of Model 1

| Field | Value | Field | Value |
|---|---|---|---|
| $b^{(0)}$ | [0.5000 0.5000] | Bounds | [0.1000,0.6000] |
| $b^*$ | [0.6000,0.3461] | $\hat{b}^*$ | [0.5187,0.4730] |
| $\psi_0(b^{(0)})$ | 210.6500 | $\psi_0(\hat{b}^*)$ | 209.7600 |
| $\psi_0(b^*)$ | 212.8400 | $\widehat{\psi_0}(\hat{b}^*)$ | 210.9900 |
| $\psi_1(b^{(0)})$ | $[11.8001, 43.0163]^\top$ | $\psi_1(\hat{b}^*)$ | $[0.0003, 0.0014]^\top$ |
| $\psi_1(b^*)$ | $[0.000, 0.000]^\top$ | $\widehat{\psi_1}(\hat{b}^*)$ | $[0.0000, 0.0023]^\top$ |
| $\text{Time}_{orig}$ | 2.8109 sec | $\text{Time}_{surr}$ | 2.3694 sec |
| $\text{Time}_{cnstr}$ | 37.8406 sec | $\epsilon$ | 0.0058 |

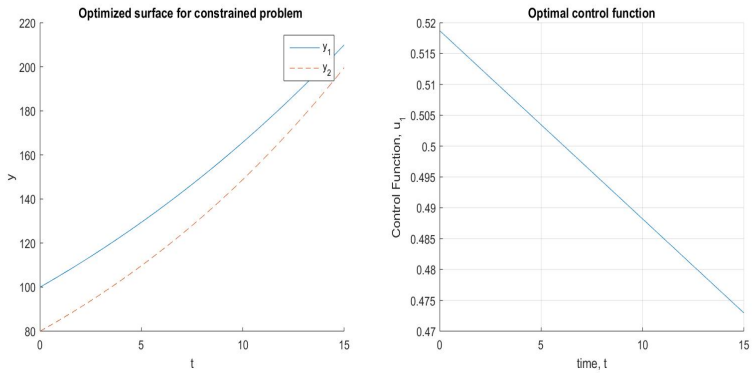


**Fig. 6** Optimal surface and control functions for Model 1. SurrogateModel_SciencePolicy

## 5.2 Model 2: Population Dynamics

### 5.2.1 Description of the Model

In this subsection, a more complex application of optimal control theory is presented with a general model of non-linear system of ODEs defined by:

$$c(y,u)=0 \sim \begin{cases} \begin{cases} y_1' - p_1 y_1 - p_2 y_2^2 - u_1 y_1 F\left(y_1,t\right) y_2 = 0, \\ y_2' - p_3 y_2 - p_4 y_2^2 - u_1 u_2 y_1 F\left(y_1,t\right) y_2 = 0, \end{cases} t \in \Omega_t = (t_0, T] \\ y_1\left(t_0\right) - y_{10} = 0 \\ y_1\left(t_0\right) - y_{20} = 0 \\ F\left(y_1,t\right) = 1 - e^{-p_5 y_1} \end{cases} \tag{17}$$

These type of dynamical problems are usually observed in population dynamics in biology, ecology and environmental economics. These problems are variation of prey-predator model presented by Lotka-Volterra. This subsection aims at generalizing the approach of POD-RBF on these non-linear models without providing specific details of the model parameters of the optimization problem.

The optimization problem considered here consists of finding a value of control function $u^* = \left[u_1^*, u_2^*\right]$ that minimizes the distance between $y_1$ and its desirable value $y_{1d}$ Value on control function is restricted by dual pointwise constraints and value $y_2$ do not exceed maximum value $y_{2d}$. The optimization problem can be formulated in the following manner: find $u^*$ that minimize optimization criterion

$$\psi_0\left(u^*\right) = \min_u \int_{t_0}^{T} \left(y_1(t,u) - y_{1d}\right)^2 dt \tag{18}$$

subject to state equation (17), box constraints on the control

$$U = \left\{u : u^-(t) \le u(t) \le u^+(t)\right\} \tag{19}$$

and pointwise constraint on state

$$y_2(t) \le y_2^+ \tag{20}$$

The pointwise state constraint (20) is transformed into an equivalent equality constraint of the integral type

$$\psi_1(u) = \tilde{\psi}_1(u, y(u)) = \int_{t_0}^{T} \left(\left|y_2(t,u) - y_{2d}\right| + y_2(t,u) - y_{2d}\right)^2 dt \tag{21}$$

