

Optimal Input Signal Design for Data-Centric Estimation Methods

Sunil Deshpande and Daniel E. Rivera

Abstract—Data-centric estimation methods such as Model-on-Demand and Direct Weight Optimization form attractive techniques for estimating unknown functions from noisy data. These methods rely on generating a local function approximation from a database of regressors at the current operating point with the process repeated at each new operating point. This paper examines the design of optimal input signals formulated to produce informative data to be used by local modeling procedures. The proposed method specifically addresses the distribution of the regressor vectors. The design is examined for a linear time-invariant system under amplitude constraints on the input. The resulting optimization problem is solved using semidefinite relaxation methods. Numerical examples show the benefits in comparison to a classical PRBS input design.

I. INTRODUCTION

Most real-world phenomena is nonlinear in nature. Towards this, there have been various approaches to nonlinear black-box system identification from experimental data such as neural networks, wavelets and Nonlinear AutoRegressive with eXogenous input model (NARX) [1]–[3]. The conventional modeling philosophy is to derive a ‘global’ mathematical representation using the input-output data and post validation, the dataset is discarded. In general, the associated optimization problem for these structures is nonconvex, thus resulting in computational issues. An alternative approach is to build a ‘local’ function approximation (which can be well approximated by linear models) *on-line* based on the current operating point. In the system identification literature, this technique is called Just-in-Time learning [4], Model-on-Demand (MoD) [5]–[7] and Direct Weight Optimization (DWO) [8]. Unlike global black-box approaches, these on-line local modeling methods are computationally tractable and well complemented by modern computing. In addition, on-line local methods can readily utilize any new data available. These local linear models can then be used by a model predictive controller, for example, for control of uncertain systems [6], [7] and nonlinear hybrid systems [9].

An area of recent interest for these methods is adaptive behavioral interventions, where intervention components are adjusted by a controller based on participant response over time [10]. These interventions are characterized by multiple participants with significant individual variability, with data being continuously collected during the process. The data-centric modeling approach is much more suitable for such

complex scenarios where building a global model is difficult and inefficient. A Model-on-Demand model predictive controller (MoDMPC) for adaptive interventions has been proposed in [9].

Traditional experiment design for system identification considers some scalar measure of the parameter covariance matrix with specific requirements on the input signal spectrum [1], [11]. Although the field is quite vast, none of these methods, to the best of our knowledge, directly address design issues in the regressor space. In this paper, an input signal design methodology is presented to generate informative data for data-centric identification algorithms by specifically addressing the distribution of the regressor vectors. Prior work for data-centric estimation addressed distribution of only the outputs using Weyl’s criterion [12], [13]. Constraints on the input and output are incorporated in this approach to achieve plant-friendly operation [11], [13]–[15]. For the purposes of this paper, the focus is specifically on the requirements for Model-on-Demand by considering linear time-invariant models under amplitude constraints on the input. The subsequent optimization problem corresponds to a maximization of convex quadratic function which is NP-hard in general [16]. A semidefinite relaxation of the nonconvex quadratic problem is proposed to obtain an approximate solution in polynomial time. A case study is undertaken to examine the feasibility of this approach.

The paper is organized as follows: Section II briefly describes the MoD data-centric modeling methodology used in this paper. In Section III, the problem formulation is presented with discussion on its implications, with semidefinite relaxation discussed in Section IV. Numerical examples are shown in Section V and summary, conclusions and directions of future work are presented in Section VI.

II. MODEL-ON-DEMAND ESTIMATION

The methodology of building local linear models on-line or Model-on-Demand [5] is now briefly described. Consider a SISO process for a given data set $(\{y(k), \varphi(k)\}_{k=1}^N)$ as

$$y(k) = f(\varphi(k)) + e(k) \quad k = 1, \dots, N \quad (1)$$

where $y(k) \in \mathbb{R}$, $f(\cdot)$ is an unknown (nonlinear) mapping, $\varphi(k) \in \mathbb{R}^m$ is the regressor vector (generally composed of lagged output and input)

$$\varphi(k) = [y(k-1) \dots y(k-n_a) \\ u(k-1) \dots u(k-n_b)]^T \quad (2)$$

and $e(k) \in \mathbb{R}$ is the noise. The MoD predictor attempts to estimate output predictions based on a local neighborhood

