Identifying and Quantifying Nonlinear Structured Relationships in Complex Manufactural Systems

Tingyang Xu*, Tan Yan†, Dongjin Song†, Wei Cheng†, Haifeng Chen†, Geoff Jiang‡ and Jinbo Bi§
*Tecent AI Lab, Shenzhen, China, Email: tingyangxu@tencent.com
†NEC Laboratories America, 4 Independence way, Princeton, New Jersey, 08540
Email: {yan,dsong,weicheng,haifeng}@nec-labs.com
‡Ant Financial, Hangzhou, China, Email: guofeijiang@yahoo.com
§University of Connecticut, Storrs, Connecticut, 06269, Email: jinbo.bi@uconn.edu

***Abstract*—Accurately identifying time-invariant operational relationships among different components is critical to autonomic management of complex manufactural systems. In this paper, we collect time series of sensor readings from manufacturing systems, and propose a solution leveraging Sparse Group LASSO to discover structured pairwise nonlinear relationships and quantify them by mathematical formulas. We consider both real-life operational patterns and underlying physical reactions inside the manufactural systems, which leads to a learning formulation for combined periodic and aperiodic system behaviors. An accelerated gradient descent algorithm is developed to efficiently solve the related optimization problem. We estimate sample correlations between proximal time points to improve the accuracy of the discovered relationships and the nonlinear quantitative formulas. The method is evaluated using both synthetic and real-world datasets, which shows superior performance over the state of the art in discovering nonlinear relationships in manufactural systems.**

1. Introduction

In the modern industry, decreasing hardware cost and increasing demand for autonomic system management make many complex manufactural systems, such as nuclear power plants, use a large network of monitoring sensors distributed across different parts of the system [1]. The continuous readings of sensors generate a huge amount of time series data every day, which contain rich information of the system, such as operational status and healthiness. Analyzing such data is challenging. On one hand, due to high system complexity, those time series contain heterogeneous dependencies among millions of parts across the system and are mixed with noise and operational patterns, which requires greater model complexity. On the other hand, autonomic management demands the identified models being highly interpretable as explicit linear or nonlinear formulas to guide system operators to diagnose issues and improve performance.

Recent work shows that identifying time-invariant relationships from sensor readings is effective to help system management tasks such as anomaly detection [2], [3], [4], and capacity planning [5]. The relationships between the sensors that measure physical parameters of the system reflect both operational patterns and physical reactions of a manufactural system. Such relationships usually hold persistently over time when the system is stable and healthy, and will be largely broken when system components fail. For example, [5] considers a linear invariant relationship between each two time series. Relationships between pairwise local components were expressed by explicit linear formulas, which can be assembled to profile a global status of the system. However, such a model suffers from low coverage as it can only discover linear interactions, and neglects many obviously nonlinear interactions between components, such as power and voltage (with fixed resistance), gas pressure and tube diameter, etc. Some state-of-the-art methods are able to learn nonlinear dependencies from data, such as sparse group additive models [6], [7], sparse nonlinear regressions [8], [9] and nonlinear additive ARX models [10]. However, their resultant models are not easy to be interpreted as nonlinear formulas in manufactural systems and thus cannot directly help autonomic management tasks.

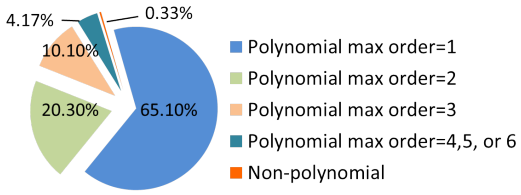

Figure 1: Types of relationships in manufactural systems.

Figure 1 shows different types of physical relationships in common manufactural systems after surveying sensor semantics of more than 10 different types of such systems and examining over 300 widely used macroscope physical formulas [11]. Interestingly, although nonlinear relationships pervasively exist in manufactural systems, 99.67% of them are in the polynomial family with the highest order no larger than six. [1] Therefore *focusing on the polynomial family will account for more than* 99% *of the relationships*, and we hence develop a novel approach to identifying the polynomials.

Our exploration is further inspired by observing the following three intrinsic characteristics from manufactural systems. (i) *Sparsity in polynomial bases:* The underlying physical laws determine that the relationship models sparsely rely on only a few polynomial terms, or the so-called bases. (ii) *Sparsity in delays:* Physical events take time to propagate.

1. The observation of polynomial relationships holds for manufactural systems. Other systems such as web systems do have many more non-polynomial relationships.

(iii) *Periodic and aperiodic:* The normal system operation usually contains repeated workload patterns, which adds periodic relationships on top of polynomials.

In this paper, we incorporate all of the above observations into the discovery of structured relationships and their quantitative formulas from complex manufactural systems. Similar to existing invariant models [5], we build regression models to regress one time series on another for each pair of the time series. Given the relationship is a combination of periodical and polynomial models, our strategy is to first deflate the periodical element from the relationship model using Discrete Fourier Transform (DFT) [12], [13], and then focus on constructing an effective explicit formula for the residual polynomial component, which is our major contribution of this paper. More particularly, to learn a polynomial relationship for each pair of time series, we kernelize the two time series by considering both their interactions and the autoregressive self interactions within each time series, and generate base polynomials for use at proximal time points. We learn the relationship by selecting relevant bases and delays and estimating their combination coefficients. The coefficients naturally form a matrix with each column representing a base and each row representing a lagged time point. Based on observations (i) and (ii), we organize the parameters into different groups according to rows as well columns, and then apply a Group LASSO with overlapping groups for joint selection of bases and delays.

