INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN BIOMEDICAL ENGINEERING
Int. J. Numer. Meth. Biomed. Engng. (2014)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cnm.2646

Emulating facial biomechanics using multivariate partial least squares surrogate models

Tim Wu[1,*,†], Harald Martens[2], Peter Hunter[1] and Kumar Mithraratne[1]

[1]*Auckland Bioengineering Institute, The University of Auckland, Auckland, New Zealand*
[2]*Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway*

SUMMARY

A detailed biomechanical model of the human face driven by a network of muscles is a useful tool in relating the muscle activities to facial deformations. However, lengthy computational times often hinder its applications in practical settings. The objective of this study is to replace precise but computationally demanding biomechanical model by a much faster multivariate meta-model (surrogate model), such that a significant speedup (to real-time interactive speed) can be achieved. Using a multilevel fractional factorial design, the parameter space of the biomechanical system was probed from a set of sample points chosen to satisfy maximal rank optimality and volume filling. The input–output relationship at these sampled points was then statistically emulated using linear and nonlinear, cross-validated, partial least squares regression models. It was demonstrated that these surrogate models can mimic facial biomechanics efficiently and reliably in real-time. Copyright © 2014 John Wiley & Sons, Ltd.

Received 2 November 2013; Revised 5 February 2014; Accepted 2 April 2014

KEY WORDS: facial expression; meta-modelling; multivariate regression; surrogate model

1. INTRODUCTION

Current knowledge of mechanisms that generate expressive facial movements is mostly based on dissection and imaging (both photographic and medical) studies, and therefore, the underlying intricate biomechanical interactions remain a not well-understood subject. As computer capacity advances, it is becoming increasingly viable to study complex biological systems using highly detailed computational models. In this context, a detailed biomechanical model of the face will allow researchers to fully understand the mechanisms involved in facial expressions.

One of the first muscle-driven face model was developed by Waters and Terzopulos [1], which allowed facial expressions to be shaped by the control of the underlying musculature. However, the expressions generated from the early models were relatively simple and looked artificial. In addition, the manipulations of the parameters that control these models were often complex and unintuitive, and hence difficult for inexperienced users. In the recent years, more complex and biomechanically accurate face models have been developed for various applications. For example, Sifakis *et al.* [2] created a detailed finite element (FE) model of the face to simulate facial expressions while incorporating collision events and environmental factors. Ramirez-Valdez and Hasimoto-Beltran [3] used biomechanically derived facial expressions in their facial screening database to improve the chance of detection. In the medical sector, there are on-going studies to create more sophisticated facial models for pre-surgical planning and to predict long-term post-surgical outcomes (both aesthetic and functional outcomes), see, for example, [4–8]. On a similar

*Correspondence to: Tim Wu, Auckland Bioengineering Institute, The University of Auckland, Auckland, New Zealand.
†E-mail: twu051@aucklanduni.ac.nz

note, biomechanical models have also been used to investigate speech articulation and phonological processes [9, 10].

In order to make realistic predictions, one needs to use complex biomechanical models with lengthy computational times. As a result, they are not appropriate for applications requiring real-time performances (e.g. interactive animation, real-time facial screening and surgical navigation). In this paper, an alternative approach using simplified surrogate models (multivariate meta-models) is described. Using a suitable surrogate model, real-time and interactive performances can be accomplished. A surrogate model is capable of handling both the forward and inverse modelling problems. In the forward modelling (classical meta-modelling), one develops a statistically based surrogate model (output = f(input)) using simulation data, and then uses it to evaluate the system's unknown output from new values of the input. Conversely, in the inverse meta-modelling, the surrogate is used to predict the system's unknown inputs from new measurements of output variables (i.e. input = f(output)). Regardless of the application, it is essential for the surrogate model to mimic the characteristics of the underlying physical (biomechanical) model, and therefore, the global behaviour of the system can be described in a less expensive manner.

In order to implement the surrogate-based approach, we require a biomechanical system that simulates facial biomechanics. The 3D geometry of the facial model used in this study is derived from MRI data of a healthy 26-year old male volunteer. Its biomechanics is characterised by the theories governing finite deformation elasticity and nonlinear contact mechanics. The elastic difference between the muscle and adipose tissues is accounted for by treating the soft tissue continuum as a heterogeneous medium, and its mechanical behaviour is represented using a modified two-parameter Mooney–Rivlin constitutive model [11]. Soft tissue deformations are driven by the contraction of the underlying network of 3D muscles. Table I summarises a list of facial muscles that were used in the current implementation; the annotated figure of the muscle geometries are shown in Figure 1. The phenomenologically derived muscle fibre model is controlled by a scalar value that represents the level of muscle activation (normalised between zero and one). These fibre forces are integrated into the computational domain through a novel FE mapping procedure that ensures the complex 3D geometry of the muscle is preserved [12]. Using this model, realistic facial expressions can be generated by activating relevant muscles as shown in Figure 2. For a more comprehensive description of the biomechanical model, the reader is referred to [13].

The paper is organised as follows. In Section 2, the idea of surrogate models built upon statistical modelling theory is introduced. With an appropriate but simple mathematical meta-model function, the underlying mechanics of the face model can be approximated more efficiently. Section 3

Table I. Facial muscle names and abbreviation used in biomechanical simulations.

Muscle name	Abbreviation
Buccinators	BUC
Corrugator supercilii	COR
Depressor anguli oris	DAO
Depressor labii inferioris	DLI
Depressor supercilii	DES
Frontalis	FRO
Levator anguli oris	LAO
Levator labii superioris	LLS
Levator labii superioris alaeque nasi	LLSAN
Mentalis	MEN
Orbicularis oculi (orbital part)	OOC-O
Orbicularis oculi (palpebral part)	OOC-P
Orbicularis oris	OOR
Platysma	PLA
Procerus	PRO
Risorius	RIS
Zygomaticus major	ZMA
Zygomaticus minor	ZMI

DOI: 10.1002/cnm

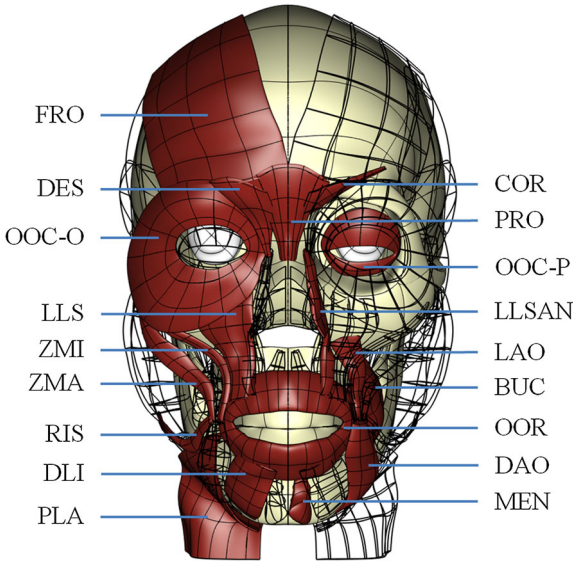

Figure 1. Finite element geometries of the facial muscles. See Table I for the list of abbreviated terms.

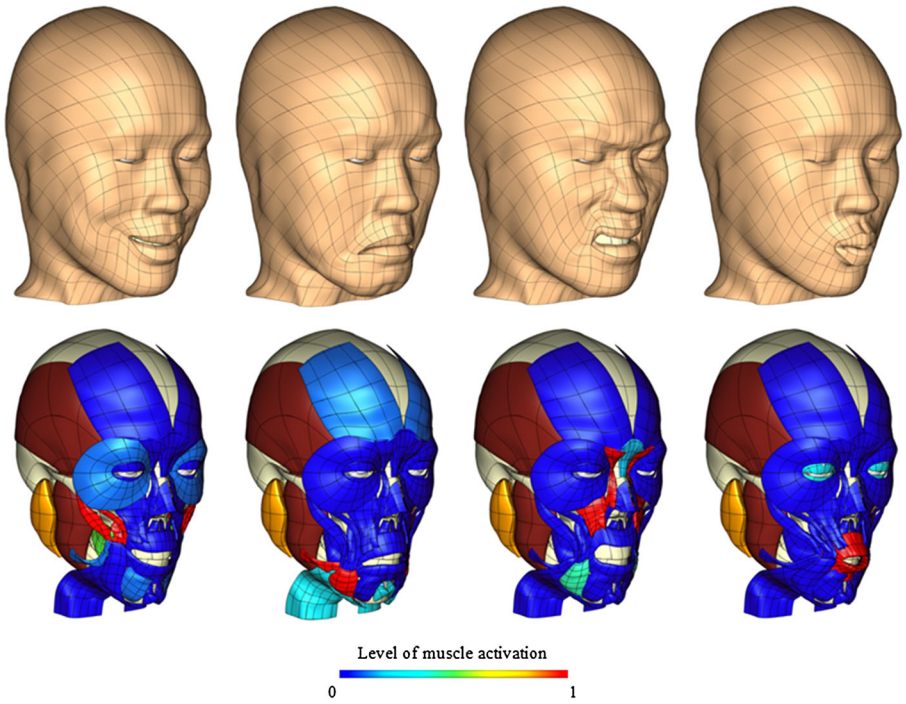

Figure 2. Biomechanical simulations of the expressions of (from left to right) joy, sadness, snarl and the kissing gesture, by activating the corresponding facial muscles.

describes the sampling procedure to obtain the statistical data required to train the surrogate model. A novel multilevel fractional factorial design proposed by [14] is employed in this study. Multivariate partial least square regression (PLSR) method is used to determine the characteristics of the surrogate model. Section 4 outlines the process of training and validating the PLSR surrogate. Four forms of PLSR are considered, including the classical linear form and three nonlinear variants. The performance and speed of each surrogate model is benchmarked, and the results are presented in Section 5. In Section 6, the applications of the surrogate model with examples demonstrating the forward modelling problem are highlighted. Finally, conclusions drawn from the study are presented in Section 7.