Sunil Deshpande and Daniel E. Rivera are with the Control Systems Engineering Laboratory (CSEL), Arizona State University, Tempe, AZ, USA. Sunil Deshpande is a doctoral student in the electrical engineering program at Arizona State. Daniel E. Rivera is with the School for Engineering of Matter, Transport, and Energy. [email: {sdeshpa2, daniel.rivera}@asu.edu]

of desired operating point (φ^*) in the regressor space. The predictor function f is obtained by a linear combination of observed outputs

$$\hat{f}(\varphi^*) = \sum_{k=1}^N w(k)y(k) \quad (3)$$

where the weights are, in general, dependent on the distance of regressors from the current operating point ($\varphi^* - \varphi(k)$), noise variance and properties of f . The weights emphasize the size of the neighborhood from the desired operating point, and this is referred to as ‘bandwidth’ of the estimator. Thus, it governs the tradeoff between bias and variance errors of the estimate. To select these weights, the following two methods can be used:

- *Kernel-based approach*: The weights are assigned as per a kernel or window function ($W(\cdot)$) according to the distance of given regressors from φ^* to asymptotically minimize the mean square error of the estimate [1], [5], [17]

$$w(k) = W\left(\frac{\|\varphi^* - \varphi(k)\|_M}{h}\right) \quad (4)$$

where $M \in \mathbb{R}^{m \times m}$ is scaling matrix for the Euclidean distance and h is the bandwidth.

- *Optimization-based approach*: The choice of weights using statistical measures is as per asymptotic arguments and in practice, the number of data points is finite. Another approach of selecting these weights is through explicit use of optimization by minimizing the exact mean square error (MSE) or worst case MSE

$$w^* = \arg \min_w W(\varphi^* - \varphi(1), \dots, \varphi^* - \varphi(N), f, w) \quad (5)$$

where $w = [w(1), \dots, w(N)]^T$ and $W(\cdot)$ is some function of the MSE [6], [8]. Consequently, some of the resulting weights are zero and thus the ‘bandwidth’ of the estimator is automatically calculated.

The methods outlined so far have focused on obtaining the function approximation $\hat{f}(\varphi^*)$. In the next step, a local linear estimate can be obtained by solving

$$\hat{\beta}^* = \arg \min_{\hat{\beta}} \sum_{k=1}^N \ell(y(k) - \hat{m}(\varphi^* - \varphi(k), \hat{\beta})) \times W(\cdot) \quad (6)$$

where $\ell(\cdot)$ is a quadratic norm function, $W(\cdot)$ assigns weight as per the estimator bandwidth and the local model structure can be:

$$\hat{m}(\varphi^* - \varphi(k), \hat{\beta}) = \hat{\beta}_0 + \hat{\beta}_1(\varphi^* - \varphi(k)) \quad (7)$$

which is linear in the unknown parameters and hence an estimate can be computed using least squares.

III. MoD INPUT SIGNAL DESIGN

The quality of the estimate from MoD depends primarily on the following factors:

- Regressor vector:
 - The structure of the regressor vector, i.e., the number of lagged outputs and inputs. The regressor

structure should correspond to what is suitable for a local linear model (e.g., from linearization of an *a priori* nonlinear model).

- The *distribution* in the regressor space. From (4) and (5), it can be observed that the choice of the weights is a function of distance of φ^* from the available regressor vectors $\varphi(k)$.

- N : number of data points. For high dimensional space and for finite N , the distance between two regressor can become very large. In practice, large number of systems can be well approximated by low order regressors.
- Properties of f : smoothness in general and in particular, bounds on the Hessian matrix.
- Noise in the experimental data.

From (3), it is clear that the estimate is formed by interpolation of available outputs. In other words, if the current operating point (φ^*) lies beyond the convex hull of available regressor points, the estimator would have to extrapolate rather than interpolate. Thus, the aim of input signal is to excite the full span of output so to make sure that the estimator is always interpolating. These observations lead to the following problem statement:

Problem Statement: Given N regressor vectors of fixed finite dimension, distribute the regressor points as far apart from each other as possible in the regressor space under constraints on the input and output signals.