Taking into account equations(18-21) the optimization problem can be written in a reduced form as follows:

$$\begin{aligned} &\psi_0(u^*) = \min_{u \in U_\partial} \int_{t_0}^{T} (y_1(t,u) - y_{1d})^2 \, dt \\ &U_{\partial u} = \{u : u \in U; \psi_1(u) = \tilde{\psi}_j(u, y(u)) = 0\} \\ &c(y(u), u) = 0 \end{aligned} \tag{22}$$

### 5.2.2 Simulation

For numerical experiments we select the following values for the input parameters: $[p_1, p_2, p_3, p_4, p_5] = [0.734, 0.175, -0.500, -0.246, 0.635]$, $[t_0, T] = [0, 10]$, $n_u = 2, u^- = \left[u_1^-, u_2^-\right] = [-0.5500, -1.0370], u^+ = \left[u_1^+, u_2^+\right] = [-0.300, -0.7870]$, $y_{1d} = 5, y_2^+ = 6$. The control functions $u_1(t), u_2(t)$ on the interval $[t_0, T]$ are approximated by linear functions. Thus, the vector of optimization parameters $b$ consist of four components: $b = \left[b_1^{(1)}, b_2^{(1)}, b_1^{(2)}, b_2^{(2)}\right]^T = [b_1, b_2, b_3, b_4]^T$.

For numerical simulations, LHS, SLHS and RS are used to define the sampling matrix with $n_s = 40, 60$ and 80. Also, RBF interpolation-linear spline and cubic spline is used for comparison of results. The solution $y = [y_1, y_2]$ where $n_y = 2$ is then computed for time instances, $t_i$ with $t_0 < t_i < t_{n_t}$, $n_t = 100$ equally spaced instances of t, and $n_s$ sampling points, and then system responses were collected to generate the snapshot matrix. The error of approximation is fixed $\epsilon_{\text{POD}} = 0.01$.

Next, the POD-RBF approach is applied to this model to first determine the dimension of POD basis through SVD using cumulative energy method (it is done for all experimental designs) and it is concluded that 3 singular values should be considered.

Having chosen $k = 3$, the numerical simulations are performed for model 2 given in equation (17). For testing of the model, $n_g = 10$ points were used to calculate the RMAE for each combination. Table 5 exhibits that among all the surrogate models that were trained using different number of sample points, different sampling techniques and RBF interpolations, the cubic spline RBF showed the lowest error for both LHS and SLHS in general, with a few exceptions. Also, as expected, the error of approximation shows a decreasing pattern as the number of sample points increase from 60 to 80, except in RS when the RMAE follows no particular trend. The least RMAE is obtained for the model trained on 80 data points from SLHS for cubic spline RBF. For one of such sample point $b = [-0.425, -0.425, -0.912, -0.912]$, the POD-RBF responses were obtained for $n_s = 40$ and the original and approximated $y_1$ and $y_2$ were plotted as shown in figure 7. For this point, all POD-RBF gave relative maximum absolute error less than 1% as desired.

### 5.2.3 Optimization

In previous subsubsection, the best results were obtained for $n_s = 80$ with SLHS and cubic spline RBF. That experimental design is used to solve the optimization problem (22) and the results are summarized in table 6. For simplicity, the number

**Table 5** RMAE for various experimental designs of Model 2

| **Sampling** | **Interpolation** | $n_s = 40$ | $n_s = 60$ | $n_s = 80$ |
|---|---|---|---|---|
| LHS | Linear | 0.45112 | 0.32948 | 0.18871 |
| | Cubic | 0.28229 | 0.24010 | 0.15794 |
| SLHS | Linear | 0.26162 | 0.19198 | 0.19204 |
| | Cubic | 0.23986 | 0.18685 | 0.15376 |
| RS | Linear | 0.59500 | 0.55080 | 0.86405 |
| | Cubic | 0.92109 | 0.15595 | 0.19902 |

of optimization parameters for each control variable are taken to be 2. We could, however, allows specification of different number of optimization parameters for each control variable. The optimization results of this model apparently highlight the efficiency of surrogate modeling. As the table 6 reports, the tolerance level is met in the first iteration, with error between approximated and actual responses being less than 0.01 in first iteration. Hence, the desired accuracy is achieved and no further iterations were required.