DOI: 10.1002/cnm

2. SURROGATE-BASED MODELLING APPROACH

Consider a physically based model $\boldsymbol{M}$, in which $\boldsymbol{M}$ is controlled by a set of input parameters (independent variables) and yields a set of output parameters (dependent variables). A surrogate model (or meta-model) of $\boldsymbol{M}$ is based on a statistical modelling approach, where the behaviour of model $\boldsymbol{M}$ is characterised using a relatively simple mathematical meta-model function, whose parameter values are estimated from simulation results that were obtained from statistically designed numerical experiments (Figure 3).

As far as the facial biomechanical model is concerned, the activation level of various facial muscles represent the independent input variables, whereas the mesh nodal DOF that describe the geometric deformation are the dependent output variables (mesh nodes). The practice of creating a surrogate model generally follows a two-step process. Firstly, a sampling procedure is performed where the entire input design space is explored from a manageable number of observation points. It is a major challenge in meta-modelling to determine a design that is as accurate and reliable as possible with the minimal investment in obtaining new observations of the system. Once an adequately defined set of observations is acquired, the appropriate mathematical function is then chosen, and its parameters are estimated to allow optimal description of the input–output relationship. It is crucial for the function to capture most of the nonlinearity and interaction of the system, hence making it useful for predicting new data.

Least squares estimators have been shown to be effective in the context of surrogate-based meta-modelling. Some common examples are the polynomial regressors [15], moving least squares methods [16], Gaussian radial basis functions [17] and its variant, the Kriging method [18]. More recently, the support vector predictor [19] is becoming increasingly popular because of its ability to handle noisy data. The interested reader is referred to [20, 21] for a comprehensive overview of these methods. In the present study, however, the multivariate PLSR method was applied. Despite being less common in the field of engineering, variants of the PLSR are a popular technique adapted in chemometrics and related areas. The PLS approach allows particularly relevant co-variation subspaces to be extracted from two or more data matrices. It serves two important purposes; firstly, it is used to reduce the dimension of the system while preserving most significant information (i.e. shrinkage estimator). Secondly, the PLS method removes multi-collinearity among the regressor and regressand variables. The multi-collinearity or near linear dependence of the independent variables is a serious problem for traditional full-rank least squares estimators as it results in inflated uncertainty variances and covariances of the regression coefficients. Specifically, the values and signs of estimated coefficients may change considerably given different training data from the same system, hence leading to the possibility that a model that fits the training data may generalise poorly to new data (Montgomery *et al*., 2006).

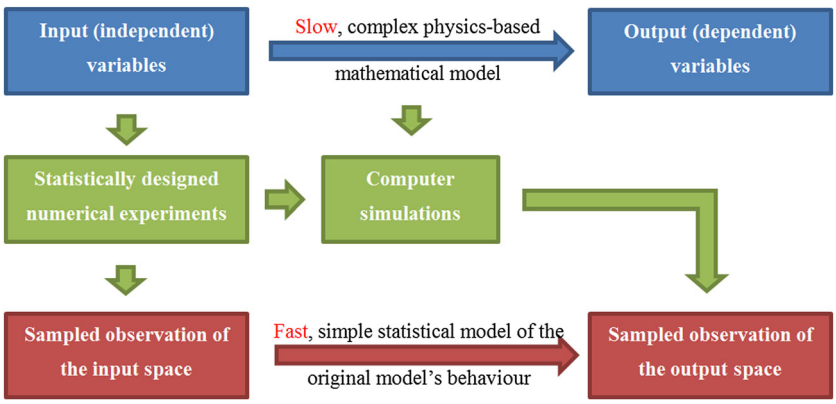

Figure 3. Diagrammatic representation of a conventional deterministic physical model (blue path), and its adaptation to the data-driven surrogate model (red path) via designed numerical experiments and statistical regression modelling of the simulation data.

DOI: 10.1002/cnm

3. DESIGN OF EXPERIMENTS

In statistics, design of experiments is the planning of information-gathering (sampling) exercises where the relevant variations of the system are investigated under a controlled environment. In terms of computer simulations, the design defines where to probe a system in its input design space [22]. The primary interest when designing the experiments is that the variance of sampling points must be minimised, thus maximising the relevant information that the sample represents. An ideal design of experiments is one that distributes the sample points uniformly in the design space; specifically, the variance can be minimised by maximising the minimum distances between design points [23]. A uniformed design may be achieved from a full factorial approach, where each input variable (denoted as factor) is discretised into two or more possible levels of value, and the system output at all possible combinations of these levels across all factors is evaluated. However, the computational cost of such a design often limits their practicality. To make the simulation counts more manageable, while maintaining the benefits of the factorial approach, fractional factorial designs are frequently employed. In the fractional factorial experiments, most of the combinations are omitted based on certain statistical assumptions. Early fractional reductions were mainly intended for factors with only two levels, where the reductions were based on the sparsity-of-effects principle (i.e. 2^{K-p} designs, where K is the number of factors and 2^{-p} is the reduction ratio). Although these two-level designs are sufficient in investigating the primary effects of factors, they have limited ability to reproduce higher-order nonlinear behaviours.

In this study, a multilevel extension to the 2^{K-p} factorial design was created using the binary replacement method proposed by Martens *et al*. [14]. The multilevel factorial design is employed as it is believed that, compared to e.g. random sampling, the factorial approach in general facilitates the spanning of the high-dimensional input factor space more readily. Moreover, contrary to computer experiment designs generated from the Latin Hypercube method, it also accommodates the modelling of nonlinear input–output relationships by splitting the parameter variables into equally spaced levels.

3.1. Multilevel binary replacement (MBR) method

Consider a complex unknown system with K input factors, where these factors are assessed at a number of levels $L(k)$ in a factorial design. With little loss of generality, the number of levels for each factor is chosen to be power of two: $L(k)=2^{M(k)}$, where the full factorial combination would be $N=2^{\sum M(k)}$. In the multilevel binary replacement (MBR) design, the input variable $x_{k,l}$ of factor k and level l can be represented in the index form $d_{k,l}$ (e.g. $\mathbf{x}_k=[0,0.25,0.5\ldots]$ is mapped into an equally spaced indexing variable $\mathbf{d}_k=[1,2,3,\ldots]$), which can be further recoded into a binary variable $\mathbf{f}_{k,l}$ that has $M(k)$ factor bits, e.g. $\mathbf{f}_{k,l}=[f_{k,l,1},\ldots,f_{k,l,M(k)}]$. For instance, a value $d_{k,l}=5$ in an eight-levelled factor ($M(k)=3$) can be written as $[f_{k,l,1}, f_{k,l,2}, f_{k,l,3}]=[1,0,1]$, where the bits (0 or 1) can be further recoded into two-level replacement design factors $\mathbf{g}_{k,l}$ with values -1 or 1 in accordance with standard factorial design procedure, that is, $[g_{k,l,1}, g_{k,l,2}$ and $g_{k,l,3}]=[1,-1,1]$. Equation (1) depicts the relationship between $d_{k,l}$, $\mathbf{f}_{k,l}$ and $\mathbf{g}_{k,l}$, which are equivalent representation of the design.

$$\begin{aligned} d_{k,l} &= \sum_{m=1}^{M(k)} 2^{m-1} f_{k,l,m} \\ \mathbf{g}_{k,l} &= \mathbf{f}_{k,l} \times 2 - 1 \end{aligned} \tag{1}$$

As a result, the multilevel design ($L(k)>2$) can be investigated as a two-level factorial system where standard 2^{K-p} reduction methods can be applied. It should be noted that while the MBR method is relatively new, it has been implemented into computer experiments of a mammalian circadian clock model, which demonstrated its potential for highly nonlinear problems [24].

DOI: 10.1002/cnm

3.2. The optimised multilevel binary replacement design

The facial expression modelling problem used in this study contains $K = 18$ factors (representing 18 facial muscles). To capture the nonlinearity of the system, each factor was discretised into four equally spaced activation levels ($\forall\ k : L(k) = 4$) ranging between 0 and 1, that is, $\mathbf{x}_k = [0,1/3,2/3,1]$. In a full factorial design, this would give $N = 4^{18} \approx 6.8 \times 10^{10}$ possible combinations. Here, a strongly reduced simulation design with only 128-parameter combinations was employed, namely, $N = 128 = 2^{2\times\ 18-29}$ with a reduction ratio of 2^{-29}.