A. Regressor Structure

The regressor vector contains finite lagged data about the system. The Autoregressive with exogenous input (ARX) regressor structure is commonly used [2], [6]:

$$\varphi(k) = [y(k-1) \dots y(k-n_a) \quad u(k-n_k) \dots u(k-n_b-n_k+1)]^T \quad (8)$$

where n_a , n_b and n_k denote the number of previous output, input and the degree of delay in the model and $\varphi(k) \in \mathbb{R}^m$ where $m = n_a + n_b$. It is assumed that the dimension of the regressor vector is selected by the user. In general, it is useful to start with $n_k = 1$ without *a priori* information.

B. Problem Formulation

Consider an input signal $u \in \mathbb{R}^N = [u(1), \dots, u(N)]^T$ and corresponding output signal $y \in \mathbb{R}^N = [y(1), \dots, y(N)]^T$ generated by a linear time-invariant system represented as:

$$y = Gu \quad (9)$$

where $G \in \mathbb{R}^{N \times N}$ is the Toeplitz matrix of system impulse responses. To be used by the optimization procedure, the regressor vector has to be parameterized. For the purpose of illustration, consider the ARX regressor vector as shown in (8). The regressor vector $\varphi(k)$ can be written in terms of u as:

$$\varphi(k) = \begin{bmatrix} P_k G \\ Q_k \end{bmatrix} u \quad (10)$$

where $P_k \in \mathbb{R}^{n_a \times N}$ and $Q_k \in \mathbb{R}^{n_b \times N}$. One can readily observe that these matrices are sparse and the rank of $P_k, Q_k = \max\{n_a, n_b\}$. The case of Finite Impulse Response (FIR) regressor structure can be derived as a special case

$$\varphi(k) = \begin{bmatrix} 0_{n_a \times N} \\ Q_k \end{bmatrix} u. \quad (11)$$

For the FIR case, the dimension of the regressor m has to be generally large to capture all of the dynamics. In contrast, a relatively low dimension is often suitable in the case of ARX structure. Before presenting the problem objective, a distance pair of two regressors has to be defined. It can be readily seen that for given N regressor points there will be ${}^N C_2 = N(N-1)/2$ unique distance pairs and hence the number of distance pairs scales polynomially in N i.e. number of distance pairs $\simeq N^2/2 \forall N \gg 1$.

The notion of distance in the Euclidean space \mathbb{R}^m can be defined using any p -norm distance. In this formulation, the Euclidean distance (2-norm distance) is used as the metric which is also used to calculate the bandwidth of the estimator as discussed in Section II. Based on the parametrization shown in (10), the square of the Euclidean distance between two regressor points d_{ij} can be defined as:

$$d_{ij}^2 = \|\varphi(i) - \varphi(j)\|^2 \quad i, j \in \{1, \dots, N\} \quad (12)$$

$$\begin{aligned} &= \left\| \begin{bmatrix} (P_i - P_j)G \\ Q_i - Q_j \end{bmatrix} u \right\|^2 \\ &= u^T \underbrace{\begin{bmatrix} (P_i - P_j)G \\ Q_i - Q_j \end{bmatrix}^T \begin{bmatrix} (P_i - P_j)G \\ Q_i - Q_j \end{bmatrix}}_{\mathbf{Q}_{ij}} u. \end{aligned} \quad (13)$$

The problem statement defined previously states that the objective is to distribute the points as far apart as possible. This directly implies that the sum of all the distances will have to be maximized. This objective can be mathematically represented as:

$$\max \sum_i^N \sum_{j=i+1}^N d_{ij}^2 \quad i \in \{1, \dots, N\}. \quad (14)$$

