

Khowaja, Kainat; Shcherbatyy, Mykhaylo; Härdle, Wolfgang Karl

Working Paper

Surrogate Models for Optimization of Dynamical Systems

IRTG 1792 Discussion Paper, No. 2021-001

Provided in Cooperation with:

Humboldt University Berlin, International Research Training Group 1792 "High Dimensional Nonstationary Time Series"

Suggested Citation: Khowaja, Kainat; Shcherbatyy, Mykhaylo; Härdle, Wolfgang Karl (2021) : Surrogate Models for Optimization of Dynamical Systems, IRTG 1792 Discussion Paper, No. 2021-001, Humboldt-Universität zu Berlin, International Research Training Group 1792 "High Dimensional Nonstationary Time Series", Berlin

This Version is available at:

<https://hdl.handle.net/10419/230835>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

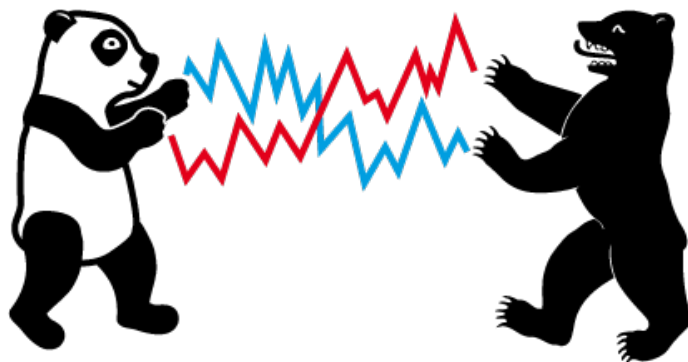


Surrogate Models for Optimization of Dynamical Systems

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".

Surrogate Models for Optimization of Dynamical Systems *

Kainat Khowaja, Mykhaylo Shcherbatyy, and Wolfgang Karl Härdle

Abstract Driven by increased complexity of dynamical systems, the solution of system of differential equations through numerical simulation in optimization problems has become computationally expensive. This paper provides a smart data driven mechanism to construct low dimensional surrogate models. These surrogate models reduce the computational time for solution of the complex optimization problems by using training instances derived from the evaluations of the true objective functions. The surrogate models are constructed using combination of proper orthogonal decomposition and radial basis functions and provides system responses by simple matrix multiplication. Using relative maximum absolute error as the measure of accuracy of approximation, it is shown surrogate models with latin hypercube sampling and spline radial basis functions dominate variable order methods in computational time of optimization, while preserving the accuracy. These surrogate models also show robustness in presence of model non-linearities. Therefore, these computational efficient predictive surrogate models are applicable in various fields, specifically to solve inverse problems and optimal control problems, some examples of which are demonstrated 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 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).

Chapter 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. 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 (ROM) 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 [21], 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. 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 [14].

This paper employs Proper Orthogonal Decomposition (POD), a model reduction technique originating in statistical analysis and known for its optimality as it captures the most dominant components of data in the most efficient way [11]. POD serves the purpose of dimension reduction by extracting hidden structures from high dimensional data and projecting it on lower dimensional space [15]. In this work, POD will be used to derive low order models of dynamical system by reducing a high number of interdependent variables to a much smaller number of uncorrelated variables while retaining as much as possible of the variation in the original variables.

Over a century ago, Pearson proposed the idea of representing the statistical data in high dimensional space using a straight line or plane, hence discovering a finite

dimensional equivalence of POD as a tool for graphical analysis [19]. 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 branch in which it is being tackled. Despite its early discovery, the availability of computational resources required to compute POD modes were limited in earlier years and the technique remained virtually unused until 1950s. The technological advancements took an upturn after that with the invention of powerful computers and led to the popularity of POD [15]. Since then, the development and applications of POD have been widely investigated in diverse disciplines such as structural mechanics [15], aerodynamics [14], 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 [16], human face recognition [23], detection of signals in multi-channel time-series [25], 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, 18], 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 [24]. These POD equivalent tools have also been adopted in analysis on several instances such as [1, 9, 17, 24]. Yet, most of the literature exploits only the real life data for dimension reduction. Even though some analysis highly relies on real life data, there is an urgent need of introduction of tools that utilize simulated data generated from the non-standard models with nonlinear differential equations that are on constant rise and hold potential for enrichment of analysis.

Moreover, optimal control problems and mathematical optimization models are widely seen in various applications. These models are often used for normative purposes to solve the minimization/maximization problems and require repetitive evaluation in various context with different parameter values to find the optimum set of parameters. This parameter exploration process can be computationally intense, specially in complex non-linear system which emphasize the need of dimension reduction for these models.

Through this research, the application of POD to reduce the dimensionality of dynamical systems is proposed. The present work resorts to explore the efficacy of POD on few common applications, the models which have been previously defined and commonly used. We hypothesize that the system responses of dynamical models can be obtained with a very high accuracy, but lower computational cost model reduction technique. The novelty of this hypothesis lies in the fact that dimensional reduction techniques have rarely been explored for optimal control problems, spe-

cially the combination of POD and Radial Basis Functions (RBF) to make surrogate models is quite under utilized, specially for the the models discussed in this paper.