In the parameter matrix, each row group overlaps with each column group by one element. We can decompose the matrix into two matrices, each of which containing either row or column groups and no overlapped groups. After decomposition we can separately regularize the row groups and column groups, which greatly reduces the problem complexity. We then design a sophisticated optimization algorithm following the framework of Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [14] to solve the problem, and prove the convergence of the algorithm. We further improve the model by estimating the correlation structure among proximal time points in time series using the Generalized Estimating Equation (GEE) [15], which usually improves the accuracy of the discovered relationships and their quantified formulas.

The proposed approach is evaluated using both synthetic datasets and real-world datasets from several manufactural systems. Experimental results demonstrate that it achieves high accuracy in discovering relationships and is feasible to be used for real-life system monitoring and diagnoses.

2. The Learning Formulation

Given two time series $\mathbf{x}$, and $\mathbf{y} \in \mathcal{R}^T$ where T represents the maximal number of time points considered from a physical system that contain polynomial dependencies and periodic patterns, the relationship between $\mathbf{x}$ and $\mathbf{y}$ can be formulated as:

$$\mathbf{y} = F_{periodic}\left(\mathbf{x}\right) + F_{polynomial}\left(\mathbf{x}\right) \tag{1}$$

We first employ a fast DFT algorithm [13] to extract periodic component $F_{periodic}(\cdot)$ from Eq.(1), which decomposes the signal into different sinusoidal signals of different frequencies and phases in the frequency domain. After decomposition, the aperiodic components have very high amplitudes in the lower frequency band whereas the periodic components are shown as peaks in the higher frequency band of the frequency domain. By subtracting the periodic components discovered by DFT from the relationship, we only need to model the polynomail part between $\mathbf{x}$ and $\mathbf{y}$: $F_{polynomial}(\cdot)$. In the following sections, we propose a Sparse Group LASSO with overlapping structures to estimate $F_{polynomial}(\cdot)$.

2.1. Problem Formulation

We first create polynomial bases for the observed time series, and then formulate a regularized optimization problem to construct a model as a function of these polynomial bases.

2.1.1. Kernelization of Signals. For two time series $\mathbf{x},\ \mathbf{y} \in \mathcal{R}^T$ where $\mathbf{x}$ is the independent signal and $\mathbf{y}$ is the response signal, we learn a regression model $\mathbf{y} = f(\mathbf{x})$. We first extend $\mathbf{x}$ to d different power bases as $\mathbf{k}_{(x;i)} = [x_i, x_i^2, \cdots, x_i^d]$ where i indexes the observation time point. We then model the interactions between $\mathbf{x}$ and the response signal $\mathbf{y}$ by first setting a mapping $\mathbf{k}_{(x,y;i)} = [y_i, \cdots, y_i^d, y_i x_i, y_i x_i^2, \cdots, y_i^d x_i^{d-1}, y_i^d x_i^d]$, and then forming an autoregressive model as a function of the following matrix where each power basis (column) consists of the current and τ previous records of the repeated measurements

$$\mathbf{K}_{(x,y;t)} = \begin{bmatrix} \mathbf{k}_{(x;t)}^\top & \mathbf{k}_{(x;t-1)}^\top & \cdots & \mathbf{k}_{(x;t-\tau)}^\top \\ \mathbf{0} & \mathbf{k}_{(x,y;t-1)}^\top & \cdots & \mathbf{k}_{(x,y;t-\tau)}^\top \end{bmatrix}^\top.$$

This is a $(\tau+1) \times d(d+2)$ kernel matrix, and $\mathbf{k}_{(x,y;t)}$ inside $\mathbf{K}_{(x,y;t)}$ is set to $\mathbf{0}$ because y_t is the current target to be predicted. Given T total repeated measurements for each signal, the index t of $\mathbf{K}_{(x,y;t)}$ starts from $\tau+1$ in order to have enough observations in the first training example. If $\mathbf{K}_{(x,y;t)}$ is considered as feature matrix, then the model $y_t = tr\left(\mathbf{K}_{(x,y;t)}^\top \mathbf{W}\right)$ where $\mathbf{W} = [\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_\tau]^\top$ gives a linear model with τ delays.

2.1.2. The Objective Function. We now formulate an optimization problem based on the two observations (i) and (ii), which show that the sensor relationships should sparsely rely on certain bases and delays. To select among all the features in the kernel matrix, we separately select among basis polynomials (columns) and select among the lagged effects from proximal time points (rows). In other words, each row of $\mathbf{K}_{(x,y;t)}$ is a structured group (the different bases at the same time point) and each column is a structured group (the same basis at all the τ lagged time points). We design a latent sparse regularizer as shown in Figure 2 which produces sparsity within the row groups as well as column groups. Figure 2 shows a $\mathbf{W}$ where five elements are selected from one basis column and two delay rows.