For the problem to be bounded, it is necessary to assume amplitude constraints on the input

$$u_{\min} \leq u \leq u_{\max} \quad (15)$$

where, for the purpose of illustration, the bounds are symmetric ($u_{\min} = -u_{\max}$). Based on (13), (14) and (15), the formal optimization problem can be written as

$$\begin{aligned} \max_u \quad & u^T \mathbf{Q} u \\ \text{s.t.} \quad & u_{\min} \leq u \leq u_{\max} \end{aligned} \quad (16)$$

where $\mathbf{Q} = \sum_i^N \sum_{j=i+1}^N \mathbf{Q}_{ij}$ is defined by the sum of distance pairs as per (14). It can be noted that \mathbf{Q}_{ij} is positive semidefinite ($\mathbf{Q}_{ij} \succeq 0$) as it represents the square of the distance. Hence, \mathbf{Q} which is defined as the sum of positive semidefinite matrices is also positive semidefinite ($\mathbf{Q} \succeq 0$). This implies that the resulting quadratic maximization problem (or minimization of negative semidefinite problem) is NP-hard [16].

IV. CONVEX RELAXATION

Convex relaxation of nonconvex optimization problems has developed as a powerful approach to solve (approximately, and sometimes exactly) otherwise hard problems [18]–[20]. In a relaxation approach, the idea is to identify the main source of nonconvexity and then either drop it or replace it with a ‘relaxed’ version that is convex and hence more tractable. Clearly, there has to be a meaningful relationship between the original nonconvex problem and the relaxed convex problem. Semidefinite programming (SDP) relaxation (also dual of the Lagrangian dual) for the case of quadratic problems has been particularly successful with proven bounds on suboptimality for many scenarios [21].

A. Semidefinite Relaxation

Consider the problem in (16) which is restated here for completeness.

$$\begin{aligned} \nu_{\text{QP}} = \max_u \quad & u^T \mathbf{Q} u \\ \text{s.t.} \quad & u_{\min} \leq u \leq u_{\max} \end{aligned} \quad (17)$$

where ν_{QP} is the objective value of the original nonconvex problem. First, by change of variables

$$u^T \mathbf{Q} u = \text{Tr}(\mathbf{Q}(uu^T)) = \text{Tr}(\mathbf{Q}U) \quad (18)$$

where $\text{Tr}(\cdot)$ is the trace operator and $uu^T = U \in \mathbb{R}^{N \times N}$. Similarly, (15) can be written in terms of variable U . Thus, the original problem can be written as

$$\begin{aligned} \max_{u, U} \quad & \text{Tr}(\mathbf{Q}U) \\ \text{s.t.} \quad & \text{diag}(U) \leq u_{\max}^2 \\ & U = uu^T \end{aligned} \quad (19)$$

where $\text{diag}(\cdot)$ denotes vector formed from the matrix’s diagonal. This problem (19) is exactly equivalent to (17) and hence nonconvex. The key step in relaxing this problem is to first note that

$$U = uu^T \iff U \succeq 0, \text{rank}(U) = 1. \quad (20)$$

Hence, the nonconvexity of the problem has been transformed into the rank constraint. Now problem (19) can be relaxed into a SDP problem by neglecting the (nonconvex) rank constraint on U

$$\begin{aligned} \nu_{\text{SDP}} = \max_U \quad & \text{Tr}(\mathbf{Q}U) \\ \text{s.t.} \quad & \text{diag}(U) \leq u_{\max}^2 \\ & U \succeq 0, \end{aligned} \quad (21)$$

where ν_{SDP} gives an upper bound on the nonconvex maximization problem in (17) i.e. $\nu_{\text{QP}} \leq \nu_{\text{SDP}}$.

B. Feasible Input and Bounds on Suboptimality

The solution of (21) would yield an objective value ν_{SDP} and the optimal matrix variable U^* . Given that this was a relaxation and bounds are generally not exact, the rank of U^* will not be unity and hence it is not possible to decompose as $U^* = u^*u^{*T}$. Hence, the SDP only gives the bound on the

problem and no direct feasible solution. The general process in such cases is to generate feasible inputs through a process of randomization. In this problem, a random feasible input is sampled from a Gaussian distribution with covariance matrix U^* [21]. For the case of maximization, it can be observed that the maximum values will be obtained when the input is either u_{\min} or u_{\max} . In other words, a binary signal will attain the maximum value. When enough random samples are picked, a good suboptimal point can be obtained [19]. Generally for any given SDP relaxation, there is no bound on this suboptimality. For this maximization structure, it can be shown that there not only exists a hard finite upper bound (as with any feasible SDP relaxation), but also a hard lower bound on the solution.