The optimised MBR design in this study was optimised according to the following procedure:

1. Create a new confounding pattern by randomly re-assigning two-level factors $\mathbf{g}_{k,l}$ to the standard $2^{K\text{-}p}$ design method.
2. Convert back to original factor variables $\mathbf{x}_k$.
3. Determine an optimisation criterion based on $\mathbf{x}_k$.
4. Repeat several times (5000 repetitions were computed in this study) and choose the one with maximum criterion value.

The criterion used in this study is a combination of maximal rank optimality and maximal volume filling. The maximal rank optimality is calculated by summing the eigenvalues of the quadratic information matrix, revealing the design with maximum variance in the main effects as well as some (as many as possible sets of) two-factor interactions. On the other hand, the maximal volume filling function identifies the design with minimal number of large empty holes in the parameter space, based on the pairwise combinations of these $K = 18$ parameters. It is important to note that, unlike in the standard $2^{K\text{-}p}$ methods, the optimisation criterion is calculated based on the values of quantitative design factors ($\mathbf{x}_k$), not the two-levelled counterparts ($\mathbf{g}_{k,l}$). This is to compensate for the fact that the levels in a design factor $\mathbf{x}_k$ may not be evenly spaced, and in addition, the responses are expected to behave differently when changes are made within a design factor $\mathbf{x}_k$ and at different levels of other design factors $\mathbf{x}_{k^*\neq k}$. The optimal MBR design generated from the described procedure is plotted pairwise in Figure 4. Note that the design gives an almost complete coverage of all possible pairwise interactions, except between the frontalis (FRO) and palpebral part of the orbicularis oculi (OOC-P) muscles.

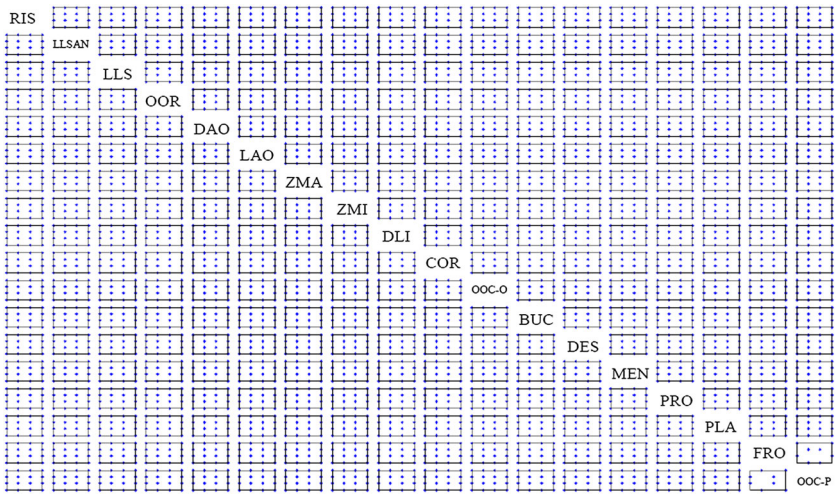

Figure 4. Pairwise plot of the optimised multilevel binary replacement design used to generate statistical model of facial expressions and gestures. This reduced design with $2^{2\times\ 18-29} = 128$ simulations spans 18 factors where each factor was discretised into four levels (i.e. the levels of activation for 18 considered facial muscles). The muscles abbreviation are given in Table I, and they are sorted from the most likely (top-left) to the least likely muscles being recruited from a predefined set of expressions and gestures (smile, laugh, sadness, anger, terror, crying, disgust, snarl and kissing).

DOI: 10.1002/cnm

4. MODELLING RELATION

In order to generate an accurate and reliable surrogate model, all of the available simulation data were employed. This included the 128 simulations for the optimised MBR design (Section 3.2) and 18 simulations of individual muscle actions. In addition, to further extend the sample space, converged intermediate solutions obtained from the load stepping strategy [25] were also used. The load stepping strategy was necessary for the biomechanical simulations to converge. Specifically, instead of directly applying the maximum level of muscle activations, they were applied sequentially in small increments, and the solution obtained from the previous iteration is used as the initial guess for the next increment. The converged solutions from each of the incremental step were then added to the overall sample database. Together, the original 128 end point parameter solutions and the intermediate increment solutions generated 6081 observations of the system. These input/output data were employed to train the surrogate model.

Methods of modelling the input–output relations can be broadly classified into parametric and non-parametric form. The parametric methods presume a global functional form of the relationship between the independent and dependent variables, while in non-parametric methods, simple local models are applied in different regions of the data to build up an overall surrogate. This study explores both the parametric and non-parametric methods based on PLSR.

4.1. Partial least squares regression method

The PLSR [26] is a method for modelling relations between two sets of observed variables **X** and **Y** (e.g. the input parameters and the output responses) by means of estimated latent variables [27]. In contrast to the more well-known principal component regression, which first compresses matrix **X** into a reduced set of orthogonal components by means of principal component analysis and then uses these components from **X** as regressors for **Y**, the PLSR creates the components by modelling the relationship between the **X** and **Y**, thereby ensuring that the components extracted from **X** are more relevant to **Y**. Previous studies have demonstrated that PLSR achieves the same or better results as principal component regression while using significantly fewer, and qualitatively different components [28, 29].

Consider a $(N\times K)$ matrix of the mean-centred input space (**X**) and a $(N\times J)$ matrix of the mean-centred output space (**Y**), where K is the number of independent variables (factors) per observation, J is the number of dependent variables per observation and N is the number of the observations. The variables within X and **Y** are usually scaled to have approximately equal expected uncertainty. The PLSR method then decomposes **X** and **Y** matrices into bilinear structure models consisting of linear combinations of score and loading matrices.

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} \tag{2}$$

$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^{\mathrm{T}} + \mathbf{F} \tag{3}$$

where **T**(=**XV**) is a $(N\times A)$ matrix of A extracted score vectors (or components) from **X**, obtained via weight matrix **V** $(K\times A)$ that maximises the explained covariance of **X** and **Y**. The $(K\times A)$ **P** matrix and the $(J\times A)$ **Q** matrix are the loading matrices, and **E** and **F** are residual matrices (i.e. the unexplained parts of **X** and **Y**, respectively, after A components are extracted). The PLS estimation principle ensures that the X-scores, **T** are Y-relevant by using a temporary model of **Y** (Equation (4)) inside the estimation algorithm:

$$\mathbf{Y} = \mathbf{U}\mathbf{C}^{\mathrm{T}} + \mathbf{F} \tag{4}$$

where **U** is a $(N\times A)$ matrix that represents Y-score vectors (linear combinations of the Y-variables). It shall be noted that Equation (4) is only used for preliminary estimation guidance, and not in the final model. The PLSR method assumes that a known inner relation between the **Y** and **X**-scores exists: that

DOI: 10.1002/cnm

is, $\mathbf{U} = f(\mathbf{T}) + \mathbf{H}$, where $\mathbf{H}$ denotes the matrix of residual resulted from this inner relation mapping. Replacing $\mathbf{U}$ in Equation (4) by $\hat{\mathbf{U}} = f(\mathbf{T})$, the PLSR model can be rewritten as

$$\mathbf{Y} = f(\mathbf{T}) \times \mathbf{C}^{\mathrm{T}} + \mathbf{F}^{*} \tag{5}$$

with $\mathbf{F}^{*}$ being the combined residuals from the decomposition and the inner relation mapping. In this study, the process of determining $\mathbf{T}$, $\mathbf{U}$, $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{C}$ and $\mathbf{V}$ from the training data was performed using the nonlinear iterative PLS (NIPALS) algorithm [30].

4.1.1. Classical linear form. The classical form of the PLS regression method originally presented by [26] and elaborated by [31] was based on a linear inner relation between the scores: $\mathbf{u}_a = d\mathbf{t}_a + \mathbf{h}_a$, where d is a scalar regression coefficient and $\mathbf{u}_a$, $\mathbf{t}_a$ and $\mathbf{h}_a$ are the column vectors of the matrices $\mathbf{U}$, $\mathbf{T}$ and $\mathbf{H}$ associated with the a^{th}-component. Equivalently, in the matrix form, the linear inner relation can be written as

$$\mathbf{U} = \mathbf{TD} + \mathbf{H} \tag{6}$$

with $\mathbf{D}$ being the $(A \times A)$ diagonal matrix of regression coefficients. The PLSR is an iterative process; starting from the original data matrices ($\mathbf{X}$ and $\mathbf{Y}$), the classical linear form finds score vectors $\mathbf{t}_{a=1}$ and $\mathbf{u}_{a=1}$ with maximum covariance. The subsequent components are chosen such that $\mathbf{T}$ is orthogonal ($\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_j = 0$, $i \neq j$), and in addition, each $\mathbf{t}_a$ and the corresponding $\mathbf{u}_a$ are maximally covariated. In NIPALS algorithm, orthogonality is ensured by deflating the data matrices (Equation (8)) every time after a component has been extracted, before operating on the subsequent component. The steps for determining the components of each rank are as follows (for brevity, the subscript a is neglected hereon).