Therefore, besides the least square loss for the model, we form the optimization problem as Sparse Group LASSO with overlapping structures as:

$$\min_{\mathbf{P},\mathbf{Q}}\ \ell\left(\mathbf{W}\right) + (1-\alpha)\lambda_1 \sum_{g=1}^{G} \Omega_g(\mathbf{W}) + \alpha\lambda_2 \|\mathbf{W}\|_1 \tag{2}$$

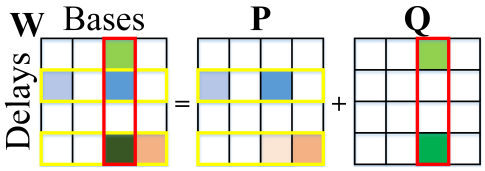

Figure 2: Parameter Matrix with Group Formation

where $\ell(\mathbf{W}) = \sum_{t=1}^{T} \|y_t - tr(\mathbf{K}_{(x,y;t)}^{\top}\mathbf{W})\|_2^2$, α, λ_1, and λ_2 are tuning parameters, and $\Omega_g(\cdot)$ refers to the structured-sparsity-inducing penalty. The regularizer $\Omega_g(\cdot)$ plays a major role for encouraging the closely related inputs (e.g., consecutive time points) to be selected together as relevant to the output by setting the corresponding regression coefficients to non-zero values. Commonly, this regularizer uses the $\ell_{1,2}$ matrix norm applied to $\mathbf{W}$ (as $\|\mathbf{W}\|_{1,2} = \sum_i \|\mathbf{W}_{i,\cdot}\|_2$) as well as to $\mathbf{W}^{\top}$ where $\mathbf{W}_{i,\cdot}$ represents the i-th row of $\mathbf{W}$.

The optimization of Eq.(2) is challenging because $\mathbf{W}$ has overlapping elements in all the base and delay groups. Traditional block-wised gradient descent methods generally do not allow the overlapping among the groups [16]. The general proximal gradient method has to solve an optimization sub-problem at each step of updates, which is costly to get the optimal solution [17]. Moreover, this generalized sparse group LASSO (SGLASSO) usually requires an index matrix to indicate the group membership, which quadratically increases the usage of memory space and can be a challenge to large-scale problems.

2.1.3. The Revised Regularizer. In our problem, the parameters naturally form a matrix, where row groups only overlap with column groups or vice versa but never within rows or columns themselves. We can hence decompose $\mathbf{W}$ into a summation of two component matrices as $\mathbf{W} = \mathbf{P} + \mathbf{Q}$, and select only row groups in $\mathbf{P}$ and column groups in $\mathbf{Q}$ as shown in Figure 2. Then, both rows and columns are selected in $\mathbf{W}$ after summation, as shown in Figure 2. This decomposition slightly changes our objective function, but it degenerates the regularization condition of the original problem to two easier sparse group LASSO penalties that do not have overlapping groups. In this regularizer, we apply the $\ell_{1,2}$ norm to $\mathbf{P}$ row-wisely, so the optimal $\mathbf{P}$ will contain rows with all zero entries and hopefully only a few rows with non-zero entries. Similarly, the $\ell_{1,2}$ norm is applied to $\mathbf{Q}^{\top}$ (i.e., to $\mathbf{Q}$ column-wisely) to encourage the selection among columns of $\mathbf{Q}$.

Overall, we revise the regularization condition in Eq.(2) and solve the following optimization problem for the optimal model parameter matrix $\mathbf{W}$, which is computed as $\mathbf{P} + \mathbf{Q}$:

$$\begin{aligned} \min_{\mathbf{P},\mathbf{Q}} \; & \ell(\mathbf{W}) + \alpha\left(\lambda_1\|\mathbf{P}\|_1 + \lambda_2\|\mathbf{Q}\|_1\right) \\ & +(1-\alpha)\left(\lambda_1\|\mathbf{P}\|_{1,2} + \lambda_2\|\mathbf{Q}^{\top}\|_{1,2}\right) \\ s.t. \quad & \mathbf{W} = \mathbf{P} + \mathbf{Q} \end{aligned} \tag{3}$$

where $\mathbf{W}$ in the least squares loss will simply be replaced by $\mathbf{P} + \mathbf{Q}$, and α, λ_1, and λ_2 are tuning parameters in the model. α is playing a role of balancing the weight between $\ell_{1,2}$-norm and ℓ_1-norm.

Although the regularization part of Eq.(2) is decomposed to two sparse group LASSO penalties, the least squares loss is intact. Moreover, $\mathbf{P}$ and $\mathbf{Q}$ are estimated jointly, and they both contribute to the optimal solution of $\mathbf{W}$. Therefore, we cannot just apply existing sparse group LASSO solvers to solve Eq.(3). In the following section, we develop an accelerated gradient descent method based on the FISTA [14], which efficiently solves the optimization problem Eq.(3).

2.2. Optimization Algorithm

The FISTA algorithm [14] can be viewed as an extension of the classical gradient descent algorithm with global convergence and its convergence rate has been proven to be super-linear. It uses the proximal operator to deal with non-smooth regularizers in optimization problems.

2.2.1. Reformulation of FISTA. To solve the optimization problem Eq.(3), we follow the FISTA framework that provides an accelerated gradient framework to minimize the proximal approach of the objective function. Problem (3) contains a smooth loss function and non-smooth regularizers, which meets the formulation requirement of the FISTA.