Theorem 1 (Nesterov [22], Ye [23]): Given the quadratic optimization problem

$$\begin{aligned} \nu_{QP} = \max_u \quad & u^T \mathbf{Q} u \\ \text{s.t.} \quad & u_{\min} \leq u \leq u_{\max} \end{aligned} \quad (22)$$

where $\mathbf{Q} \succeq 0$, the approximation bounds can be given by

$$0.63\nu_{SDP} \leq \nu_{QP} \leq \nu_{SDP}. \quad (23)$$

This result holds for both FIR and ARX type regressors. For the FIR case, Q has additional properties and a tighter bound holds true:

Theorem 2 (Goemans and Williamson [24]): Given the quadratic optimization problem

$$\begin{aligned} \nu_{QP} = \max_u \quad & u^T \mathbf{Q} u \\ \text{s.t.} \quad & u_{\min} \leq u \leq u_{\max} \end{aligned} \quad (24)$$

where $\mathbf{Q} \succeq 0$ and all off-diagonal elements are nonpositive ($\mathbf{Q}(c, d) \leq 0$, $c \neq d$), the approximation bounds can be given by

$$0.87\nu_{SDP} \leq \nu_{QP} \leq \nu_{SDP}. \quad (25)$$

This fact is simple to confirm based on (11) and (13).

C. Extensions

So far, only amplitude constraints on the input have been considered. Since the problem is formulated in the decision variable u , any time-domain constraints can be included. An important constraint, for example, is bound on the output based on a given model. More time-domain constraints are discussed in [13] which can be written as linear inequalities and hence can also be included. It should be noted that it is unlikely that bounds on suboptimality exist in those scenarios. However, in practice good objective values are found although generation of feasible inputs remains nontrivial and is not considered in this paper.

V. NUMERICAL EXAMPLE

Consider a continuous-time second order system with a zero

$$G(s) = \frac{K_p(\tau_a s + 1)}{\tau^2 s^2 + 2\zeta\tau s + 1}. \quad (26)$$

where $K_p = 1$, $\tau_a = 2$, $\tau = 3$ and $\zeta = 0.6$. The system step response is shown in Figure 1. This model was discretized

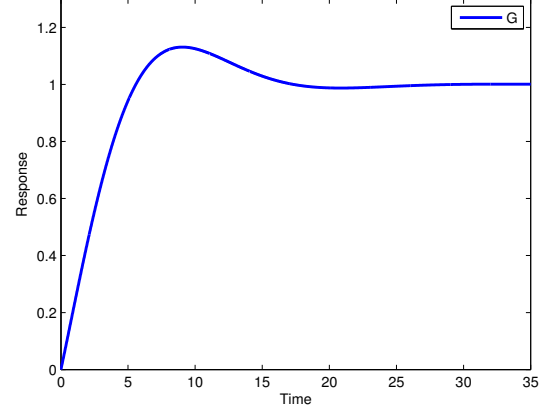


Fig. 1: Step response of system as shown in (26).

at unit sampling using zero-order hold. This system can be fully parameterized by an ARX regressor of order [2 2 1]

$$\varphi(k) = [y(k-1) \ y(k-2) \ u(k-1) \ u(k-2)]^T. \quad (27)$$

The output prediction at time k can be written as

$$\hat{y}(k) = \varphi(k)^T \theta \quad (28)$$

where $\theta = [1.58, -0.6703, 0.2282, -0.1375]^T$ based on the discretization of (26).

The optimal input was generated by solving the semidefinite program shown in (21). For purposes of simulation, it was assumed $u_{\min} = [-1, \dots, -1]$, $u_{\max} = [1, \dots, 1]$. The SDP was coded in MATLAB with YALMIP [25] interface using SeDuMi [26] as the SDP solver. Using the SDP results, a feasible input was generated using randomization. This binary (sub) optimal signal is compared with a classical Pseudo Random Binary Sequence (PRBS) [1]. A PRBS signal is uniquely defined by switching time T_{sw} and the number of shift registers n_r to excite the system bandwidth:

$$\omega_* = \frac{1}{\beta_s \tau_{\text{dom}}^H} \leq \omega \leq \frac{\alpha_s}{\tau_{\text{dom}}^L} = \omega^*, \quad (29)$$

where typically $\alpha_s = 2$ and $\beta_s = 3$ [12]. Based on these specifications, the switching time can be calculated as:

$$T_{sw} \leq \frac{2.8\tau_{\text{dom}}^L}{\alpha_s} \quad (30)$$

and the length of signal (N) can be calculated as:

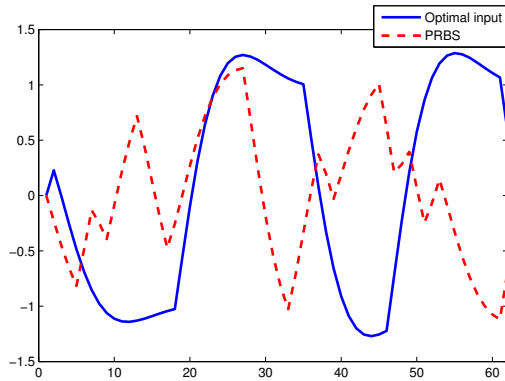
$$N = 2^{n_r} - 1 \geq \frac{2\pi\beta_s\tau_{\text{dom}}^H}{T_{sw}} \quad (31)$$

The time constant was found to be $\tau_{\text{dom}}^L = \tau_{\text{dom}}^H = 2.14$ from (26) and consequently the length of the PRBS input for one period was found to be $N = 62$.

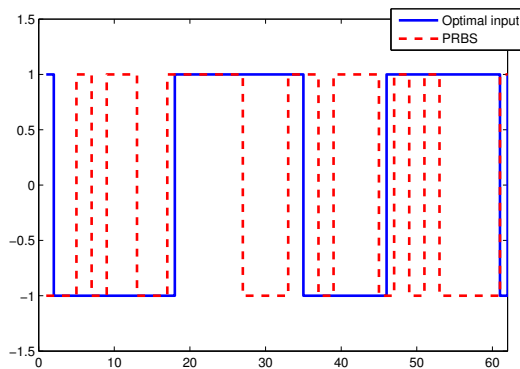
Table I compares the objective values for the two signals. For the optimal signal, ν_{SDP} is the maximum value that can be attained by the maximization problem shown in (16). Based on Theorem 1, a lower bound on the objective can be assigned as shown. The randomization procedure then

TABLE I: Tabulation of sum of unique distances between all regressors for the example problem (26).

Upper bound (ν_{SDP})	16038
Lower bound ($0.63\nu_{SDP}$)	10104
Best objective from randomization	$14516 \simeq (0.9\nu_{SDP})$
PRBS objective	10065



(a) Dynamic simulation of outputs.



(b) Optimal input and PRBS.

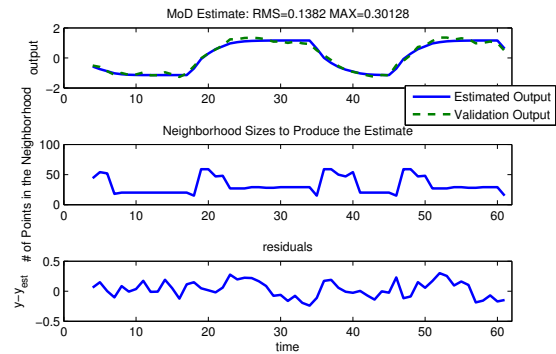
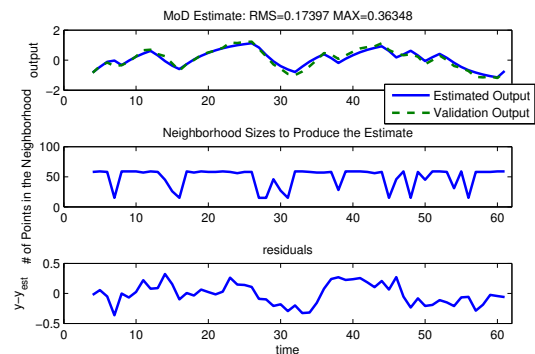
Fig. 2: Input-output simulation for example system shown in (26) under optimal input and PRBS input.

produces an input whose objective value lies in this bound. In this case, it was found that the best objective value from 20000 candidates was 0.9 times of the SDP objective. It is worth noting that it is not known where the global optimal lies, but that it lies between the upper and lower bound calculated from the SDP problem. Further, the objective value for PRBS was lower than the guaranteed objective from the optimal input. In fact, the randomization procedure gave an objective value improvement of 44.2% over the PRBS objective.