The computational procedure of the 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 of ODE (model of dynamical system), 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). 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 [21]. Next, the quality of the model is validated and evaluated by carrying out error analysis and various experimental designs are employed by varying sampling and interpolation techniques and changing the size of training set to determine the combination that generates that results in the least maximum absolute error. Finally, using that experimental design, optimization criterion are calculated using both models to evaluate accuracy of the model. For the computations, a MATLAB software is developed by the author which utilizes a combination of inbuilt and user-defined functions. The illustrations used in this work are also generated using MATLAB.

The next chapter will lay down theoretical concepts related to POD, SVD and RBF, and how surrogate models are constructed to project the dynamical system onto subspaces consisting of basis elements that contain characteristics of the expected solution. Chapter 2 will explain how the computational procedure (algorithm) and Chapter 3 will implement the concepts developed in Chapter 1 and methodology presented in Chapter 2 on a set of dynamical systems. Finally, last chapter will conclude the main results and provide a summary of current research, its limitations, as well as future prospects.

Chapter 2

Mathematical Formulation

Model reduction techniques have been known for their ability to reduce the computational complexity of mathematical models in numerical simulations. The main reason ROM has found applications in various disciplines is due to its strong theoretical foundations and the demand of model reduction techniques in ever-so-rising computational complexities and intrinsic property of high dimensionality of physical system. ROM addresses these issues effectively by providing low dimensional approximations.

Although a variety of dimensionality-reduction techniques exist such as operational based reduction methods [20], reduced basis methods [6], the ROM methodology is often based upon POD. Analogous to PCA, the POD theory requires to 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 formulated 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.

This chapter discusses in depth the mathematical concepts associated with POD and its correspondence with SVD and RBF for construction of surrogate models. The computational procedure adapted in Chapter 3 and Chapter 4 is strictly based on the theory formulated in this chapter.

1 Formulation of Optimization Problem

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 func-

tion. 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 Chapter 4 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 $y^* \in Y_S$, that minimizes the optimization criterion (objective function)

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

subject to ODEs (state equation)

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

box constrains on the control variable

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

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

$$\begin{aligned} \tilde{\psi}_j(u, y) &= 0, j = 1, \dots, m_1, \\ \tilde{\psi}_j(u, y) &\leq 0, j = m_1 + 1, \dots, m. \end{aligned} \quad (4)$$

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

We assume that for each $u \in U$, there exists a unique solution $y(u)$ of state equation $c(y, u) = 0$. With the aim of compactness, we will write optimization problem (1-4) in reduced form: find a function u^* such that

$$\begin{aligned} u^* \in U_{\partial u}, \psi_0(u_*) &= \min_{u \in U_{\partial u}} \psi_0(u) \\ U_{\partial u} &= \{u : u \in U; \psi_j(u) = 0, j = 1, \dots, m_1; \psi_j(u) \leq 0, j = m_1 + 1, \dots, m\} \\ c(y(u), u) &= 0 \\ \psi_j(u) &= \tilde{\psi}_j(u, y(u)), j = 0, 1, \dots, m \end{aligned} \quad (5)$$

The optimal control problems in this research are solved using 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 advantage of direct methods is that the optimality conditions of a nonlinear programming problem are generic, whereas optimality conditions of undiscritized optimal control problems need to be reestablished for each new problem and often require partial a-priori knowledge of the mathematical structure of the solution which in general is not available for many practical problems.

The first step in the direct method is to approximate each component of the control vector by a function of finite parameters $u_i(t) = u_i(t, b^{(i)}), b^{(i)} =$

$[b_1^{(i)}, \dots, b_{n_i}^{(i)}]^\top, i = 1, \dots, n_u$. 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 $u_i(t), i = 1, \dots, n_u$.

The optimization problem can be written as nonlinear programming problem in such a way that we have to find a vector b^* such that

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

2 Surrogate Model for Optimization Problem

Solution of optimization problem in equation (6) requires multiply solutions of state equation $c(y(b), b) = 0$ and calculation of optimization criteria $\tilde{\psi}_0$ and constraints $\tilde{\psi}_j, j = 1, \dots, m$ of the system for different values of optimization parameters b . Complexity of mathematical models (state equation), which describe state and behavior of considered dynamical system requires significant computing resources (CPU time, memory,...) and occasionally puts in question the solving of the optimization problem itself. In order to solve multi-query problems within limited computational cost, there is a need to construct approximation models (also known as surrogates models, meta-models or ROMs). Surrogate model replaces the high-fidelity problem and tends to much lower numerical complexity.

In this paper surrogate models are constructed by first selecting a sampling strategy. Then, n_s sampling points are generated and for each sample point $b^{(i)}$, we solve state equation in equation (6) (ODEs) and obtain n_s vectors of solutions (snapshots) $Y_i = [y(t_1, b^{(i)})^\top, \dots, y(t_{n_t}, b^{(i)})^\top]^\top \in \mathbb{R}^m, m = n_y \times n_t$ at different time instances, $t_0 < t_1 < t_2 < \dots < t_{n_t} = T$. Snapshots vectors Y_i create snapshot matrix $Y = [Y_1, Y_2, \dots, Y_n] \in \mathbb{R}^{m \times n_s}$.