We denote the objective function of Eq.(3) by $f(\mathbf{P},\mathbf{Q})$, and use $\ell(\mathbf{P},\mathbf{Q})$ to denote its continuously differentiable part that is the least squares loss, and $R(\mathbf{P},\mathbf{Q})$ to denote its nonsmooth part that constitutes the regularizers. We hence have $f(\mathbf{P},\mathbf{Q}) = \ell(\mathbf{P},\mathbf{Q}) + R(\mathbf{P},\mathbf{Q})$.

We develop the following iterative procedure to find the optimal values of $\mathbf{P}$ and $\mathbf{Q}$. Let $\nabla_{\mathbf{P}}\ell(\mathbf{P},\mathbf{Q})$, $\nabla_{\mathbf{Q}}\ell(\mathbf{P},\mathbf{Q})$ be the partial derivative of $\ell(\mathbf{P},\mathbf{Q})$ with respect to $\mathbf{P}$ and $\mathbf{Q}$, respectively. For any given point $(\tilde{\mathbf{P}},\tilde{\mathbf{Q}})$, the following $Q_{L,\tilde{\mathbf{P}},\tilde{\mathbf{Q}}}(\mathbf{P},\mathbf{Q})$ is a *well-defined* proximal map for the non-smooth R as

$$\begin{aligned} Q_{L,\tilde{\mathbf{P}},\tilde{\mathbf{Q}}}(\mathbf{P},\mathbf{Q}) &= \ell(\tilde{\mathbf{P}},\tilde{\mathbf{Q}}) + R(\mathbf{P},\mathbf{Q}) + \langle\nabla_{\mathbf{P}}\ell(\tilde{\mathbf{P}},\tilde{\mathbf{Q}}), \mathbf{P}-\tilde{\mathbf{P}}\rangle \\ &+ \frac{L}{2}\|\mathbf{P}-\tilde{\mathbf{P}}\|_F^2 + \langle\nabla_{\mathbf{Q}}\ell(\tilde{\mathbf{P}},\tilde{\mathbf{Q}}), \mathbf{Q}-\tilde{\mathbf{Q}}\rangle + \frac{L}{2}\|\mathbf{Q}-\tilde{\mathbf{Q}}\|_F^2 \end{aligned}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product. If $\ell(\mathbf{P},\mathbf{Q})$ has a Lipschitz continuous gradient with a Lipschitz modulus L, then according to Lemma 2.1 in [14], the inequality $f(\mathbf{P},\mathbf{Q}) \leq Q_{L,\tilde{\mathbf{P}},\tilde{\mathbf{Q}}}(\mathbf{P},\mathbf{Q})$. holds, which indicates that $Q_{L,\tilde{\mathbf{P}},\tilde{\mathbf{Q}}}(\mathbf{P},\mathbf{Q})$ is an upper bound on the objective function $f(\mathbf{P},\mathbf{Q})$.

2.2.2. Updates at Each Step. To minimize $Q_{L,\tilde{\mathbf{P}},\tilde{\mathbf{Q}}}(\mathbf{P},\mathbf{Q})$, we now describe the iterative procedure to update the iterates at each step in our algorithm. Starting from an initial point $(\mathbf{P}_0,\mathbf{Q}_0)$, we iteratively search for the optimal solution. Denote the iterates at the $\mathcal{K}$-th iteration by $\mathbf{P}_{\mathcal{K}}$ and $\mathbf{Q}_{\mathcal{K}}$. At each iteration $\mathcal{K}$, we first use the iterates $(\mathbf{P}_{\mathcal{K}-1},\mathbf{Q}_{\mathcal{K}-1})$ and $(\mathbf{P}_{\mathcal{K}-2},\mathbf{Q}_{\mathcal{K}-2})$ to compute (at the first iteration, $(\tilde{\mathbf{P}}_1,\tilde{\mathbf{Q}}_1) = (\mathbf{P}_0,\mathbf{Q}_0)$)

$$\begin{aligned} \tilde{\mathbf{P}}_{\mathcal{K}} &= \mathbf{P}_{\mathcal{K}-1} + \left(\frac{t_{\mathcal{K}-1}-1}{t_{\mathcal{K}}}\right)(\mathbf{P}_{\mathcal{K}-1} - \mathbf{P}_{\mathcal{K}-2}), \\ \tilde{\mathbf{Q}}_{\mathcal{K}} &= \mathbf{Q}_{\mathcal{K}-1} + \left(\frac{t_{\mathcal{K}-1}-1}{t_{\mathcal{K}}}\right)(\mathbf{Q}_{\mathcal{K}-1} - \mathbf{Q}_{\mathcal{K}-2}), \end{aligned} \tag{4}$$

where $t_{\mathcal{K}}$ is a scalar and updated at each iteration as:

$$t_{\mathcal{K}+1} = \frac{1+\sqrt{1+4t_{\mathcal{K}}^2}}{2}. \tag{5}$$

Since there is no interacting term between $\mathbf{P}$ and $\mathbf{Q}$ in $Q_{L,\tilde{\mathbf{P}},\tilde{\mathbf{Q}}}(\mathbf{P},\mathbf{Q})$, the problem can be decomposed into two separate subproblems in terms of $\mathbf{P}$ and $\mathbf{Q}$, respectively. Then, an optimal solution $(\mathbf{P}_{\mathcal{K}},\mathbf{Q}_{\mathcal{K}})$ near the point $(\tilde{\mathbf{P}}_{\mathcal{K}},\tilde{\mathbf{Q}}_{\mathcal{K}})$ can be solved by the following two proximal operators:

$$\min_{\mathbf{P}} \frac{1}{2}\left\|\mathbf{P}-\left(\tilde{\mathbf{P}}_{\mathcal{K}}-\frac{1}{L}\nabla_{\mathbf{P}}\ell_{\mathcal{K}}\right)\right\|_F^2 + \frac{\lambda_1}{L}\left(\alpha\|\mathbf{P}\|_1+(1-\alpha)\|\mathbf{P}\|_{1,2}\right), \tag{6}$$

$$\min_{\mathbf{Q}} \frac{1}{2}\left\|\mathbf{Q}-\left(\tilde{\mathbf{Q}}_{\mathcal{K}}-\frac{1}{L}\nabla_{\mathbf{Q}}\ell_{\mathcal{K}}\right)\right\|_F^2 + \frac{\lambda_1}{L}\left(\alpha\|\mathbf{Q}\|_1+(1-\alpha)\|\mathbf{Q}\|_{1,2}\right),$$

where $\nabla_{\mathbf{P}}\ell_{\mathcal{K}}$ and $\nabla_{\mathbf{Q}}\ell_{\mathcal{K}}$ are respectively the partial derivatives of ℓ computed at $(\tilde{\mathbf{P}}_{\mathcal{K}},\tilde{\mathbf{Q}}_{\mathcal{K}})$, and L acts as a learning step size. The two subproblems share the same structure and thus can be solved following the same procedure. Hence, we only show how to solve Eq.(6) for the best $\mathbf{P}$.

Eq.(6) has a closed-form solution [18] where each row of $\mathbf{P}_{\mathcal{K}}$, $\mathbf{P}^{\mathcal{K}}_{(i,)}$ is $\mathbf{P}^{\mathcal{K}}_{(i,)} = \max\left(0, 1-\frac{(1-\alpha)\lambda_1}{L\|\mathbf{S}^{(\mathcal{K})}_{(i,)}\|_2}\right)\mathbf{S}^{(\mathcal{K})}_{(i,)}$, with $\mathbf{S}^{(\mathcal{K})} = \max\left(0, \tilde{\mathbf{P}}^{(\mathcal{K})}-\alpha\lambda_1\right)$ and $\tilde{\mathbf{P}}^{(\mathcal{K})} = \tilde{\mathbf{P}}_{\mathcal{K}} - \frac{1}{L}\nabla_{\mathbf{P}}\ell_{\mathcal{K}}$. The gradient vector $\nabla_{\mathbf{P}}\ell_{\mathcal{K}}$ can be computed as

$$\nabla_{\mathbf{P}}\ell_{\mathcal{K}} = \text{reshape}\left(\mathbf{D}_{(x)}\left(\mathbf{y}-\mathbf{D}^{\top}_{(x)}\text{vect}\left(\tilde{\mathbf{P}}_{\mathcal{K}}+\tilde{\mathbf{Q}}_{\mathcal{K}}\right)\right)\right) \tag{7}$$

where $\mathbf{D}_{(x)} = \left[\text{vect}\left(\mathbf{K}_{(x;1)}\right),\cdots,\text{vect}\left(\mathbf{K}_{(x;T)}\right)\right]$ and reshape$(\cdot)$ refers to an operator of reshaping a vector into a matrix of the size related to the context. Following this way, we can update $\mathbf{P}$ and $\mathbf{Q}$ at each step.

2.2.3. Estimation of Lipschitz Constant. In the above discussion, the Lipschitz modulus L needs to be computed at each step. However, the calculation of L can be computational expensive. We therefore follow the similar argument in [19] to find a proper approximation $\tilde{L}$. Algorithm 1 summarizes the steps for finding optimal $\mathbf{P}$ and $\mathbf{Q}$.

Algorithm 1 Search for optimal $\mathbf{P}$ and $\mathbf{Q}$

Input: $\mathbf{X}$, $\mathbf{y}$, λ_1, λ_2, α
Output: $\mathbf{P}$, $\mathbf{Q}$
1. $\mathcal{K}$ = 1, compute $\tilde{L}$ and initialize $t_1 = 1$, $\mathbf{P}_0 = \tilde{\mathbf{P}}_1 = \mathbf{0}$ and $\mathbf{Q}_0 = \tilde{\mathbf{Q}}_1 = \mathbf{0}$;
2. Solve Eq.(6) to obtain $\mathbf{P}_{\mathcal{K}}$ and $\mathbf{Q}_{\mathcal{K}}$.
3. Compute $t_{\mathcal{K}+1}$ by Eq.(5).
4. Compute $\tilde{\mathbf{P}}_{\mathcal{K}+1}$ and $\tilde{\mathbf{Q}}_{\mathcal{K}+1}$ by Eq.(4).
5. $\mathcal{K} = \mathcal{K}+1$.
Repeat $2\sim5$ until convergence.

2.3. Convergence Analysis

We show that Algorithm 1 converges to the optimal solution of Eq.(3) with a convergence rate of $O(1/\mathcal{K}^2)$.