Figure 2 compares the optimal input and the PRBS input, as shown in Figure 2b, with the resulting output signal as shown in Figure 2a. An interesting observation is to note that the PRBS input switches more frequently to emphasize the frequency band where as the optimal signal switches relatively less frequently, and thereby covers more output

TABLE II: Tabulation of average rms errors over 100 simulations from the MoD estimator for two input signals.

σ_{val}	Optimal input avg. rms	PRBS avg. rms
0	0.10593	0.1441
0.1	0.14817	0.1723
0.2	0.22642	0.2362
0.5	0.50942	0.52259
1	1.0063	1.0171


 Fig. 3: MoD estimate for the dataset generated from the optimal input under noisy conditions ($\sigma_{est} = 0.1, \sigma_{val} = 0.1$).

 Fig. 4: MoD estimate for the dataset generated from the PRBS input under noisy conditions ($\sigma_{est} = 0.1, \sigma_{val} = 0.1$).

span than the PRBS input as shown in Figure 2a.

While the optimal input is not directly designed to minimize root mean square (rms) estimation error, it is interesting to note how it performs in comparison to PRBS input under noisy conditions. Towards this, a database of regressors from the two inputs and their corresponding outputs was created. In practice, the data is corrupted so a zero mean Gaussian noise with standard deviation $\sigma_{est} = 0.1$ was added to the estimation dataset. Next, sets of input-output validation dataset were created with output corrupted by zero mean Gaussian noise with standard deviation σ_{val} . The function values were estimated from the MoD estimator and compared with true predictions as per (28). Average rms values of error taken over 100 simulations for each σ_{val} are tabulated in Table II. It can be observed that, with increasing noise,

the average estimation error from the optimal input dataset is consistently of a lower value than from the PRBS input dataset. The improvement in rms error is 26.4% for no noise in the validation set ($\sigma_{\text{val}} = 0$) and approximately 1% under very noisy conditions ($\sigma_{\text{val}} = 1$). One simulation case under noise with standard deviations $\sigma_{\text{est}} = 0.1, \sigma_{\text{val}} = 0.1$ is shown in Figure 3 for the optimal input, and in Figure 4 for the PRBS input. These figures plot the simulated output along with the number of data points required from the neighborhood and the estimation error. For this case, the rms error for the optimal input (0.1382) is lower than rms error for the PRBS input (0.17397) under similar noisy conditions.

VI. SUMMARY, CONCLUSIONS AND FUTURE WORK

This paper introduces an approach for optimal input signals for data-centric estimation algorithms such as Model-on-Demand. The objective is to distribute the regressors in the given finite dimensional space, as measured by their Euclidean distances, so that they are as far apart as possible under dynamical constraints. To illustrate this proposition, a linear time-invariant system is considered subject to amplitude constraints on the input. The resulting nonconvex quadratic optimization problem was approximately solved through the method of semidefinite relaxation. When using the ARX regressor structure, generation of optimal input requires knowledge of the true system; this is a known issue in optimal experiment design [1]. However, for the FIR structure the regressor is only a function of the input and has tighter relaxation bounds. The total sum of all regressor distances was evaluated where the optimal input offers significant improvement over the PRBS input.

In future work, the formulation will be extended to account for distribution of regressors under stochastic and deterministic disturbances, knowledge of model uncertainty and additional time-domain constraints of practical importance. Further, this formulation can be extended to the case of MISO systems by stacking the individual SISO regressors to form a MISO regressor, and hence results from SISO systems naturally carry over to the MISO systems. It is expected that this design approach will provide significant improvements in models from data-centric methods for nonlinear systems and this is currently under investigation.