The optimization criteria obtained using surrogate model $\widehat{\psi_0}(\hat{b}^*) = 43.5647$ is very close to $\psi_0(b^*) = 43.3287$. Moreover, since results of optimization problem were obtained within one iteration, the construction time of surrogate model can be considered once for all. Therefore, the total computational time for optimization through surrogate model of 6.6 seconds + 15.35 seconds is less than 23.40 seconds taken by original problem. Relatively, the surrogate method is four times faster than the original method in solving optimization problem. In a nutshell, for this highly non-linear model, surrogate model gave highly accurate and computationally efficient result of the optimization problem.

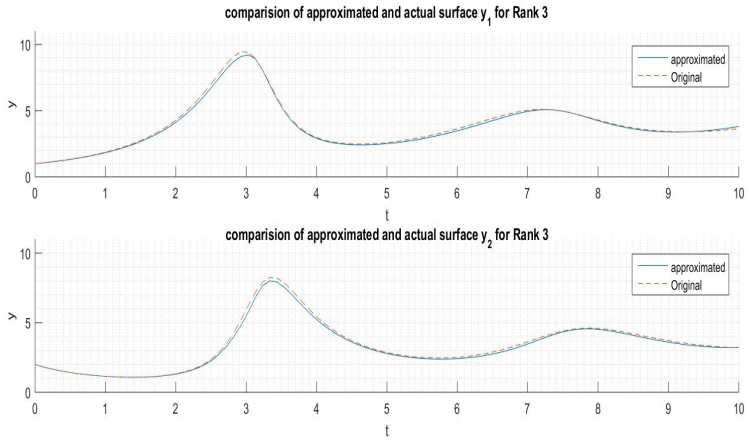


**Fig. 7** Actual vs approximated surface of Model 2. SurrogateModel_PopulationDynamics

**Table 6** Optimization results of Model 2

| Field | Value | Field | Value |
|---|---|---|---|
| $b^{(0)}$ | [-0.4250,-0.4250, -0.9120,-0.9120] | Bounds | [-0.5500, -0.300]; [-1.0370,-0.7870] |
| $b^*$ | [-0.5006,-0.3250, -1.0120,-1.0120] | $\hat{b}^*$ | [-0.4922,-0.3334, -1.0120,-1.0120] |
| $\psi_0(b^{(0)})$ | 55.2817 | $\psi_0(\hat{b}^*)$ | 43.9127 |
| $\psi_0(b^*)$ | 43.3287 | $\widehat{\psi_0}(\hat{b}^*)$ | 43.5647 |
| $\psi_1(b^{(0)})$ | 22.9396 | $\psi_1(\hat{b}^*)$ | 0.0162 |
| $\psi_1(b^*)$ | 0.0000 | $\widehat{\psi_1}(\hat{b}^*)$ | 0.0000 |
| $\text{Time}_{orig}$ | 23.3983 sec | $\text{Time}_{surr}$ | 6.6241 sec |
| $\text{Time}_{cnstr}$ | 15.3470 sec | $\epsilon$ | 0.0081 |

## 5.3 Model 3: Quality Control in Production and Process Management

### 5.3.1 Description of the Model

The third model in this series is that of optimal control strategy in production and process management. This non-linear dynamical system is taken from work of M.D.

Haider Ali Biswas [6]. The model addresses the issue of declining quality of goods in production processes over time by the help of state constraints in optimization problem. Even though the model is introduced in the context of industrial engineering in the original paper, the model in equally valid in industrial economics since the quality of products is one of the main factors in determining the aggregate demand and supply of each firm and it comes to economists to keep check of the market demand by controlling the quality of products. The mathematical model of the problem taken directly from [6] is as following:

$$
Model\ 3 := \begin{cases}
\max\limits_{(u_1,u_2)\in U} J = l(y(T)) + \int_0^T L(t, y(t), u(t)) \\
\text{subject to} \\
y_1'(t) = y_2(t)u_1(t) - d(t), \\
y_2'(t) = -(\alpha + u_2(t))y_2(t) + u_2(t), \\
g(t, x(t)) \leq 0, \forall t \in [0, T], \\
(u_1(t), u_2(t)) \in U \text{ a.e. } t \in [0, T], \\
y_1(t) \geq 0, \text{ a.e. } t \in [0, T], \\
y(0) = y_0, \\
\text{where} \\
g(t, y(t)) = -y_2(t) + 0.5, \\
l(y(T)) = by_2(T)e^{-\rho T}, \\
L(t, y(t), u(t)) = (wd - hy_1(t) - ru_1^2(t) - cu_2(t))e^{-\rho t} \\
U = \{(u_1, u_2), 0 \leq u_1(t) \leq U_1, 0 \leq u_2(t) \leq U_2, \text{ a.e. } t \in [0, T]\}
\end{cases}
\tag{23}
$$