Next, we construct surrogate model using POD and RBF and calculate the value of functionals $\hat{\psi}_j(b) = \tilde{\psi}_j(b, \hat{y}), j = 0, 1, \dots, m$. Detailed description of POD-RBF procedure is presented in the following paragraphs of this chapter. The formulation of optimal control problem for surrogate model is to find a vector \hat{b}^* such that:

$$\begin{aligned}
\hat{b}^* &\in U_{\partial}, \hat{\psi}_0(\hat{b}^*) = \min_{b \in U_{\partial}} \hat{\psi}_0(b) \\
U_{\partial} &= \{b : b \in U_b, \hat{\psi}_j(b) = 0, j = 1, \dots, m_1; \hat{\psi}_j(b) \leq 0, j = m_1 + 1, \dots, m\} \\
U_b &= \{b : b \in \mathbb{R}^n, b^- \leq b \leq b^+, b^- \in \mathbb{R}^n, b^+ \in \mathbb{R}^n\} \\
\hat{y} &= S(b) \\
\hat{\psi}_j(b) &= \tilde{\psi}_j(u(b), \hat{y}), j = 0, 1, \dots, m
\end{aligned} \tag{7}$$

Replacing the state equation in (6) with surrogate model given in equation (7) is hypothesized to decrease the computational time by a significant amount, because 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. The hypothesis is tested by comparing the accuracy of system responses and time of calculation for both equation (6) and equation (7). The detailed procedure for testing of surrogate model is discussed in the next chapter.

3 Initial Sampling and Method of Snapshots

The method of snapshots for POD was developed by Sirovich [22] 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 $t_i, i = 1, \dots, n_t$. 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. 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 [14].

The main sampling methodology used in the computational procedure is Latin Hypercube Sampling (LHS) and its variant Symmetric Latin Hypercube Sampling (SLHS). LHS is a 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 proved to have faster speed and lower sampling error than RS [10].

SLHS was introduced as an extension of LHS that achieves the purpose of optimal design in a relatively more efficient way. It was also established that sometimes,

SLHS had 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.

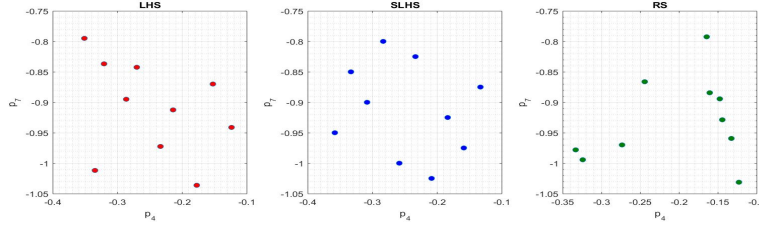


Fig. 1 Comparison of various sampling techniques.  SurrogateModel

4 Approximation

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 ROMs by projecting the original model onto the reduced space spanned by the POD modes [14]. A simple mathematical formulation of POD technique is laid out in this section which closely follow the references [7, 8, 21].

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 $b \in \mathbb{R}^{n_u}$ over a certain domain Ω . The ROMs approximate the state function $y(t)$ in domain Ω using linear combination of some basis function $\phi^i(x)$ such that

$$y(t) \approx \sum_{i=1}^M a_i \cdot \phi^i(t) \quad (8)$$

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 and choice is clearly not unique. Once the basis function is chosen, the amplitudes are determined by a minimization process and the least square error of approximation is calculated. It is ideal to take orthonormal set as the basis with the property

$$\int_{\Omega} \phi_{k_1}(t) \cdot \phi_{k_2}(t) dx = \begin{cases} 1 & k_1 = k_2 \\ 0 & k_1 \neq k_2 \end{cases} \quad (9)$$

This way, the determination of the amplitudes a_k only depends on function $\phi_k^i(t)$ and not on any other ϕ . Along with being orthonormal, the basis should approximate

the function in best possible way in terms of the least square error. Once found, these special ordered orthogonal functions are called the POMS for the function $y(t)$ and the equation (8) 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 is further clarified in the context of SVD in the next section.

5 Singular Value Decomposition

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. Since the theory of SVD is so widespread, this section will only highlight the most general and relevant details of SVD that are helpful in derivation of POMs and POD basis.

In general, SVD is a technique that is used to decompose any real rectangular matrix Y into three matrices, U , Σ and V , where U and V are orthogonal matrices, where Σ is a diagonal matrix that contains the singular values σ_i of Y , sorted in a decreasing order such that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_d \geq 0$, where 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 $k < d$, the first k columns of the matrix U serve as the basis $\phi^i, i = 1, \dots, k$. These set of columns, gathered in matrix Φ , form an orthonormal set of basis for our new low-dimensional surface and k is referred as the rank.