Theorem 1. *Let $\mathbf{P}_{\mathcal{K}}$ and $\mathbf{Q}_{\mathcal{K}}$ be the pair of the matrix generated by Algorithm 1. Then for any $\mathcal{K}\geq 1$*

$$f(\mathbf{P}_{\mathcal{K}},\mathbf{Q}_{\mathcal{K}})-f(\hat{\mathbf{P}},\hat{\mathbf{Q}}) \leq \frac{2\tilde{L}\left(||\mathbf{P}_0-\hat{\mathbf{P}}||_F^2+||\mathbf{Q}_0-\hat{\mathbf{Q}}||_F^2\right)}{(\mathcal{K}+1)^2}$$

where $(\hat{\mathbf{P}},\hat{\mathbf{Q}})$ is a globally optimal solution of Eq.(3).

The theorem can be proved following the similar steps discussed in our early work [20].

2.4. Estimation of Correlation Structure

The model in Eq.(3) assumes that the training examples are *i.i.d.*. However, obviously, the consecutive time points are not mutually independent and the kernelized signals contain overlapping records over time. In this case, the Generalized Estimating Equation (GEE) is an ideal solution as it provides a mechanism to estimate sample correlation simultaneously while constructing regression models. However, if we were to adopt the GEE to estimate the covariance matrix which is of a very large size (e.g., 30,000 time points needs a $30,000,000$ matrix), it would create potential momery issues. Since the physical sensors are assumed to have relationships in τ neighborhood time points, we can estimate a $\tau\times\tau$ covariance matrix to each τ-sized chunk of kernel as $\mathbf{D}_{(x;p)} = \left[\text{vect}\left(\mathbf{K}_{(x,p\times\tau)}\right),\cdots,\text{vect}\left(\mathbf{K}_{(x,\min(T,(p+1)\times\tau))}\right)\right]$ and the response signal as $\mathbf{y}_p = \mathbf{y}_{[p\times\tau,\cdots,\min(T,(p+1)\times\tau)]}$, where p is the index of the chunk, instead of estimating on the whole time points. Hence, the gradient update used in Eq.(7) is replaced by the following formula:

$$\nabla_{\mathbf{P}}\ell_{\mathcal{K}} = \text{reshape}\left(\text{vcat}_{p=0}^{\lfloor T/\tau\rfloor}\left(\mathbf{D}_{(x;p)}\left(\mathbf{R}_p(\boldsymbol{\rho})\right)^{-1}\mathbf{s}_p\right)\right)$$

where $\mathbf{s}_p = \mathbf{y}_p - \mathbf{D}^{\top}_{(x;p)}\text{vect}\left(\tilde{\mathbf{P}}_{\mathcal{K}}+\tilde{\mathbf{Q}}_{\mathcal{K}}\right)$, $\text{vcat}_{p=0}^{\lfloor T/\tau\rfloor}(\cdot)$ denotes an operator that vertically concatenates matrices, and $\mathbf{R}(\boldsymbol{\rho})$ refers to a covariance matrix that can be estimated from the current Pearson residuals. We develop an expectation-maximization (EM) algorithm to alternatively estimate $\mathbf{R}(\boldsymbol{\rho})$ and optimize Eq.(3).

3. Experimental Results

We first generated synthetic data to evaluate the fitness accuracy of the learned regression model. We then evaluated the quantified models constructed by our approach by comparing the selected bases and delays with the ground-truth formulas. The *periodic* components of the formulas were firstly estimated by our approach and subtracted from the synthesized signals. We compared our approach against the following methods in terms of the learning performance on the *polynomial* components of formulas: 1) **Non-linear Regression**: the close-form solution by a linear (the first-order Taylor expansion) approximation to the non-linear components. This is used as a baseline. 2) **LASSO**: we used a similar design of the kernel bases in the loss function but only used the ℓ_1-norm regularizer. 3) **Longi-LASSO** [20]: one of our earlier work that estimated the most influential time points and features. We treated the different power terms as different features.

We also tested our method on the real-world datasets collected from two manufactural systems in terms of the model stability and its application to anomaly detection.

3.1. Synthetic Data

We generated a synthetic dataset consisting of 10,000 pairs where each signal had 40,000 time points. We first produced a seed signal $\mathbf{x}$ as follows, which was then used to create other signals that had a relationship with $\mathbf{x}$:

$$x_t = \epsilon_1 \times \sin(0.001t) \quad \forall\ t = [1, \ldots, 40000] \tag{8}$$

where ϵ_1 followed the standard Gaussian distribution $\mathrm{N}(0, 1)$. Then, we simulated the noisy sensor readings in real manufactural systems to generate the related signals $\mathbf{y}$ from the seed signal. Each $\mathbf{y}$ was computed from an explicit formula that combined the selected elements in Figure 3, where $f_{\sin}(x) = \sin(10\pi x + 10)$ and $f_{\cos} = \cos(16\pi x - 3)$.

$$\begin{bmatrix} f_{\sin}(x_t) & f_{\sin}(x_{t-1}) & f_{\sin}(x_{t-2}) & f_{\sin}(x_{t-3}) \\ f_{\cos}(x_t) & f_{\cos}(x_{t-1}) & f_{\cos}(x_{t-2}) & f_{\cos}(x_{t-3}) \\ x_t & x_{t-1} & x_{t-2} & x_{t-3} \\ \cdots & \cdots & \cdots & \cdots \\ x_t^4 & x_{t-1}^4 & x_{t-2}^4 & x_{t-3}^4 \\ 0 & y_{t-1} & y_{t-2} & y_{t-3} \\ 0 & y_{t-1}x_{t-1} & y_{t-2}x_{t-2} & y_{t-3}x_{t-3} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & y_{t-1}x_{t-1}^4 & y_{t-2}x_{t-2}^4 & y_{t-3}x_{t-3}^4 \end{bmatrix}$$

Figure 3: An example of the selected rows and columns that were used in a formula to create the related signal $\mathbf{y}$.