In this model, the state variables $y_1(t)$ and $y_2(t)$ represent the inventory level, and proportion of 'good'(appropriate quality) from the end items at any time $t$ respectively. At initial time, $y_{10} = 3$ and $y_{20} = 1$. The control function $u_1(t)$ represented the scheduled production rate with upper limit $U_1 = 3$ and control function $u_2(t)$ represents the preventive maintenance rate to reduce the proportion of defective units produced with an upper limit of $U_2 = 4$. The demand rate ($d$) and the obsolescence rate of process performance in absence of maintenance ($\alpha$) are fixed at $d = 4$ and $\alpha = 2$. The negative sign associated with the second differential equation in the dynamical system corresponds to the declining proportion of 'good' items in absence of maintenance. The maintenance is introduced in the optimization problem by declaration of lower bound on proportion of good items in form of state constraint. The final objective of the problem is to minimize the salvage cost (the estimated resale value of a good at the end of its useful life).

### 5.3.2 Simulation

The positive constants in equation (23) are: $[\rho, w, h, c, b, r] = [0.1, 8, 1, 2.5, 10, 2]$. We carry out similar analysis for this model. The rank determined using the energy method is $k = 4$ for all variants. The RMAE results as recorded in table 7 show that once again cubic spline RBF dominated linear spline in higher number of training points. Also, the least error of approximation was obtained when the combination of SLHS, cubic spline RBF, and $n_s = 80$. The approximation of system responses for an arbitrary point and its comparison with original system responses is displayed in figure 8.

**Table 7** RMAE for various experimental designs of Model 3

| Sampling Strategy | Interpolation Type | $n_s = 40$ | $n_s = 60$ | $n_s = 80$ |
|---|---|---|---|---|
| LHS | Linear | 0.08475 | 0.00886 | 0.00813 |
| | Cubic | 0.03809 | 0.00567 | 0.00697 |
| SLHS | Linear | 0.01432 | 0.02622 | 0.00212 |
| | Cubic | 0.02214 | 0.00586 | 0.00157 |
| RS | Linear | 0.03245 | 0.06733 | 0.06917 |
| | Cubic | 0.06310 | 0.00377 | 0.06124 |

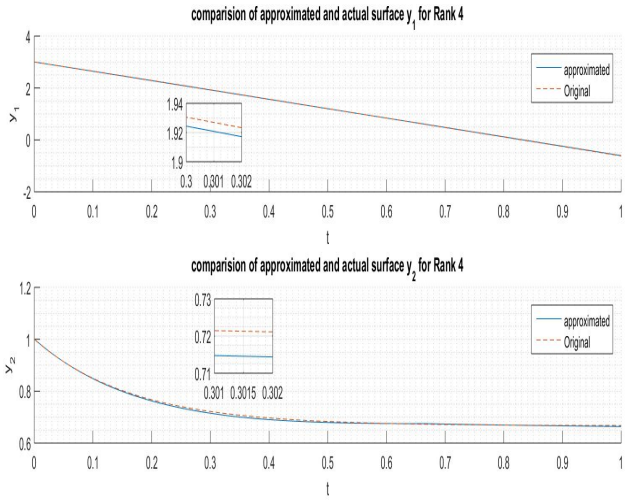


**Fig. 8** Actual surface vs approximated surface for Model 3. SurrogateModel_Production

### 5.3.3 Optimization

The experimental design concluded in previous subsection was used to solve the optimization problem of Model 3 with 4 optimization parameters for each control function. Piecewise-linear interpolation was used to interpolate the optimization parameters. Here, unlike Model 2, the error of approximation was above the set threshold of $\epsilon_{POD} = 0.01$ when the optimization results were obtained for the first time. This entails the use of iterative process discussed in Section 4. After applying the aforementioned algorithm, the error of approximation was achieved in 3 iterations and the summary of results for each iteration is presented in table 8 and figure 6. The optimization results here match the results reported in the original paper, i.e. in the presence of state constrains, the declining trend of number of good items represented by state variable $y_2$ was halted by imposing a minimum proportion.