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

Literature on SVD has established that the relative magnitude of each singular value with respect to all the others give a measure of importance of the corresponding eigen-function 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 ϵ_{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.

$$E(k) = \frac{\sum_{i=1}^k \sigma_i^2}{\sum_{i=1}^d \sigma_i^2} \leq 1 - \epsilon_{\text{POD}}^2 \quad (10)$$

6 Radial Basis Functions

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, they can be any arbitrary combination of numbers between those ranges. That is why, the newly constructed model needs to be approximated in a better way. In this research, POD is coupled with RBF to create low-order parameterization of high-order systems for accurate prediction of system responses.

RBF is a unique interpolating technique that determines one continuous function that is defined over the whole domain. It is a widely used smoothing and multi-dimensional approximation technique. For construction of surrogate model using our current basis, a function $f(b) = y$, where b is the vectors of some parameters and y is the output of the system that has to be estimated. Let Y_k be the reduced dimensional matrix calculated by multiplication of basis and amplitudes matrices. It is now easy to apply RBF to reduced dimensional space where system responses are expressed as amplitudes in the matrix A_k . Therefore, the surrogate model takes the form $f_a(b) = a$, where a is the vector of amplitudes. Hence,

$$f(b) = y = \Sigma_k A_k = \Sigma_k f_a(b) = \phi f_a(b) \quad (11)$$

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

$$f_a(b) = \begin{bmatrix} a_1^i \\ a_2^i \\ \vdots \\ a_K^i \end{bmatrix} = \begin{bmatrix} d_{11} \\ d_{21} \\ \vdots \\ d_{K1} \end{bmatrix} \cdot g_1(b) + \begin{bmatrix} d_{12} \\ d_{22} \\ \vdots \\ d_{K2} \end{bmatrix} \cdot g_2(b) + \dots + \begin{bmatrix} d_{1N} \\ d_{2N} \\ \vdots \\ d_{KN} \end{bmatrix} \cdot g_N(b) = D \cdot g(b) \quad (12)$$

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 (11), our initial space y can be approximated by:

$$y \approx \Phi \cdot D \cdot g(b) = \hat{y} \quad (13)$$

In this work, linear and cubic spline RBF are used for analysis, given by:

$$\text{linear spline : } g_j(b) = ||b - b_j||; \quad \text{cubic spline : } g_j(b) = ||b - b_j||^3; \quad (14)$$

Since matrix Φ 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 response.

Replacing the state equation (2) with surrogate model given in equation (13) is hypothesized to decrease the computational time by a significant amount, because 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. The hypothesis is tested by comparing the accuracy of system responses and time of calculation for both equation (2) and equation (13). The detailed procedure for testing of surrogate model is discussed in the next section.

7 Error Analysis

The final step in the analysis of surrogate models is to determine how accurate the low-dimensional surrogate model are in determination of the system responses.

This is done by generating n_g sample points of set of parameters P , using the same sampling technique that had been adapted for generation of training test. It must be noted that the newly generated test points are not same as the one used to train the model and hence, the system responses of these points occur in between nodes and are ideal for checking the accuracy of the models. Moving on, the system responses $Y_g = [y_1, y_2, \dots, y_{n_g}] \in \mathbb{R}^{m \times n_g}$ are obtained using initial numerical method (that solves entire system), and also $\hat{Y}_g = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{n_g}] \in \mathbb{R}^{m \times n_g}$ are recorded using newly constructed surrogate model. Then, the accuracy and error measures are generally calculated using the following formulas:

$$R^2 = 1 - \frac{\sum_{j=1}^{n_g} |y_j - \hat{y}_j|}{\sum_{j=1}^{n_g} |y_j - \bar{y}_j|} \quad (15)$$

$$MAE = \frac{1}{n_g} \sum_{j=1}^{n_g} |y_j - \hat{y}_j| \quad (16)$$

$$MXAE = \max_{1 \leq j \leq n_g} |y_j - \hat{y}_j| \quad (17)$$

$$RMAE = \max_{1 \leq i \leq m} \max_{1 \leq j \leq n_g} \frac{|y_{ji} - \hat{y}_{ji}|}{y_{ji}} \quad (18)$$

All four measures are put to use at various instances in the thesis, for example, coefficient of determination (R^2) in equation (15), Mean Absolute Error (MAE) in equation (16), Maximum Absolute Error (MXAE) in equation (17) are evaluated for various rank approximations of SVD, whereas a tolerance threshold for relative Maximum Absolute Error (RMAE) in equation (18) is defined for testing the accuracy of optimization results obtained through original and surrogate models.

Chapter 3

Algorithm

While understanding of mathematical formulation of POD-RBF procedure presented in Chapter 2 is essential, its implementation can be quite technical as it involves high-dimensional matrices, a series of functions, complex loops and iterative processes. The idea of this chapter is to give detailed description of the algorithm that was implemented in MATLAB for this research. The whole computational procedure is divided into three parts for simplicity: experimental design, training phase and testing phase. For each part, a section of the chapter is devoted in which importance of the steps of algorithm are discussed and the intricacies of computational procedure are highlighted. Finally, the iterative nature of algorithm is elaborated in Section 4.