In order to simulate relationships from physical reactions in the synthetic data, for every pair we randomly chose 2 rows and 1 column, and selected 5 functions from the chosen rows and column to construct a formula. We denoted these 5 selected bases by $f_1, \ldots, f_5$. The signal $\mathbf{y}$ was computed as $y_t = \sum_{i=1}^{5} \beta_i f_i + \epsilon_2$, where β_i followed an Uniform distribution as $\mathcal{U}(0, 1)$ and ϵ_2 followed $\mathrm{N}(0, 0.1)$.

Following the aforementioned process, we generated 9,000 time series that had specific nonlinear relationships with $\mathbf{x}$. We then also created 1,000 random time series which were used as non-related signals to $\mathbf{x}$.

3.1.1. Accuracy of Fitness. The coefficient of determination R^2 was used to measure the fitness accuracy. The R^2 statistic ranges from 0 to 1 with higher values indicating better performance. We used the first 2/3 time points of each signal as training data and the remaining signals for test. All methods in comparison were examined using the same paired data in the synthetic dataset. The number of bases and delays were set as $d = 4, \tau = 3$. The hyper-parameters, λ_1 and λ_2, were equally chosen from 0.001 to 0.0002 with a step size of 0.0002 and α was set to 0.6. Particularly, the nonlinear regression method had no parameter to tune.

Table 1 shows the average test R^2 of the different methods for the related and unrelated pairs. It indicates that the unrelated pairs usually had a R^2 closed to 0, which means that no relationships were found in those pairs. For related pairs, our method achieved the highest fitness over the others. We believe that the group-structured selection of both bases and delays did help improve regression performance.

TABLE 1: Accuracy of Fitness and Formula Expression (Standard deviations are shown in $(\cdot)$.)

Methods	Related	Unrelated	Selected	Correct
Nonlinear	0.38	0.0011	-	-
LASSO	0.82	0.0012	6.96(0.73)	3.87(0.26)
Longi-LASSO	0.84	0.0009	10.1(0.23)	4.76(0.01)
Our Method	0.89	0.0010	7.02(0.12)	4.76(0.01)

3.1.2. Accuracy of Formula Expression. In addition to fitness accuracy, we validated the quantitative formula constructed by the methods by evaluating the total number of selected bases and the number of correct selections in the related pairs. Table 1 shows the averages and the standard deviations of the numbers of selections when different λ values were used in the LASSO-based methods. The nonlinear regression models had no function of feature selection. From this table we can see both longitudinal LASSO and the proposed method selected similar number of correct elements. However, in order to achieve that, longitudinal LASSO selected about 40% more elements, making its model rely on more unrelated elements than our method. LASSO and our method selected similar numbers of elements, but our method selected more correct elements. Moreover, our method was less sensitive to the choices of parameters as the standard deviations were very small when λs varied substantially. These results domonstrate that the proposed method can recover the synthesized formulas with a better accuracy than the compared methods.

3.2. Case Study with Real-World Data

In this section, we tested the proposed method using two real-world time series datasets collected from a large amount of sensors in two manufactural systems of power plant. Due to the proprietary nature of the data, we use Power Plant A to name the first system. The system contained over 5000 sensors, distributed to monitor pressures, temperatures, voltages of the system for several months, and it included three system faults and one system reset.

We used the first 2/3 of the time points in the paired signals in training and the remaining data for test. In training, we set $d = 4, \tau = 3, \lambda_1 = \lambda_2 = 0.001$, and $\alpha = 0.6$ for the proposed method. A pair of signals was identified to be related if its training $R^2 > 0.7$ and test $R^2 > 0.5$.

Another dataset from Power Plant B contained 6000 sensors/time series, distributed to monitor pressures, temperatures, voltages of the system for 4 months. Each time series consisted of $175,680$ time points and each point corresponded to 1 minute of data reading. Data of the first 2 months contained normal operations and the last 2 months contained 3 system faults and 1 system reset. We applied our algorithm only to the first month's data to train and identify pairwise relationships, which represented the normal system status. We then examined the identified pairs on the other

3 months' data to detect possible anomalies. The anomaly detection algorithm was designed following a widely used method in SIAT [5]. Namely, the residuals of the related pairs at each time point in the test data would be evaluated. At a time point, a related pair was considered *broken* if the residual of that pair exceeded a threshold of $1.1 \times r_0$ where r_0 referred to 99.5% confidence rate of the residuals from the training data. In the test phase, we examined all the pairs identified in the training phase, and counted the total number of broken pairs at each time point along time.