**Table 8** Optimization results for Model 3

| Parameter | Iteration 1 | Iteration 2 | Iteration 3 |
|---|---|---|---|
| $b^{(0)}$ | [2.7,2.7,2.7,2.7, 0.0,0.0,0.0,0.0] | [0.7022,0.0000, 1.8041,1.4614, 1.8999,1.3773, 1.9082,4.0000] | [0.8376,0.7384, 2.1002,0.5137, 2.2801,1.3892, 1.6146,4.0000] |
| $Bounds$ | [0,3] [0,4] | [0,3] [1,4] | [0.0000,2.0137] [2.0000,4.0000] |
| $b^*$ | [0,0,0,0, 4,4,4,4] | - | - |
| $\hat{b}^*$ | [0.7022,0.0000, 1.8041,1.4614, 1.8999,1.3773, 1.9082,4.0000] | [0.8376,0.7384, 2.1002,0.5137, 2.2801,1.3892, 1.6146,4.0000] | [0.3097,0.3571, 0.1290,0.3428, 3.1945, 3.0046, 2.706, 4.0000] |
| $\psi_0(b^{(0)})$ | 16.0926 | - | - |
| $\psi_0(b^*)$ | 21.4360 | - | - |
| $\widehat{\psi_0(\hat{b}^*)}$ | 22.2183 | 26.1910 | 25.1481 |
| $\psi_0(\hat{b}^*)$ | 20.9054 | 25.0440 | 25.0711 |
| $Time_{orig}$ | 20.4522 sec | - | - |
| $Time_{cnstr}$ | 15.1739 sec | 16.3270 sec | 20.1867 sec |
| $Time_{surr}$ | 2.2351 sec | 2.8383 sec | 7.9384 sec |
| $\psi_1(b^{(0)})$ | 0.1568 | - | - |
| $\psi_1(\hat{b}^*)$ | 0 | 0 | 0 |
| $\widehat{\psi_1(\hat{b}^*)}$ | 0 | 0 | 0 |
| $\epsilon$ | 0.0628 | 0.0458 | 0.0031 |

However, the main goal of this analysis is not to evaluate the accuracy of actual responses, but rather analyze how accurately and efficiently surrogate model could

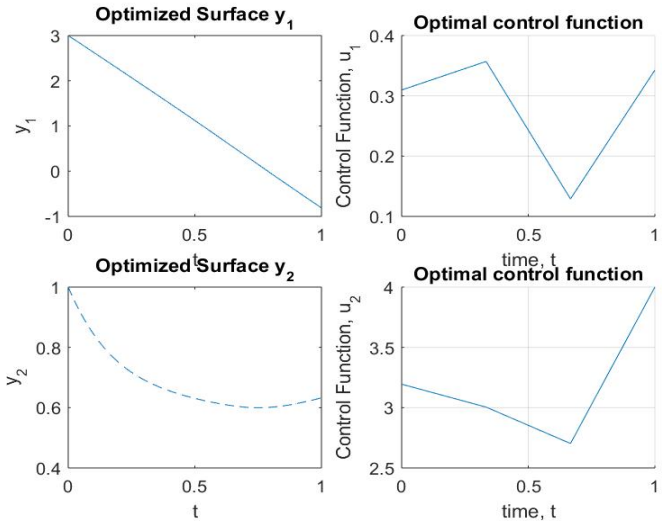


**Fig. 9** Optimal surface and optimal control plots for Model 3. SurrogateModel_Production

predict the system responses. The results from this model have given an ideal example of the iterative procedure described before. As illustrated in figure 10 for first optimization parameter of each control function, the domain of the control function decreases with each iteration and a new surrogate model is constructed to determine $\hat{b}^*$, which moves closer to $b^*$ as the algorithm moves forward. This domain is centered at the interpolated value of optimization parameters for each control function (note: this is not evident in figure under consideration as only one optimization parameter is plotted for simplicity whereas 4 optimization parameters are used). The distance between $\hat{b}^*$ of final iteration and $b^*$ is visibly higher in this case. This can be interpreted using the results reported in table 8.