1 Experimental Design

For the construction of surrogate model for a dynamical system, the proper definition of the optimal control problem and planning an appropriate experimental design holds high importance since these conditions are hypothesized to reflect on the accuracy of the model. This pivotal decision relies on choice of fixed and variable parameters, values of constants for fixed parameters, the domain of variable/control parameters, number of initial sampling points, number of time-instances for computation of snapshots, the sampling strategy, the interpolation technique, and minimum error of approximation/ stopping criteria.

Because of the inherent dependence of model on the factors enlisted above, the decision about experimental design has to be made before the construction of surrogate model. In this research, various combination of these factors are accounted for to determine which experimental design results in the highest accuracy while satisfying the time constraints for generation of the snapshots. The error of approximation can be defined for accuracy of system responses or computational time or both.

2 Offline/Training Phase

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 of ODE (model of dynamical system), obtainment of proper orthogonal modes via singular value decomposition and estimation of POD expansion coefficients that approximate the POD basis via RBFs.

The next step in the analysis is to determine the appropriate number of POD modes to be used in the surrogate model. For that, the orthogonal basis are found using SVD and the error measures (15), (16), and (17) are used to determine the singular values (rank) whose corresponding eigenvectors will be used as POD basis. Next, the amplitudes of approximation a_i are computed using the basis vectors $\phi_i, i = 1, \dots, k$ and stored in amplitude matrix A_k . With this, the dimensionality of this problem is cut from n_s to just k (rank). Now, to obtain the system response for any arbitrary data point, it is sufficient to simply multiply the reduced basis with corresponding amplitude.

In the final part of offline phase, POD is combined with RBF. The coefficients of RBF interpolation collected in matrix D are calculated using our initial data points in u and our final matrix of amplitudes A_k as inputs. With this, the training phase comes to an end. Now, for the computation of system responses, $y \approx \phi \cdot D \cdot g(b)$, surrogate model can be used with only $g(b)$ remaining to be calculated, which depends upon the test points.

3 Online/Testing Phase

The last step in construction of low dimensional model is to check the overall error of approximation. It is done by taking the sample points and for each of the data point, first the original response of the function is recorded by solving the ODEs using MATLAB solver *ode15s*, and then the newly developed surrogate model is used to calculate the system response for the same data point. Finally the error measure RMAE (18) is calculated for each experimental design and compared to determine which combination meets the required tolerance level.

After deciding the final sampling strategy, number of sampling points, and interpolation technique, the optimization problem is solved using system responses for both original and surrogate models. Then, RMAE is calculated to evaluate the accuracy of surrogate model. If the accuracy level is above the decided threshold, the algorithm enters an iterative process that allows decreasing the width of domain of control parameters. A detailed discussion of iterative process is demonstrated in next section.

4 Iterative Process

As stated in the previous section, when the optimization results are obtained using both original and surrogate model, sometimes the desired accuracy of the model is not obtained in the first iteration, despite selecting the best experimental design. This is because the optimal values are usually the corner points and the predictive models in general tend to perform poorly on extreme ends. One of the most effective method to overcome this issue is by the use of adaptive sampling, a method that has been used by many researchers such as [14] with the aim of finding optimal design space points. Despite the effectiveness of the approach, it was not adopted in this work due to limited scope of the research, as previously explained in Section 3.

The algorithm used for this research, on the other hand, caters to the aforementioned 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 design space, the sampling points move closer and 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. If the accuracy is not achieved, the iterative algorithm becomes active: every time the error of approximation is higher than the tolerance level, it shortens the domain, and reconstructs the surrogate model for analysis. The iterative process can be summarized in four steps:

1. Initialization: In this step parameters of algorithm are initialized that are required for the iterative process, such as 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 the data structure. 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 to exceed the initial bounds. The step 2 of iterative process is depicted for two optimization parameters in figure 2.
3. Optimization: This is the main step of algorithm which was discussed in detail in the second and third section of this chapter. In summary, we make sampling set and snapshots, create surrogate model, solve optimization problem and calculate error.

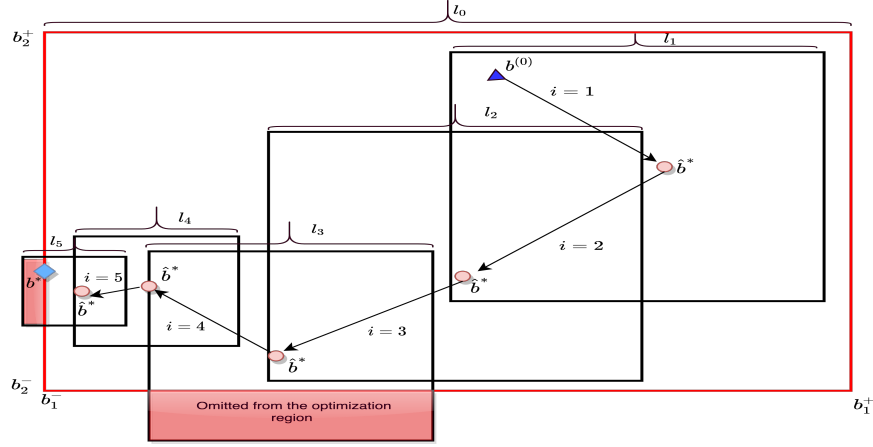


Fig. 2 Example of iterative algorithm of two optimization parameters b_1 and b_2 with iterations $i = 1, \dots, 5$ and recursively decreasing lengths $l_i, i = 1, \dots, 5$

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, it stops the iterative process. Else it goes back to step 2.