1. Initialise $\mathbf{u}$, for example, as the column vector of $\mathbf{Y}$ with the maximum Euclidean norm.
2. Estimate loading weights $\mathbf{w}$ by regressing columns of $\mathbf{X}$ on $\mathbf{u}$ and then normalise to unit length: $\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{u}$, $\|\mathbf{w}\| \rightarrow 1$.
3. Calculate the input score vector: $\mathbf{t} = \mathbf{Xw}$.
4. Estimate $\mathbf{c}$ by regressing columns of $\mathbf{Y}$ on $\mathbf{t}$ and then normalise to unit length: $\mathbf{c} = \mathbf{Y}^{\mathrm{T}}\mathbf{t}$, $\|\mathbf{c}\| \rightarrow 1$.
5. Calculate the output score vector: $\mathbf{u} = \mathbf{Yc}$.
6. Repeat steps 2–5 until convergence is achieved.

After obtaining the converged scores, the loading vectors ($\mathbf{p}$ and $\mathbf{q}$) and the regression coefficient (d) can then be computed as

$$\begin{aligned} \mathbf{p} &= \frac{\mathbf{X}^{\mathrm{T}}\mathbf{t}}{\mathbf{t}^{\mathrm{T}}\mathbf{t}} \\ d &= \frac{\mathbf{u}^{\mathrm{T}}\mathbf{t}}{\mathbf{t}^{\mathrm{T}}\mathbf{t}} \\ \mathbf{q} &= d\mathbf{c} \end{aligned} \tag{7}$$

Subsequently, the training data matrices are deflated (Equation (8)) and repeated for extraction of the succeeding component, until the maximum number of components are reached ($A \leq \min(K,J)$) or the required tolerance on the residuals ($\mathbf{E}$ and $\mathbf{F}$) has been achieved.

$$\begin{aligned} \mathbf{X} &\leftarrow \mathbf{X} - \mathbf{tp}^{\mathrm{T}} \\ \mathbf{Y} &\leftarrow \mathbf{Y} - \hat{\mathbf{u}}\mathbf{c}^{\mathrm{T}} \end{aligned} \tag{8}$$

In the NIPALS algorithm, the parameters $\mathbf{w}$, $\mathbf{t}$, $\mathbf{p}$, $\mathbf{c}$, $\mathbf{q}$ and d for all components $a = 1,2,\ldots,A$ are collected in matrices $\mathbf{W}$, $\mathbf{T}$, $\mathbf{P}$, $\mathbf{C}$, $\mathbf{Q}$ and $\mathbf{D}$. Moreover, the weight matrix $\mathbf{V}$ can be expressed as a linear transformation of the orthonormal loading weights $\mathbf{W}$:

DOI: 10.1002/cnm

$$\mathbf{V} = \mathbf{W}\left(\mathbf{P}^{\mathrm{T}}\mathbf{W}\right)^{-1} \tag{9}$$

Once all significant components are extracted, the model can then be used to predict new data using the following relationship:

$$\begin{aligned}\mathbf{Y} &= \mathbf{T}\mathbf{Q}^{\mathrm{T}} + \mathbf{F} \\ &= \mathbf{X}\mathbf{V}\mathbf{D}\mathbf{C}^{\mathrm{T}} + \mathbf{F}^{*} \\ &= \mathbf{X}\mathbf{B} + \mathbf{F}^{*}\end{aligned} \tag{10}$$

where B is a (K× J) matrix of regression coefficients.

4.1.2. Nonlinear extension. While the classical form of PLS may be a simple surrogate model, if the nonlinearity of the system is severe, the linear approximations are often inadequate and do not achieve the required accuracy. The nonlinear extension of the PLSR method can be distinguished into two distinct forms. In accordance with [32], they are denoted as Type I and Type II forms. The Type I PLSR first applies a nonlinear transformation to the observed variables, and subsequently, a linear PLS model is used to relate the transformed variables. However, despite the ease of implementation, a drawback in the Type I approach is that it represents a 'black-box' model, making it difficult for the user to interpret the results. To overcome this problem, Type II PLSR employs a more general approach, where a nonlinear inner relation is constructed without transforming the observed variables, therefore addressing the problem of loss of interpretability. In this study, both Type I and Type II methodologies were examined.

Type I: nonlinear transformation of observed variables

Assuming that a nonlinear relationship exists between the independent and dependent variables: $\mathbf{x} \xrightarrow{f} \mathbf{y}$. This relation can be equally represented using a two steps mapping process: $\mathbf{x} \xrightarrow{\phi} \boldsymbol{\varphi} \xrightarrow{f1} \mathbf{y}$, where the original input data ($\mathbf{x} \in \mathbb{R}^{K}$) are first mapped to a higher dimensional feature space ($\varphi(\mathbf{x}) = \boldsymbol{\varphi} \in \mathbb{R}^{K*}$, $K* > K$) before transforming to $\mathbf{y}$. It is then easily conceived that there exist pairs of ϕ and f_1 functions, where all the nonlinear behaviours are accounted for in φ and thus f_1 becomes a simple linear mapping between $\boldsymbol{\varphi}$ and $\mathbf{y}$. However, in practise, it is difficult to find such pairs of mappings as the form of the nonlinear relation between the original independent and dependent variables is often unknown. Nevertheless, following the idea that in a higher dimension (where the original variables are mapped), the nonlinearity of the system is reduced, the resulting relation (between $\boldsymbol{\varphi}$ and $\mathbf{y}$) can therefore be approximated linearly (using linear NIPALS algorithm) with lesser residual values. A common mapping function is the quadratic surface projection [33], where the original data are extended by considering the squares and cross products of the entries.

$$\mathbf{x} = (x_1, \ldots, x_k) \rightarrow \boldsymbol{\varphi} = \left(x_1, \ldots, x_k, x_1^2, \ldots, x_k^2, x_1x_2, \ldots, x_{k-1}x_k\right) \tag{11}$$

The Type I principle is simple and straightforward as there are no algorithmic modifications to the classical NIPALS algorithm. However, there are some limitations that prevent the general usage of this form of nonlinear modelling. Firstly, because the PLSR is performed between transformed variables, not the originally observed variables, it may sometimes be difficult for the user to interpret the results and comment on the regression coefficients. Another drawback of Type I principle is related to the expansion in the dimensionality of the regressor. As the dimensionality increases, more orthogonal components are required to capture the characteristics of the mapped input space $\boldsymbol{\varphi}$, which results in higher computational costs. In addition, because the regression model is fitted on the expanded input space, in a least squares sense, its generalisation to the original observed data in the lower dimension may not be optimal. On a similar note, an adequately dense

DOI: 10.1002/cnm

sampling strategy in original space may become too sparse in higher dimensions, and therefore also results in poor generalisation of the model.

Type II: nonlinear inner relation

In contrast to Type I nonlinear PLSR models, in the Type II approach, the assumption that the score vectors **t** and **u** are linear projection of the original variables **x** and **y** is maintained. Instead, a continuous nonlinear function is introduced to replace the linear inner relation (Equation (6)). As a consequence of a different inner relation, some modifications to the classical NIPALS algorithm are required. First, it can be observed from the linear formulation (see the NIPALS algorithm in Section 4.1.1) that the weight vector **w** corresponds to the covariance between the output score vector **u** and the input data **x**. However, in the nonlinear case, **w** represents the nonlinear association between **u** and **x**, and is only related to covariance if the nonlinear function is monotonic and slightly nonlinear. As a consequence, the nonlinear update of the weight vector $\Delta\mathbf{w}$ needs to be considered at each inner iteration of the NIPALS algorithm [34].

$$\Delta\mathbf{w} = \left(\frac{\partial\hat{\mathbf{u}}^{\mathrm{T}}}{\partial\mathbf{w}}\frac{\partial\hat{\mathbf{u}}}{\partial\mathbf{w}}\right)^{-1}\left[\frac{\partial\hat{\mathbf{u}}}{\partial\mathbf{w}}\right]^{\mathrm{T}}(\mathbf{u}-\hat{\mathbf{u}}) \tag{12}$$

where $\hat{\mathbf{u}} = f(\mathbf{t},\boldsymbol{\alpha})$ is a nonlinear function of **t**, fitted to the function parameters $\boldsymbol{\alpha}$. Based on this update scheme, the modified NIPALS algorithm with nonlinear inner relation is given by the following steps:

1. Initialise **u**, for example, as the column vector of **Y** with the maximum Euclidean norm.
2. Initialise **w** based on the linear preliminary model (i.e. $\mathbf{X} = \mathbf{u}\mathbf{w}^{\mathrm{T}}$) and then normalise to unit length: $\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{u}$, $\|\mathbf{w}\| \to 1$.
3. Calculate the input score vector: $\mathbf{t} = \mathbf{X}\mathbf{w}$.
4. Fit the nonlinear inner relation to obtain $\hat{\mathbf{u}}$: $\boldsymbol{\alpha} \leftarrow \mathrm{fit}_{\boldsymbol{\theta}}[\mathbf{u} = f(\mathbf{t},\boldsymbol{\alpha}) + \mathbf{h}]$, $\hat{\mathbf{u}} = f(\mathbf{t},\boldsymbol{\alpha})$.
5. Estimate **c** by regressing columns of **Y** on $\hat{\mathbf{u}}$ and then normalise to unit length: $\mathbf{c} = \mathbf{Y}^{\mathrm{T}}\hat{\mathbf{u}}$, $\|\mathbf{c}\| \to 1$.
6. Calculate the output score vector: $\mathbf{u} = \mathbf{Y}\mathbf{c}$.
7. Update **w** using $\Delta\mathbf{w}$ defined in Equation (12) and normalise: $\mathbf{w} = \mathbf{w} + \Delta\mathbf{w}$, $\|\mathbf{w}\| \to 1$.
8. Repeat steps 3–7 until convergence.