For this model, the surrogate model has better results of optimization than the original model: the maximum value of the optimization criteria for the original model $\psi_0(b^*)$ was 21.4360, whereas, at the end, the optimal value obtained by surrogate model is 25.0711. This implies that surrogate model performed better than the original model in this case. Surrogate model also outperformed the original model in terms of time for solution of optimization problem: the problems that took the over 20 seconds to be solved by original model could be solved in less than 3 seconds with the surrogate model. On the other hand, The construction time for the surrogate model exceeds the time taken by the optimization through the original model. This could be a concern if the optimization problem is being solved only once.

As previously discussed, for many applications, specially those involving inverse problems, the optimization problems are required to be solved repetitively. Since the

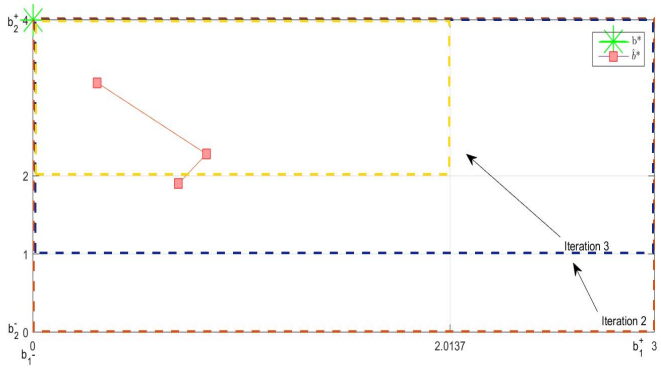


**Fig. 10** Illustration of iterative algorithm for Model 3

surrogate model is constructed once for all, the computational time gained by using surrogate models for the optimization can easily overtake the time of construction. Furthermore, with more complicated dynamical systems, the computational cost demanded by original models increase rapidly, while the computational time needed to solve the same using surrogate models remain low. Hence, these reduced order models dynamically adapt to the system behavior and provide highly accurate solutions to the optimal control problems, while decreasing the computational complexity and are therefore recommended to be adapted in many other applications.

## 6 Conclusion

This research employs Proper Orthogonal Decomposition (POD), a surrogate modeling technique integrated in optimization framework for dimension reduction of associated dynamical systems. POD extracts hidden structures from high dimensional data and projects them on lower dimensional space using the statistical method of singular value decomposition. In the first instance, POD is coupled with various Radial Basis Functions (RBF)— a smoothing technique— and the computational procedure is hypothesized to provide compact, accurate and computationally efficient solution of optimal control problems. The main contribution of this research is to enhance these surrogate models by introducing various sampling methods to the algorithm, and using an iterative algorithm, to achieve more accurate results.

The algorithm and computational procedure is implemented on three real-life optimal control problems that are taken directly from literature sources. It is demonstrated that the dimensionality of high order models in the form of ODEs of dynamical systems could be reduced substantially to as low as 3 with relative maximum absolute error less than 0.01 between original and approximated system responses.

Hence approximated surrogate model gave a good alternative method of solution of ODEs with low CPU intensity. The simulation part of PDF-RBF procedure is carried out by varying the number of sample points, sampling strategy, and RBF interpolation types in the training phase. The results showed that the approximation is more precise if the model is trained on higher number of sample points. Also, the interpolated surrogate model constructed using cubic-spline RBF led to better results in the complex model than its liner counterpart. Furthermore, LHS and SLHS both led to better approximations than RS, which highlights the significance of our proposal to use memory-based sampling techniques.

In solution of optimization problems, the system responses obtained by surrogate model invariably gave accurate results with improved computational time. As a whole, all three models agreed with the hypothesis of this work that surrogate models can increase the computational efficiency in solution of dynamical systems while maintaining the accuracy of system responses. However, the construction time of the surrogate models is subject to the available computational resources and the numerical simulation might be much faster in a high-performance computer, compensating for the time used in iterative process of POD-RBF algorithm.

The presented surrogate model algorithm significantly enhance the existing surrogate modelling technique. The algorithm also establishes a new paradigm for coupling optimization and modern statistics using data analytics methods. The proposed framework opens up a whole new avenue of research for utilizing surrogate models, specially in the machine learning research that estimate hyper-parameter through optimization problems, and require methods to ease the computational burden.