The computation procedure discussed throughout this chapter is summarized in flowchart presented in the figure 3.

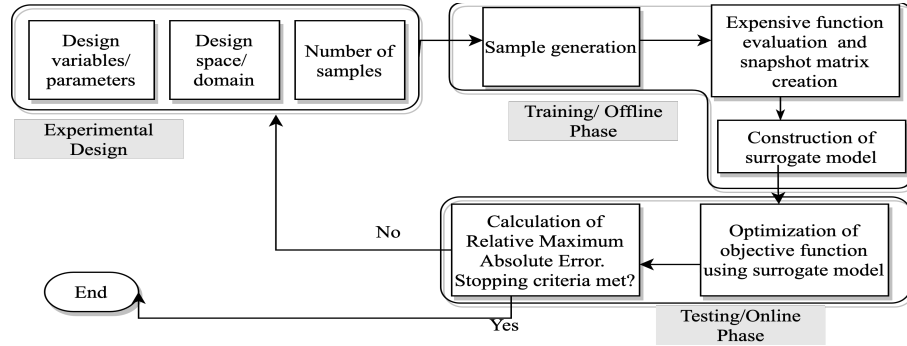


Fig. 3 POD-RBF algorithm flowchart

Chapter 4

Application of POD-RBF Procedure on Dynamical Systems

In this chapter, the POD-RBF procedure is trained and used to construct the surrogate models for real-life dynamical systems and solve associated optimization problems. Two dynamical systems with various complexity are presented, with model 1 being the simple non-linear ODE problem, and model 2 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 chapter, the variables with 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 |

1 Model 1: Science Policy

1.1 Description of the Model

This section 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" [13]. 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 [12]. In order to find the optimal allocation, the optimal control problem was formulated as following:

$$\begin{aligned} \max_{(u,y) \in U \times Y} \tilde{\psi}_0 &= \int_{t_0}^T [0.5y_1(t) + 0.5y_2(t)] dt, \quad (19) \\ \text{subject to} \quad c(y, u) = 0 &\sim \begin{cases} y_1'(t) - u(t)gy_1(t) + \delta y_1(t) = 0, t \in [t_0, T] \\ y_2'(t) - (1 - u(t))gy_1(t) + \delta y_2(t) = 0 \\ y_1(t_0) - y_{10} = 0, y_2(t_0) - y_{20} = 0 \end{cases} \\ \begin{bmatrix} \tilde{\psi}_1 \\ \tilde{\psi}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} &\sim \begin{cases} y_1(T) - y_{1T} = 0 \\ y_2(T) - y_{2T} = 0 \end{cases} \\ u^- &\leq u(t) \leq u^+ \end{aligned}$$

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. As the control variable u represents the number of new scientists becoming teachers, $(1 - u)$ represents the proportion of researchers. Hence, the differential equations determine the rate of change of number of teachers and researchers by subtracting the exiting 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.

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 (19). 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

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 |
| δ | 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 |

completely. The assumption, even though unrealistic, is granted for simplicity and can be complicated at the later stages.

1.2 Simulation

This system of equation 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 plots of singular values of depicted similar pattern for all the experimental designs. The singular value plot for one specific example, SLHS and $n_s = 40$ is presented figure 4 and shows that the first 4 singular values explain almost 100% variance. Given the criterion in equation (10), we choose the rank of $k = 4$.

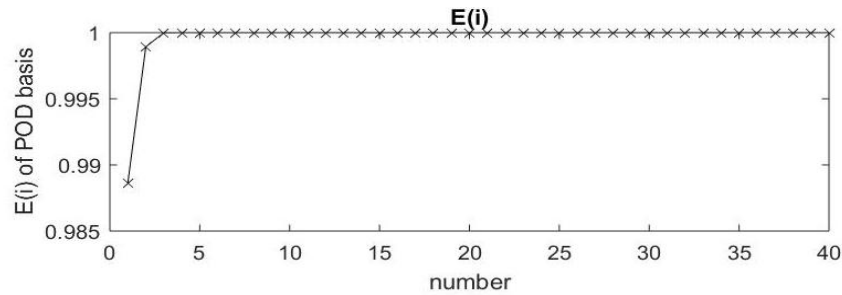



Fig. 4 Cumulative energy plot to determine singular values for Model 1.  Surrogate-Model_SciencePolicy

The surrogate model was 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 |

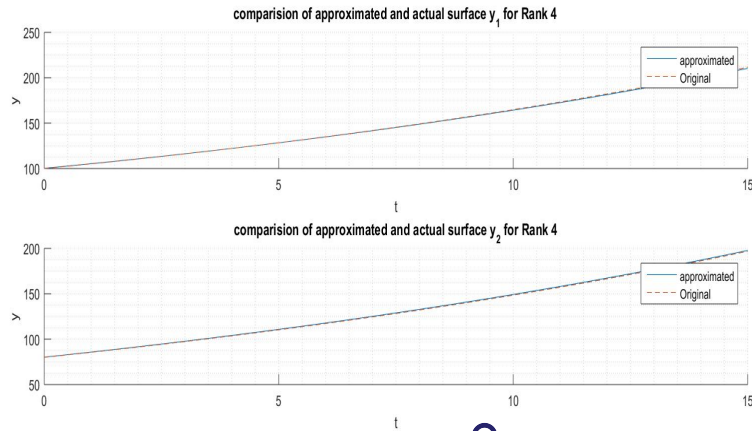


Fig. 5 Actual surface vs approximated surface for Model 1. [SurrogateModel_SciencePolicy](#)