After the convergence of the inner loop, the X-loading vector **p** is computed using Equation (7), and the data matrices are deflated (Equation (8)) for the extraction of subsequent components. The nonlinear PLSR model is then described as

$$\begin{aligned}\mathbf{Y} &= \mathbf{T}\mathbf{Q}^{\mathrm{T}} + \mathbf{F}\\ &= f(\mathbf{T}) \times \mathbf{C}^{\mathrm{T}} + \mathbf{F}^{*}\\ &= g(\mathbf{X}) \times \mathbf{C}^{\mathrm{T}} + \mathbf{F}^{*}\end{aligned} \tag{13}$$

where $g(\mathbf{X}) = f(\mathbf{T}) = \hat{\mathbf{U}}$ is the Y-score matrix generated from the nonlinear inner relation. From Equation (12), it is clear that $\Delta\mathbf{w}$ can only be derived when $\hat{\mathbf{u}}$ is a continuous and first order differentiable function of **w**. In this study, two continuous and first order differentiable nonlinear forms of the inner relation were applied; they are quadratic polynomial and piecewise (non-parametric) cubic spline functions. It should be noted that other recent PLSR modifications are also available for nonlinear meta-modelling. One such example is the cluster-based hierarchical PLSR [35], but they will not be considered here.

4.2. Model assessment and comparison

Four surrogate models were generated based on variants of the PLSR model discussed in the previous section. They are the linear model, Type I quadratic surface model, Type II quadratic polynomial model and Type II cubic smoothing spline model. Figure 5 demonstrates the modelling of the inner relationship using these four PLSR methods. It can be seen that with Type II approach,

DOI: 10.1002/cnm

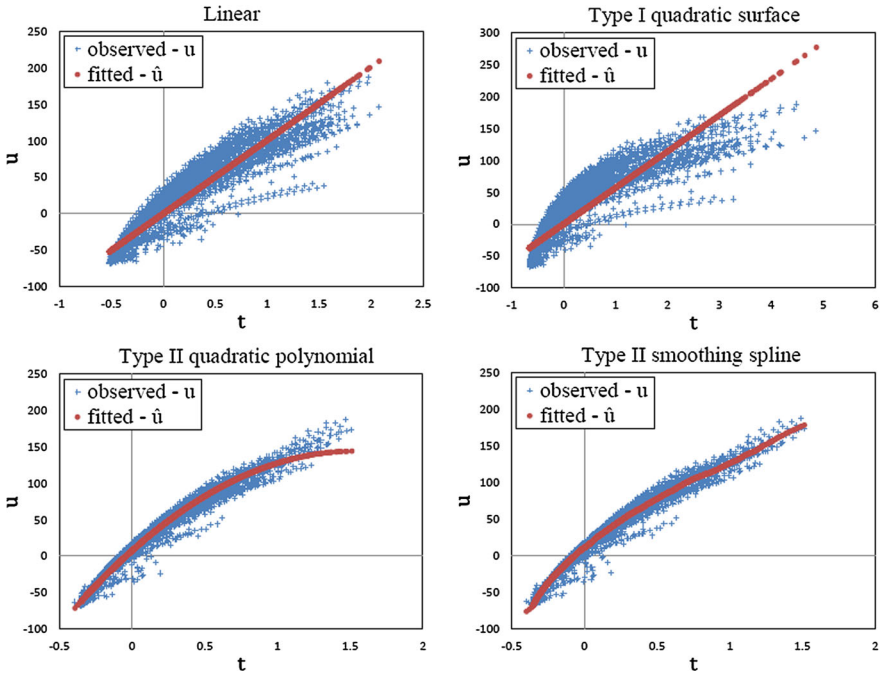

Figure 5. Scatter plots of scores and model of inner relationship for the first component of the partial least square regression surrogates.

the nonlinear inner associations are captured by the respective fitting functions. Also notice that while the Type I method models the inner relationship linearly, it extracts a different pair of **t** and **u** vectors in comparison to the linear PLSR as the result of the nonlinear transformation on the input matrix. This section compares the reproducibility (ability to reproduce training data) and predictability (ability to predict new data) of the model through the analysis of explained variance and cross-validation.

4.2.1. Reproducibility. A common technique to assess the goodness-of-fit of a PLSR model is to analyse the cumulative percentage of variance explained (CPVE). The CPVE for the input and output data blocks are defined as

$$\begin{aligned} \mathrm{CPVE_X} &= \frac{\sum_{n=1}^{N}\sum_{k=1}^{K}(X^*_{nk})^2}{\sum_{n=1}^{N}\sum_{k=1}^{K}(X_{nk})^2}\times 100\% \\ \mathrm{CPVE_Y} &= \frac{\sum_{n=1}^{N}\sum_{j=1}^{J}(Y^*_{nk})^2}{\sum_{n=1}^{N}\sum_{j=1}^{J}(Y_{nk})^2}\times 100\% \end{aligned} \tag{14}$$

where $\mathbf{X}^*$ and Y* are the reconstructed input and output data matrices using the PLS method. Figures 6 and 7 plot the CPVE as a function of the number of components (or equivalently the rank of the PLSR model). The variance values of the first two and the last components are also shown in Table II for quantitative comparison. It can be seen that the Type II cubic smoothing spline model performs slightly better than the Type II quadratic polynomial model, which, in turn, is better than the linear version. Nevertheless, the differences between these three PLSR models are small, due to the relatively linear response of the system (as can be seen by the fact that the linear model was able to reproduce 91% of the overall variability). When applying Type I quadratic surface model, a much higher percentage of variance can be captured (99%). However, it is at a cost of increased dimensionality on the input space and therefore an increase

DOI: 10.1002/cnm

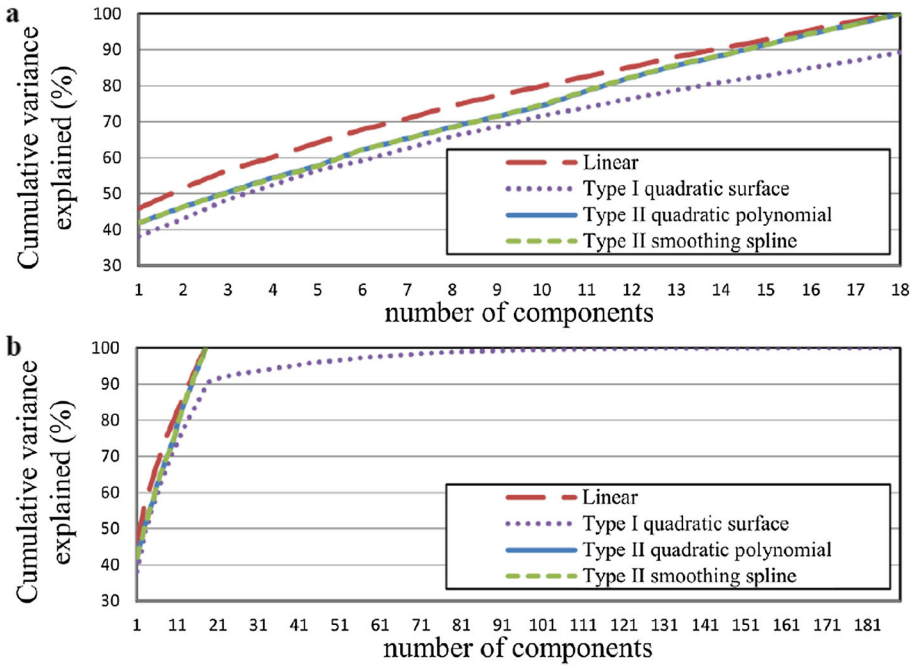

Figure 6. Cumulative percentage of variance explained in the independent variables (**X**) as a function of the number of components, showing (a) the first 18 components and (b) all 189 components required for the Type I quadratic surface partial least square regression.