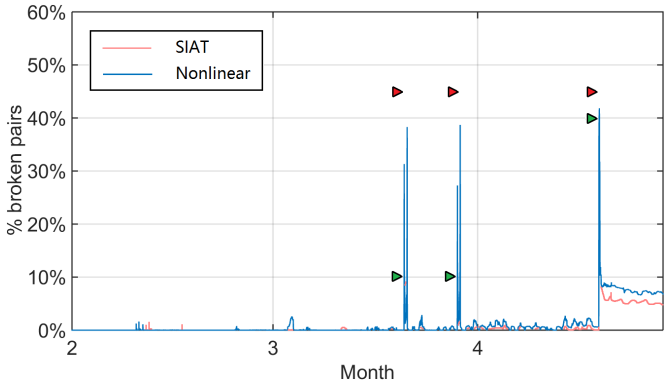

Figure 4: Case study of anomaly detection. Trained on Month 1 and Tested on Month 2, 3, 4.

We compared our method with SIAT, a linear regression-based pairwised anomaly detection method. Our method obtained $23,231$ related pairs, which involved 92% of sensors. Figure 4 illustrates the number of broken relationships of our method and SIAT along time. In the figure, there were only a few broken relationships scattered in the second month and the first half of the third month, which are normal operational deviations and indicated a healthy status. Two clusters of spikes were found in the second half of the third month, where over 30% of learned relationships were broken in our method. Similarly, a cluster of spikes existed in the fourth month. We confirmed this was due to the system operators. These were pipe failures inside the system and lasted for several hours. They also propagated to affect other components, which resulted in previously learned relationships being widely broken. Our method detected the failures about two hours earlier than the operators, because it found a gradual increase in the number of broken relationships, which started earlier than the failure. After fixing the third issue, the operators reset some components, so the previously learned relationships no longer held. The model started to persistently break after the third failure.

However, the pairs generated by SIAT involved less than 10% of sensors, which resulted in most of the sensors related to failures not been modeled. It could not detect the first 2 alerts because very few relationships were broken to reach the alert level. The consistent broken signals from system reset in the fourth month were detected by SIAT.

4. Conclusions

In this paper, we have designed a solution to identify and quantify nonlinear structured relationships in complex manufactural systems. We formulate the problem as a Sparse Group LASSO problem with overlapping groups and design a novel algorithm following the FISTA framework. Sample correlation structure is also estimated and accounted for in our model, which improves the discovery of relationships. The evaluations on the synthetic data and two real-world examples illustrate the accuracy and usefulness of our method.

References

[1] Z. Han, H. Chen, T. Yan, and G. Jiang, "Time series segmentation to discover behavior switching in complex physical systems," in *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 161–170.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[3] B. Liu, H. Chen, A. Sharma, G. Jiang, and H. Xiong, "Modeling heterogeneous time series dynamics to profile big sensor data in complex physical systems," in *Big Data, 2013 IEEE International Conference on*. IEEE, 2013, pp. 631–638.

[4] A. B. Sharma, H. Chen, M. Ding, K. Yoshihira, and G. Jiang, "Fault detection and localization in distributed systems using invariant relationships," in *2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2013, pp. 1–8.

[5] G. Jiang, H. Chen, and K. Yoshihira, "Discovering likely invariants of distributed transaction systems for autonomic system management," in *2006 IEEE International Conference on Autonomic Computing*. IEEE, 2006, pp. 199–208.

[6] J. Yin, X. Chen, and E. P. Xing, "Group sparse additive models," in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012, pp. 871–878.

[7] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman, "Sparse additive models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 5, pp. 1009–1030, 2009.

[8] G. A. Seber, "Nonlinear regression models," in *The Linear Model and Hypothesis*. Springer, 2015, pp. 117–128.

[9] Y. Plan and R. Vershynin, "The generalized lasso with non-linear observations," *IEEE Transactions on Information Theory*, vol. 62, no. 3, pp. 1528–1537, 2016.

[10] R. Chen and R. S. Tsay, "Nonlinear additive arx models," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 955–967, 1993.

[11] R. A. Serway and C. Vuille, *College physics*. Cengage Learning, 2014.

[12] J. O. Smith, *Mathematics of the discrete Fourier transform (DFT): with audio applicaitons*. Julius Smith, 2007.

[13] H. J. Nussbaumer, *Fast Fourier transform and convolution algorithms*. Springer Science & Business Media, 2012, vol. 2.

[14] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[15] K. Y. Liang and S. L. Zeger, "Longitudinal data-analysis using generalized linear-models," *Biometrika*, vol. 73, no. 1, pp. 13–22, 1986.

[16] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, 2013.

[17] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing, "An efficient proximal gradient method for general structured sparse learning," *stat*, vol. 1050, p. 26, 2011.

[18] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," *arXiv preprint arXiv:1001.0736*, 2010.

[19] P. Gong, J. Ye, and C. Zhang, "Robust multi-task feature learning," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2012, pp. 895–903.

[20] T. Xu, J. Sun, and J. Bi, "Longitudinal lasso: Jointly learning features and temporal contingency for outcome prediction," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: ACM, 2015, pp. 1345–1354. [Online]. Available: http://doi.acm.org/10.1145/2783258.2783403