1.3 Optimization

For the final step of analysis, the surrogate model was constructed with 40 training points, LHS, and linear spline RBF. Here $n_s = 40$ was used 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 was 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 $\widehat{\psi}_0(\hat{b}^*)$) as low as 0.005 in the first iteration.

Even though the optimization using surrogate model was slightly quicker than the original model, the time taken for construction of surrogate model was 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) was 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 |
| $\psi_1(b^{(0)})$ | $[11.8001, 43.0163]^\top$ | $\psi_1(\hat{b}^*)$ | $[0.0003, 0.0014]^\top$ |
| $\psi_1(b^*)$ | $[0.000, 0.000]^\top$ | $\widehat{\psi}_1(\hat{b}^*)$ | $[0.0000, 0.0023]^\top$ |
| Time _{orig} | 2.8109 sec | Time _{surr} | 2.3694 sec |
| Time _{cnsr} | 37.8406 sec | ϵ | 0.0058 |

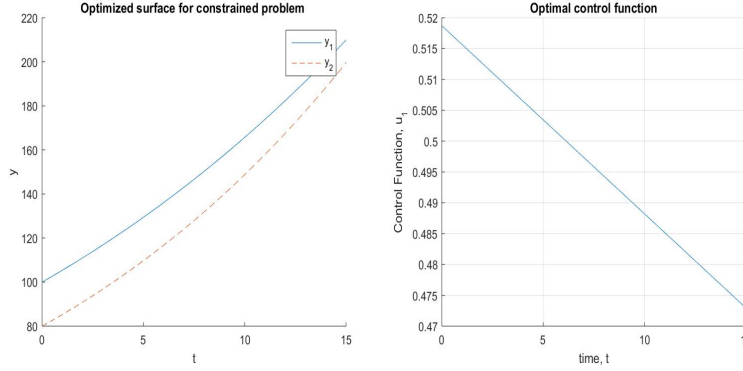


Fig. 6 Optimal surface and control functions for Model 1. [SurrogateModel_SciencePolicy](#)

2 Model 2: Population Dynamics

2.1 Description of the Model

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

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

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 section 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 $u^* = [u_1^*, u_2^*]$ 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

$$\psi_0(u^*) = \min_u \int_{t_0}^T (y_1(t, u) - y_{1d})^2 dt \quad (21)$$

subject to state equation (20), box constraints on the control

$$U = \{u : u^-(t) \leq u(t) \leq u^+(t)\} \quad (22)$$

and pointwise constraint on state

$$y_2(t) \leq y_2^+ \quad (23)$$

The pointwise state constraint (23) is transformed into an equivalent equality constraint of the integral type

$$\psi_1(u) = \tilde{\psi}_1(u, y(u)) = \int_{t_0}^T (|y_2(t, u) - y_{2d}| + y_2(t, u) - y_{2d})^2 dt \quad (24)$$

Taking into account equations(21-24) the optimization problem can be written in a reduced form as follows:

$$\begin{aligned} \psi_0(u^*) &= \min_{u \in U_\partial} \int_{t_0}^T (y_1(t, u) - y_{1d})^2 dt \\ U_{\partial u} &= \{u : u \in U; \psi_1(u) = \tilde{\psi}_j(u, y(u)) = 0\} \\ c(y(u), u) &= 0 \end{aligned} \quad (25)$$

2.2 Simulation

For numerical experiments we select the following values of the input parameters: $[p_1, p_2, p_3, p_4, p_5] = [0.734, 0.175, -0.500, -0.246, 0.635]$, $[t_0, T] = [0, 10]$, $n_u = 2$, $u^- = [u_1^-, u_2^-] = [-0.5500, -1.0370]$, $u^+ = [u_1^+, u_2^+] = [-0.300, -0.7870]$, $y_{1d} = 5$, $y_2^+ = 6$. 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: $b = [b_1^{(1)}, b_2^{(1)}, b_1^{(2)}, b_2^{(2)}]^T = [b_1, b_2, b_3, b_4]^T$.

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 $n_y = 2$ was 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 was 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 as the rank of the POD basis as shown by the figure of singular values in figure 7. 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 3 approximation is very accurate and adding more vectors (by increasing rank) in approximation further increases the precision.