## 6.1 Limitations and Future Work

ROMs are usually thought of as computationally inexpensive mathematical representations that offer the potential for near real-time analysis. The hypothesis of this research is based on the same notion. However, while analyzing the performance POD-RBF procedure on non-linear dynamical systems in the last section of this work, it is brought into consideration that the even though the optimization process itself is faster with surrogate responses, their construction is sometimes computationally expensive as it involved accumulating a large number of system responses to input parameters. It is also noteworthy that sometimes ROMs lack robustness with respect to parameter changes. These limitations are considered, however the detailed discussion of these issues and their solutions are left for the future work.

In future, the performance of surrogate models can be evaluated on more complicated models consisting of highly non-linear ordinary and partial differential equations. Also, other sampling techniques which allow inclusion of corner and optimization points in the training set, methods of obtaining POMs, and interpolation methods can be explored as an extension of this work. Furthermore, the computational time of each of the model can be calculated with more efficient machines in

homogeneous computer environment to get near-exact insight into the performance of surrogate models.

## References


[1] A. A. Adongo, A. Lewis, and J. C. Chikelu. "Principal Component and Factor Analysis of Macroeconomic Indicators". In: *IOSR Journal Of Humanities And Social Science* 23.7 (2015), pp. 01–07. ISSN: 2279-0837, 2279-0845.

[2] A. Bai, S. Heera, and P.S. Deshpande. "An Application of Factor Analysis in the Evaluation of Country Economic Rank". In: *Procedia Computer Science* 54.3 (2015), pp. 311–317. DOI: 10.1016/j.procs.2015.06.036.

[3] J. Bai and P. Wang. "Econometric Analysis of Large Factor Models". In: *Annual Review of Economics* 8.1 (2016), pp. 53–80. DOI: 10.1146/annurev-economics-080315-015356.

[4] A. Benaarbia and A. Chrysochoos. "Proper orthogonal decomposition preprocessing of infrared images to rapidly assess stress-induced heat source fields". In: *Quantitative InfraRed Thermography Journal* 14.1 (2017), pp. 132–152. DOI: 10.1080/17686733.2017.1281553.

[5] V. Berardi et al. "Proper Orthogonal Decomposition Methods for the Analysis of Real-Time Data: Exploring Peak Clustering in a Secondhand Smoke Exposure Intervention". In: *Journal of Compuatational Science* 11 (2015), pp. 102–111.

[6] M. H. A. Biswas and A. Ali. "Production and process management: An optimal control approach". In: *Yugoslav Journal of Operations Research* 26 (Jan. 2016), pp. 8–8. DOI: 10.2298/YJOR141015008K.

[7] S. N. Boyaval et al. "Reduced Basis Techniques for Stochastic Problems". In: *Archives of Computational Methods in Engineering* 17 (Dec. 2010), pp. 435–454. DOI: 10.1007/s11831-010-9056-z.

[8] V. Bujlak. *Inverse Analysis with Model Reduction (Proper Orthogonal Decomposition in Structural Mechanics)*. Milan: Springer, 2012.

[9] Y. Chang et al. "Reduced-Order Modeling for Complex Flow Emulation by Common Kernel-Smoothed Proper Orthogonal Decomposition". In: *AIAA Journal* 59.9 (2021), pp. 3291–3303. DOI: 10.2514/1.J060574. eprint: https://doi.org/10.2514/1.J060574.

[10] A. Chatterjee. "An Introduction to the Proper Orthogonal Decomposition". In: *Current Science* 78.7 (Apr. 2000).

[11] Y. Chen et al. "Risk related brain regions detection and individual risk classification with 3D image FPCA". In: *Statistics & Risk Modeling* 35.3-4 (2018), pp. 89–110. DOI: 10.1515/strm-2017-0011.

[12] L. Chrisman. *Latin Hypercube vs. Monte Carlo Sampling*. July 2014.

[13] Sushant S. Garud, Iftekhar A. Karimi, and Markus Kraft. "Smart Adaptive Sampling for Surrogate Modelling". In: *26th European Symposium on Computer Aided Process Engineering*. Ed. by Z. Kravanja and M. Bogataj. Vol. 38.

Computer Aided Chemical Engineering. Elsevier, 2016, pp. 631–636. DOI: https://doi.org/10.1016/B978-0-444-63428-3.50110-7.

[14] B. Gooijer et al. "Evaluation of POD based surrogate models of fields resulting from nonlinear FEM simulations". In: *Advanced Modeling and Simulation in Engineering Sciences* 8 (Dec. 2021). DOI: 10.1186/s40323-021-00210-8.