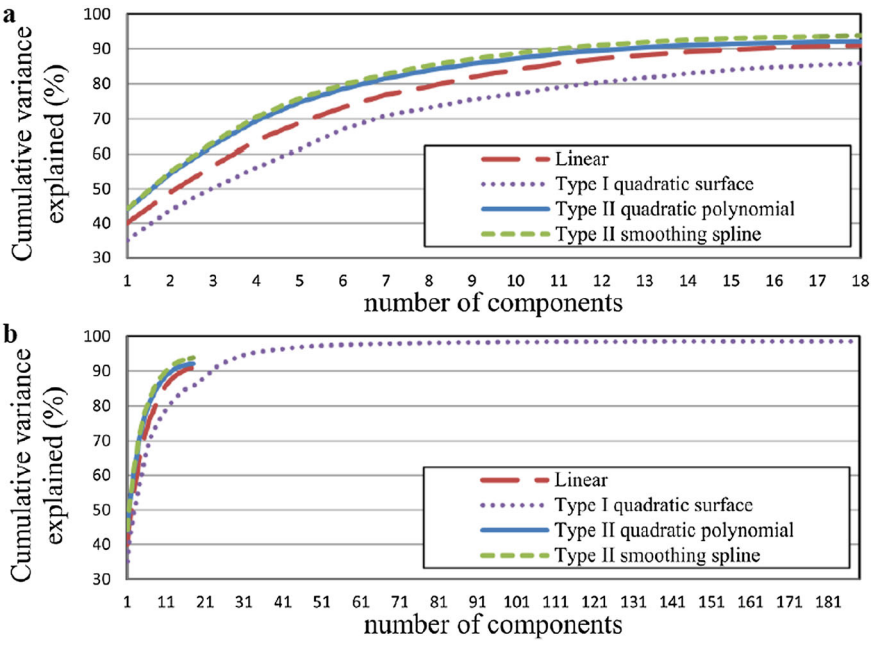

Figure 7. Cumulative percentage of variance explained in the dependent variables (**Y**) as a function of the number of components, showing (a) the first 18 components and (b) all 189 components required for the Type I quadratic surface partial least square regression.

in the rank of the model, leading to a greater computational cost. Moreover, as discussed previously, the Type I principle has the problem of interpretability and issues relating to the expansion of the dimensions.

DOI: 10.1002/cnm

Table II. Comparison of percentage of variance explained by different types of PLS model.

	X-block		T-block	
	% variance	Cumulative % variance	% variance	Cumulative % variance
Linear PLS model				
1	45.9064	45.9064	40.1120	40.1120
2	5.6734	51.5798	8.8971	49.0090
18	1.9944	100.0000	0.2863	90.9832
Type I nonlinear—quadratic surface PLS model				
1	38.3489	38.3489	35.0629	35.0629
2	4.7071	43.0560	8.7508	43.8137
189	0.0000	100.0000	0.0009	98.5530
Type II nonlinear—quadratic polynomial PLS model				
1	41.9094	41.9094	43.9849	43.9849
2	4.4019	46.3113	10.3980	54.3829
18	2.9449	100.0000	0.1177	92.1305
Type II nonlinear—cubic smoothing spline PLS model				
1	41.9453	41.9453	44.1818	44.1818
2	4.3252	46.2705	10.7255	54.9073
18	2.7967	100.0000	0.1555	93.6541

PLS, partial least square.

4.2.2. Cross-validation. Although analysing the CPVE of a particular surrogate is a useful practice, it has limited ability in establishing whether the surrogate is ready to be applied to data, which are not part of the training sample. Theoretically, for linear input–output systems, or for meta-model that is suitably extended to handle nonlinearities, a perfect prediction of the model's outputs $\mathbf{Y}$ from its inputs $\mathbf{X}$ should be possible, provided that the $\mathbf{X}$ and $\mathbf{Y}$ data are error free (apart from algorithmic problems such as inadequate convergence, round-off errors etc.). However, the problem is, if the statistical estimation process has estimated too many independent meta-model parameters, compared to the information content of the available training data, over-fitting (due to over-parameterisation) may arise. This means that small irrelevant variations in the input–output relationship (e.g. stochastic noise from measurements) are built into the model, whereby its predictive ability deteriorates. Therefore, it is important to assess the model's predictive ability in $\mathbf{Y}$ as a function of increasing PLS model complexity (i.e. number of estimated PLS components and number of nonlinearity parameters), and choose the model complexity with the best predictive performance. This can be performed empirically, using an independent test set (if one has a high number of training samples, N), or by cross-validation (if N is limited, as in this study).

In order to conduct cross-validation, the sampled data were randomly split into M subsets ($\mathbf{X}_1,\ldots,\mathbf{X}_M$ and $\mathbf{Y}_1,\ldots,\mathbf{Y}_M$) with approximately equal size (i.e. M-fold cross-validation). Following that, each variant of the PLS methods is then applied M times, leaving out one subset from training in each construction. From M reconstructed subset models, the sum of squared differences of the prediction is computed using the omitted subset, and the generalisation mean square error (GMSE) measurement is then obtained as the average of all the sum of squared differences values.

$$\mathrm{GMSE} = \frac{1}{M}\sum_{m=1}^{M}\left\|\mathbf{Y}_m - \mathbf{Y}_m^{*(-m)}\right\|^2 \tag{15}$$

where $\mathbf{Y}_m$ is the matrix of dependent variables of subset m and $\mathbf{Y}_m{}^{*(-m)}$ represents the prediction at $\mathbf{X}_m$ using the PLSR model trained from all sample data except subset m. In this study, $M=4$ subsets were used for cross-validation. Figure 8 plots the GMSE against the number of components

DOI: 10.1002/cnm

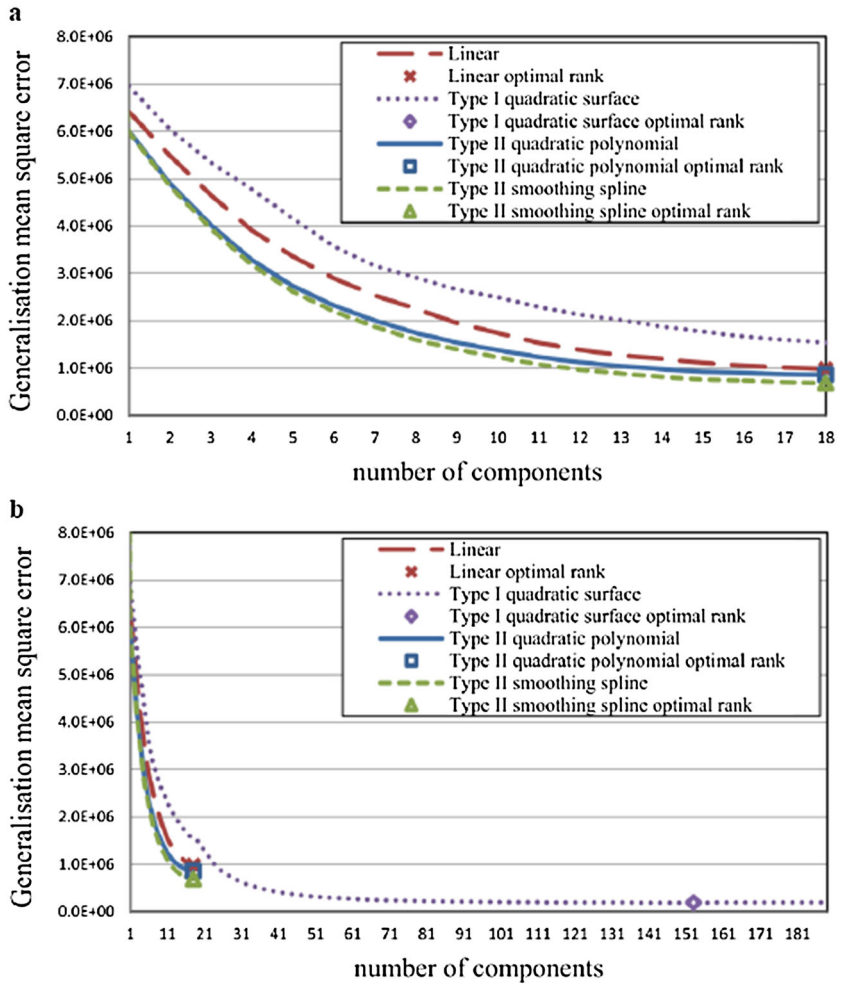

Figure 8. Cross-validation for the partial least square models, showing the generalisation mean squared errors (GMSE) using (a) first 18 components and (b) all 189 components required for the Type I quadratic surface partial least square regression. The optimal ranks (number of components) are depicted as having the lowest GMSE.

for all four variants of the PLSR model. In the figure, the optimal ranks (number of components) are depicted as having the lowest GMSE.

It can be seen that the Type I quadratic surface model performed better (with lower GSME) compared to Type II cubic smoothing spline model, which is closely followed by the Type II quadratic polynomial model and then the linear model. This conclusion is similar to the one drawn from the CPVE study. For the Type I quadratic surface model, the minimal GMSE was obtained at rank 153 instead of the largest possible rank (i.e. rank 189). This is indicative of over-parameterisation. However, the over-fitting in this case was insignificant, due to the large number of observations in comparison to the number of regression parameters. Moreover, it is likely that a different optimal rank can be obtained when using a different cross-validation scheme, because of the negligible increase of CPVE beyond rank 80 (less than 0.5% changes from rank 80 to rank 189). In the present case, a model rank of 153 (minimal GMSE) was employed. If future tests reveal that this is over-fitted and gives inadequate predictions in some parts of the parameter space, the nonlinear PLSR model may then be improved with better nonlinearity handling and a new cross-validation study.