Having chosen $k = 3$, the numerical simulations are performed for model (20). For testing of the model, $n_g = 10$ points were used to calculate the RMAE for

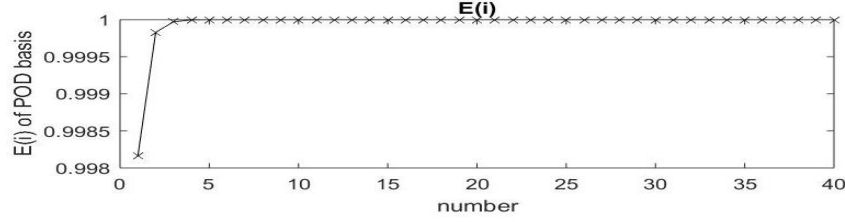


Fig. 7 Cumulative energy plot to determine singular values for Model 2. [Surrogate-Model_PopulationDynamics](#)

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 was 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 8. For this point, all POD-RBF gave relative maximum absolute error less than 1% as desired.

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 |

2.3 Optimization

In previous subsection, 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 (25) and the results are summarized in table 6. For simplicity, the number 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 was met in the first iteration, with error between approximated and actual responses being less

than 0.01 in first iteration. Hence, the desired accuracy was achieved and no further iterations were required. Also, 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 was 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.

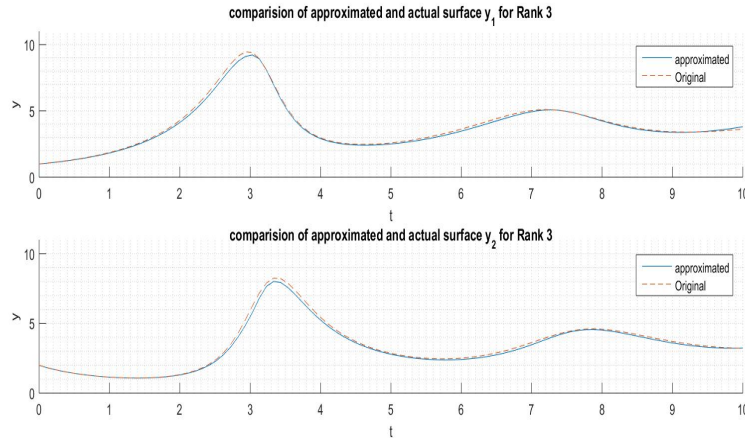


Fig. 8 Actual vs approximated surface of Model 2. [SurrogateModel_PopulationDynamics](#)

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 |
| Time _{orig} | 23.3983 sec | Time _{surr} | 6.6241 sec |
| Time _{cnsr} | 15.3470 sec | ϵ | 0.0081 |

Chapter 5

Conclusions

This research employed Proper Orthogonal Decomposition (POD), a surrogate modeling technique integrated in optimization framework for dimension reduction by extracting hidden structures from high dimensional data and projecting them on lower dimensional space. In the first instance, POD was coupled with various Radial Basis Functions (RBF)— a smoothing technique— and the computational procedure was hypothesized to provide compact, accurate and computationally efficient solution of optimal control problems. The surrogate models using POD-RBF were constructed. The computational procedure of surrogate model was divided into problem setup and training/testing phase for effective implementation of the reduced order modeling techniques. Furthermore, an iterative algorithm was introduced methodically to achieve more accurate results.

The algorithm and computational procedure was implemented on two real-life optimal control problems that were taken directly from literature sources. It was 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 was 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 was more precise if the model was 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 is in coherence with the theory.

In solution of optimization problems, the system responses obtained by surrogate model invariably gave accurate results with improved computational time. As a whole, both the 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 computational performance 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.

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 was based on the same notion. However, while analyzing the performance POD-RBF procedure on non-linear dynamical systems in the last chapter of this thesis, it was brought into consideration that the even though the optimization process itself was faster with surrogate responses, their construction was 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 were considered and elaborated throughout the analysis and the scope of extension of this research was discussed alongside.

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 Computational Science* 11 (2015), pp. 102–111.
- [6] 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.
- [7] V. Bujlak. *Inverse Analysis with Model Reduction (Proper Orthogonal Decomposition in Structural Mechanics)*. Milan: Springer, 2012.
- [8] A. Chatterjee. “An Introduction to the Proper Orthogonal Decomposition”. In: *Current Science* 78.7 (Apr. 2000).
- [9] Ying Chen et al. “Risk related brain regions detection and individual risk classification with 3D image FPCA”. In: *Statistics & Risk Modeling* 35.3–4 (1Jul. 2018), pp. 89–110. doi: 10.1515/strm-2017-0011.
- [10] L. Chrisman. *Latin Hypercube vs. Monte Carlo Sampling*. July 2014.
- [11] 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.
- [12] M.D. Intriligator. “Applications of Optimal Control Theory in Economics”. In: *Synthese* 31.2 (1975), pp. 271–288.
- [13] 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.
- [14] E. Iulianoa and D. Quagliarella. “Proper Orthogonal Decomposition, surrogate modelling and evolutionary optimization in aerodynamic design”. In: *Computers and Fluids* 84 (2013), pp. 327–350.
- [15] 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.

- [16] 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: 10.1088/09641726156036.
- [17] 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.
- [18] A. Maravalle and L. Rawdanowicz. “Changes in Economic and Financial Synchronisation”. In: 1517 (2018).
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] Ngoc 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.
- [25] 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.

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.