[15] M. Hinze and S. Volkwein. "Proper Orthogonal Decomposition Surrogate Models for Nonlinear Dynamical Systems: Error Estimates and Suboptimal Control". In: vol. 45. Jan. 2005, pp. 261–306. DOI: 10.1007/3-540-27909-1_10.

[16] S. Im, J. Lee, and M. Cho. "Surrogate modeling of elasto-plastic problems via long short-term memory neural networks and proper orthogonal decomposition". In: *Computer Methods in Applied Mechanics and Engineering* 385 (2021), p. 114030. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2021.114030.

[17] M.D. Intriligator. "Applications of Optimal Control Theory in Economics". In: *Synthese* 31.2 (1975), pp. 271–288.

[18] M.D. Intriligator and B. Smith. "Some Aspects of the Allocation of Scientific Effort between Teaching and Research". In: *The American Economic Review* 56.1/2 (1966), pp. 494–507. ISSN: 00028282.

[19] E. Iulianoa and D. Quagliarellaa. "Proper Orthogonal Decomposition, surrogate modelling and evolutionary optimization in aerodynamic design". In: *Computers and Fluids* 84 (2013), pp. 327–350.

[20] G. Kerschen et al. "The Method of Proper Orthogonal Decomposition for Dynamical Characterization and Order Reduction of Mechanical Systems: An Overview". In: *Nonlinear Dynamics* 41 (2005), pp. 147–169.

[21] F. Lanata and A. D. Grosso. "Damage detection and localization for continuous static monitoring of structures using a proper orthogonal decomposition of signals". In: *Smart Materials and Structures* 15.6 (Nov. 2006), pp. 1811–1829. DOI: 10.1088/0964-1726/15/6/036.

[22] K. Li, J. D. Cursio, and Y. Sun. "Principal Component Analysis of Price Fluctuation in the Smart Grid Electricity Market". In: *Sustainability* 10.11 (2018). ISSN: 2071-1050. DOI: 10.3390/su10114019.

[23] A. Maravalle and L. Rawdanowicz. "Changes in Economic and Financial Synchronisation". In: 1517 (2018).

[24] K. Pearson. "On Lines and Planes of Closest Fit to Points in Space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720.

[25] W. Schilders, H. V. Vorst, and J. Rommes. *Model Order Reduction: Theory, Research Aspects and Applications*. Vol. 13. Jan. 2008. DOI: 10.1007/978-3-540-78841-6.

[26] M. Shcherbatyy and I. Shcherbata. "Proper Orthogonal Decomposition for Ordinary Differential Equations and Partial Differential Equations". In: *Proceedings XXXII International Conference PDMU, Czech Republic, Prague* (Aug. 2018), pp. 162–170.

[27] L. Sirovich. “Turbulence and the dynamics of coherent structures. I. Coherent structures”. In: *Quarterly of Applied Mathematics* 45.3 (1987), pp. 561–571. ISSN: 0033569X, 15524485.

[28] L. Sirovich and M. Kirby. “Low-dimensional procedure for the characterization of human faces”. In: *J. Opt. Soc. Am. A* 4.3 (Mar. 1987), pp. 519–524. DOI: 10.1364/JOSAA.4.000519.

[29] N. M. Tran et al. “Principal component analysis in an asymmetric norm”. In: *Journal of Multivariate Analysis* 171 (2019), pp. 1–21. ISSN: 0047-259X. DOI: 10.1016/j.jmva.2018.10.004.

[30] M. Wax and T. Kailath. “Detection of signals by information theoretic criteria”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33.2 (Apr. 1985), pp. 387–392. ISSN: 0096-3518. DOI: 10.1109/TASSP.1985.1164557.

[31] R. Zimmermann. “Gradient-enhanced surrogate modeling based on proper orthogonal decomposition”. In: *Journal of Computational and Applied Mathematics* 237.1 (2013), pp. 403–418. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2012.06.010.

# IRTG 1792 Discussion Paper Series 2021

For a complete list of Discussion Papers published, please visit http://irtg1792.hu-berlin.de.

001 "Surrogate Models for Optimization of Dynamical Systems" by Kainat Khowaja, Mykhaylo Shcherbatyy, Wolfgang Karl Härdle, January 2021.

**IRTG 1792, Spandauer Strasse 1, D-10178 Berlin**
**http://irtg1792.hu-berlin.de**

This research was supported by the Deutsche Forschungsgemeinschaft through the IRTG 1792.