DOI: 10.1002/cnm

5. PERFORMANCE AND SPEED

Because the main purpose of surrogate models is to reduce the computational cost, and therefore improve the feasibility for real-time and interactive applications, it is essential for the performance (in terms of speed and efficiency) of these surrogates to be evaluated and benchmarked. To do this, 100 random input parameter vectors (with uniformly distributed random values in each entry of the vector) were generated where each vector represents a single input point of the system (i.e. the activation parameters for a single gesture). Based on these input vectors, the dependant variables (i.e. mesh DOFs) were then predicted using the constructed surrogates.

Figure 9 illustrates the average computational time per parameter vector for each surrogate (note that the original biomechanical model required, on average, 2 h to complete a single simulation). As expected, it is seen that the linear PLSR model has the fastest performance, followed closely by Type II quadratic polynomial and Type II smoothing spline PLSR models. The Type I quadratic surface PLSR model demanded a significantly longer computational time due to the increased rank and the necessity to pre-transform the input space. It is also worth noting that approximately 250 machine hours were invested to produce the training sample required for this study. On top of that, an additional training time (from 5 s for the linear PLS model to 20 s for the Type I quadratic surface model) was required to build surrogate models from the sampled database. Nevertheless, this model building process is performed as a precursor and does not have an impact on real-time applications.

6. GENERATING FACIAL EXPRESSIONS USING SURROGATE MODELS

This section presents the forward modelling application where the surrogates are used in substitution of the original biomechanical model. This provides an insight into applying the surrogates to practical applications that require real-time performances. In order to demonstrate the reliability of PLSR surrogates, the four facial expressions shown in Figure 2 were regenerated using the regression models. Results are shown in Figure 10. It is seen that the displacement discrepancies between the Type I quadratic surface PLSR and full mechanics simulation are considerably lower than other nonlinear and linear regression models, in agreement with the conclusion drawn from cross-validating. In addition, all PLSR surrogates, except the Type I quadratic surface model, overestimate the tissue displacement for all four expressions, resulting in exaggerated predictions. This is possibly due to the fact that they were unable to mimic the nonlinear muscle fibre force-length relationship, where the active stress diminishes quadratically as the muscle shortens (see [12] for details). When using the Type I approach however, the higher dimensional mapping of the input space allows more of the components of the system to be captured, hence allowing this shortening effect to be predicted more accurately. Another observation is that in kissing gesture, the Type I quadratic surface PLSR is the only estimator that is capable of adequately describing the contact deformation between the lips. This has a consequential influence as the penetration of the lips immediately renders the results unrealistic.

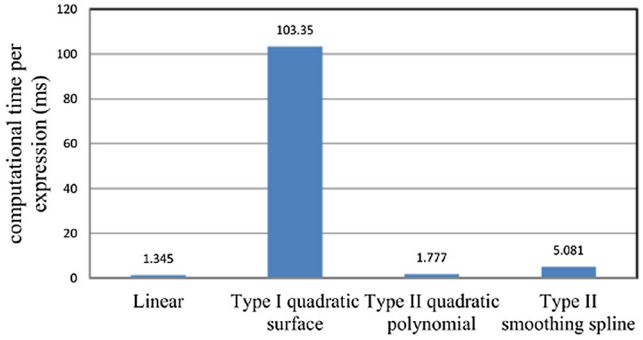

Figure 9. Average elapsed CPU time (in milliseconds) for generation of a single gesture.

DOI: 10.1002/cnm

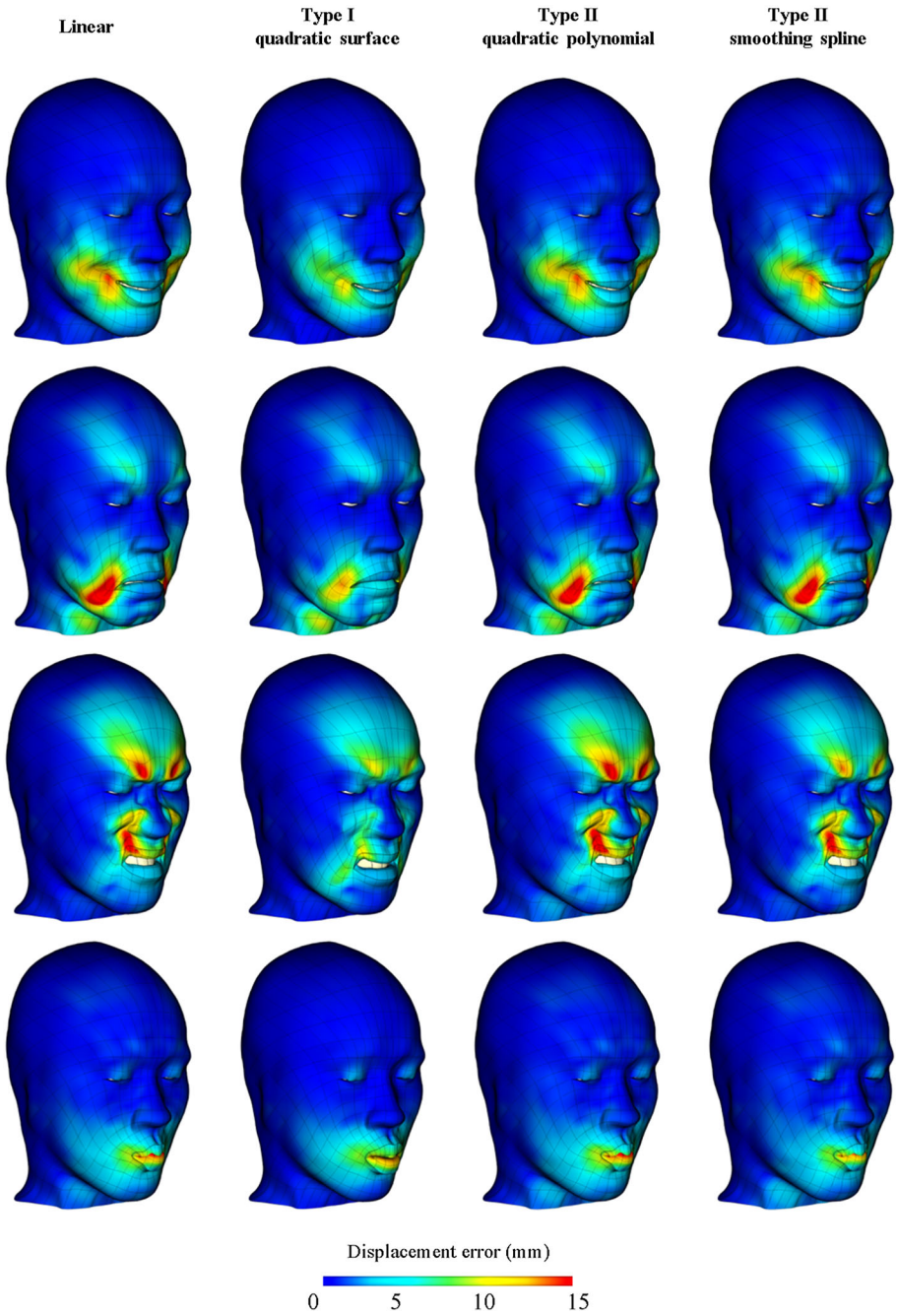

Figure 10. Reconstructed facial expressions using the partial least square regression surrogates, showing the Euclidean error between the surrogate-reconstructed deformations and biomechanically simulated deformations.

7. CONCLUSION

The primary focus of this paper is to demonstrate how statistically based surrogate models can be employed to accelerate the process of generating new facial expressions and gestures. The surrogates are built using the numerical data obtained from a detailed biomechanical computational model. This methodology is intended for real-time and interactive applications, where the speed of computation predominates over the accuracy of simulation. A significant portion of the surrogates' training data was generated using the MBR method, which produces a multileveled fraction factorial sampling scheme. This in combination with other simulation data resulted in 6108

DOI: 10.1002/cnm

observations of system that were used to train the PLSR surrogates. Four regression models, namely linear, Type I quadratic surface, Type II quadratic polynomial and Type II smoothing spline PLSR models were constructed. By means of cross-validation, it was shown that the Type I quadratic surface model outperforms all other regression models; however, it comes at a much higher computational cost, which may prohibit its use for applications that require real-time performance. Nevertheless, it shall be noted that better performance may be achieved through optimising the computational algorithms, and in addition, the suitability of a surrogate remains dependent on the specific requirement (i.e. accuracy versus speed) of a particular application. Finally, using the constructed surrogates, the facial biomechanics have been emulated with promising results. Moreover, by combining the facial surrogate models with an iterative fitting procedure, muscle activation values can be efficiently estimated from structured-light scanned geometric information (see [36] for details). The surrogate modelling framework proposed in this paper can also be applied to other complex biomechanical system where interactive speed is required but cannot be achieved through traditional physics-based techniques. It shall be noted that PLSR modelling work proposed here is merely preliminary; Bookstein [37] proposed a more rigorous method by requiring a match of energy terms between PLS predictor and FE predictands. However, the examples provided by Bookstein are for primitive geometries that have undergone simple deformations, and its extension to a realistic biomechanical problem remains to be explored.

ACKNOWLEDGEMENT

The work presented in this paper was funded by the Foundation for Research, Science and Technology of New Zealand under the grant number UOAX0712.

REFERENCES

1. Waters K, Terzopoulos D. A physical model of facial tissue and muscle articulation. In *Proceedings of the 1st Conference on Visualization in Biomedical Computing*. IEEE Computer Society Press: Los Alamitos, 1990; 77–82. doi:10.1109/VBC.1990.109305.
2. Sifakis E, Neverov I, Fedkiw R. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Transactions on Graphics* 2005; **24**(3):417–425. doi:10.1145/1073204.1073208.
3. Ramirez-Valdez L, Hasimoto-Beltran R. 3D-facial expression synthesis and its application to face recognition systems. *Journal of Applied Research and Technology* 2009; **7**(3):323–339.
4. Keeve E, Girod S, Pfeifle P, Girod B. Anatomy-based facial tissue modeling using the finite element method. In *Proceedings of the 7th Conference on Visualization*. IEEE Computer Society Press: Los Alamitos, 1996; 21–28. doi:10.1109/VISUAL.1996.567595.
5. Koch RM, Gross MH, Carls FR, von Büren DF, Fankhauser G, Parish YIH. Simulating facial surgery using finite element models. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. ACM: New York, 1996; 421–428. doi:10.1145/237170.237281.
6. Gladilin E, Zachow S, Deuflhard P, Hege HC. Anatomy- and physics-based facial animation for craniofacial surgery simulations. *Medical and Biological Engineering and Computing* 2004; **42**(2):167–170. doi:10.1007/BF02344627.
7. Chabanas M, Luboz V, Payan Y. Patient specific finite element model of the face soft tissues for computer-assisted maxillofacial surgery. *Medical Image Analysis* 2003; **7**(2):131–151. doi:10.1016/S1361-8415(02)00108-1.
8. Beldie L, Walker B, Lu Y, Richmond S, Middleton J. Finite element modelling of maxillofacial surgery and facial expressions - a preliminary study. *International Journal of Medical Robotics and Computer Assisted Surgery* 2010; **6**(4):422–430. doi:10.1002/rcs.352.
9. Nazari MA, Perrier P, Chabanas M, Payan Y. Shaping by stiffening: a modeling study for lips. *Motor Control* 2011; **15**(1):141–168.
10. Lucero JC, Munhall KG. A model of facial biomechanics for speech production. *Journal of the Acoustical Society of America* 1999; **106**(5):2834–42.
11. Wu T, Hung APL, Hunter P, Mithraratne K. On modelling large deformations of heterogeneous biological tissues using a mixed finite element formulation. *Computer Methods in Biomechanics and Biomedical Engineering*; published online ahead of print; 29 July 2013. DOI: 10.1080/10255842.2013.818662.
12. Wu T, Hung APL, Hunter P, Mithraratne K. Modelling facial expressions: a framework for simulating nonlinear soft tissue deformations using embedded 3D muscles. *Finite Elements in Analysis and Design* 2013; **76**:63–70. doi:10.1016/j.finel.2013.08.002.
13. Wu T, Hunter P, Mithraratne K. Simulating and validating facial expressions using an anatomically accurate biomechanical model derived from MRI data. *Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications*. SciTePress: Setubal, Portugal, 2013; pp. 267–272. DOI: 10.5220/0004293502670272.

14. Martens H, Mage I, Tondel K, Isaeva J, Hoy M, Saebo A. Multi-level binary replacement (MBR) design for computer experiments in high-dimensional nonlinear systems. *Journal of Chemometrics* 2010; **24**(11–12):748–756. doi:10.1002/cem.1366.
15. Box GEP, Draper NR. *Empirical Model-Building and Response Surfaces*. Wiley Series in Probability and Mathematical Statistics Applied Probability and Statistics. Wiley: New York, 1987.
16. Lancaster P, Salkauskas K. Surfaces generated by moving least squares methods. *Mathematics of Computation* 1981; **37**(155):141–158.
17. Broomhead DS, Loewe D. Multivariate functional interpolation and adaptive networks. *Complex Systems* 1988; **2**(3):321–355.
18. Jones DR. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization* 2001; **21**(4):345–383. doi:10.1023/A:1012771025575.
19. Suykens JAK, Vandewalle J. Least squares support vector machine classifiers. *Neural Processing Letters* 1999; **9**(3):293–300. doi:10.1023/A:1018628609742.
20. Queipoa NV, Haftkaa RT, Shyya W, Goela T, Vaidyanathana R, Tuckerb PK. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences* 2005; **41**(1):1–28. doi:10.1016/j.paerosci.2005.02.001.
21. Forrester AIJ, Keane AJ. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences* 2009; **45**(1–3):50–79. doi:10.1016/j.paerosci.2008.11.001.
22. Kleijnen JPC. *Design and Analysis of Simulation Experiments*. International Series in Operations Research & Management Science, Vol. **111**. Springer: New York, 2007.
23. Johnson ME, Moore LM, Ylvisaker D. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference* 1990; **26**(2):131–148. doi:10.1016/0378-3758(90)90122-B.
24. Tondel K, Gjuvsland AB, Mage I, Martens H. Screening design for computer experiments: metamodelling of a deterministic mathematical model of the mammalian circadian clock. *Journal of Chemometrics* 2010; **24**(11–12):738–747. doi:10.1002/cem.1363.
25. Bradley C, Bowery A, Britten R, Budelmann V, Camara O, Christie R, Cookson A, Frangi AF, Barbarenda Gamage T, Heidlauf T, Krittian S, Ladd D, Little C, Mithraratne K, Nash M, Nickerson D, Nielsen P, Nordbo O, Omholt S, Pashaei A, Paterson D, Rajagopal V, Reeve A, Rohrle O, Safaei S, Sebastian R, Steghofer M, Wu T, Yu T, Zhang H, Hunter P. A multi-physics & multi-scale computational infrastructure for the VPH/Physiome project. *Progress in Biophysics and Molecular Biology* 2011; **107**(1):32–47. doi:10.1016/j.pbiomolbio.2011.06.015.
26. Wold S, Martens H, Wold H. The multivariate calibration-problem in chemistry solved by the PLS method. *Lecture Notes in Mathematics* 1983; **973**:286–293. doi:10.1007/BFb0062108.
27. Rosipal R, Kramer N. Overview and recent advances in partial least sqaures. *Subspace, Latent Structure and Feature Selection Techniques*, Saunders C, Grobelnik M, Gunn S, Shawe-Taylor J (eds.). Springer-Verlag: Berlin, 2006; pp. 34–51. DOI: 10.1007/11752790_2.
28. Rosipal R, Trejo LJ. Kernel partial least squares regression in reproducing kernel hilbert space. *The Journal of Machine Learning Research* 2001; **2**:97–123.
29. Maitra S, Yan J. Principle component analysis and partial least squares: two dimension reduction techniques for regression. *Casualty Actuarial Society Spring Meeting*. Casualty Actuarial Society, 2008; pp. 79–90.
30. Wold H. Path models with latent variables: the NIPALS approach. *Quantitative Sociology: International Perspectives on Mathematical and Statistical Model Building*, Blalock HM (ed.). Academic Press: New York, 1975; pp. 307–357.
31. Wold S, Ruhe A, Wold H, Dunn WJ III. The collinearity problem in linear regression, the partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific Computing* 1984; **5**(3):735–743.
32. Rosipal R. Nonlinear partial least squares: an overview. *Chemoinformatics and advanced machine learning perspectives: complex computational methods and collaborative techniques*, Lodhi H, Yamanishi Y (eds.). IGI Global: New York, 2010; pp. 169–189.
33. Wold S, Kettaneh-Wold N, Skagerberg B. Nonlinear PLS modeling. *Chemometrics and Intelligent Laboratory Systems* 1989; **7**(1–2):53–65. doi:10.1016/0169-7439(89)80111-X.
34. Baffi G, Martin EB, Morris AJ. Non-linear projection to latent structures revisited: the quadratic PLS algorithm. *Computers and Chemical Engineering* 1999; **23**(3):395–411. doi:10.1016/S0098-1354(98)00283-X.
35. Tondel K, Indahl UG, Gjuvsland AB, Vik JO, Hunter P, Omholt SW, Martens H. Hierarchical cluster-based partial least squares regression (HC-PLSR) is an efficient tool for metamodelling of nonlinear dynamic models. *BMC Systems Biology* 2011; **5**:90. doi:10.1186/1752-0509-5-90.
36. Wu T, Martens H, Hunter P, Mithraratne K. Estimating muscle activation patterns using a surrogate model of facial biomechanics. In *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE Computer Society Press: Los Alamitos, CA, USA, 2013; 7172–7175. doi:10.1109/EMBC.2013.6611212.
37. Bookstein F. Allometry for the Twenty-First Century. *Biological Theory* 2013; **7**(1):10–25. doi:10.1007/s13752-012-0064-0.