ACKNOWLEDGMENT

Support for this work has been provided by the Office of Behavioral and Social Sciences Research (OBSSR) of the National Institutes of Health (NIH) and the National Institute on Drug Abuse (NIDA) through grants R21 DA024266 and K25 DA021173. We also thank ASU Advanced Computing Center for providing the Saguaro cluster computing platform.

REFERENCES

- [1] L. Ljung, *System identification :theory for the user*. Upper Saddle River, NJ: Prentice Hall PTR, 1999.
- [2] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.
- [3] L. Ljung, "Perspectives on system identification," *Annual Reviews in Control*, vol. 34, no. 1, pp. 1–12, 2010.
- [4] G. Cybenko, "Just-in-Time learning and estimation," in *Identification, Adaptation, Learning. The Science of Learning Models from data*, ser. NATO ASI Series, S. Bittanti and G. Picci, Eds. Springer, 1996, pp. 423–434.
- [5] M. W. Braun, D. E. Rivera, and A. Stenman, "A 'Model-on-Demand' identification methodology for non-linear process systems," *International Journal of Control*, vol. 74, no. 18, pp. 1708–1717, 2001.
- [6] A. Stenman, "Model on Demand: Algorithms, analysis and applications," Ph.D. dissertation, Linköping University, Sweden, 1999.
- [7] M. Braun, "Model-on-Demand nonlinear estimation and model predictive control: Novel methodologies for process control and supply chain management," Ph.D. dissertation, Arizona State University, USA, 2001.
- [8] J. Roll, A. Nazin, and L. Ljung, "Nonlinear system identification via direct weight optimization," *Automatica*, vol. 41, no. 3, pp. 475–490, 2005.
- [9] N. Nandola and D. E. Rivera, "Model-on-Demand predictive control for nonlinear hybrid systems with application to adaptive behavioral interventions," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, Dec. 2010, pp. 6113–6118.
- [10] S. Deshpande, N. Nandola, D. E. Rivera, and J. Younger, "A control engineering approach for designing an optimized treatment plan for fibromyalgia," in *Proceedings of the 2011 American Control Conference (ACC)*, 2011, pp. 4798–4803.
- [11] D. E. Rivera, H. Lee, H. Mittelmann, and M. W. Braun, "Constrained multisine input signals for plant-friendly identification of chemical process systems," *Journal of Process Control*, vol. 19, no. 4, pp. 623–635, 2009.
- [12] —, "High-purity distillation," *IEEE Control Systems Magazine*, vol. 27, no. 5, pp. 72–89, Oct. 2007.
- [13] S. Deshpande, D. E. Rivera, and J. Younger, "Towards patient-friendly input signal design for optimized pain treatment interventions," in *Proceedings of the 16th IFAC Symposium on System Identification*, July 2012, pp. 1311–1316.
- [14] I. R. Manchester, "Input design for system identification via convex relaxation," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, Dec. 2010, pp. 2041–2046.
- [15] S. Narasimhan and R. Rengaswamy, "Plant friendly input design: Convex relaxation and quality," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1467–1472, 2011.
- [16] P. Pardalos and S. Vavasis, "Quadratic programming with one negative eigenvalue is NP-hard," *Journal of Global Optimization*, vol. 1, pp. 15–22, 1991.
- [17] J. Fan and I. Gijbels, *Local polynomial modeling and its applications*. Chapman & Hall, 1996.
- [18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [19] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*. SIAM, 2001.
- [20] M. Laurent, "Sums of squares, moment matrices and optimization over polynomials," in *Emerging Applications of Algebraic Geometry*, ser. The IMA Volumes in Mathematics and its Applications, M. Putinar and S. Sullivant, Eds. Springer New York, 2009, vol. 149, pp. 157–270.
- [21] Z.-Q. Luo, W.-K. Ma, A. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, May 2010.
- [22] Y. Nesterov, "Semidefinite relaxation and nonconvex quadratic optimization," *Optimization Methods and Software*, vol. 9, no. 1-3, pp. 141–160, 1998.
- [23] Y. Ye, "Approximating quadratic programming with bound and quadratic constraints," *Mathematical Programming*, vol. 84, pp. 219–226, 1999.
- [24] M. Goemans and D. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.
- [25] J. Löfberg, "YALMIP : a toolbox for modeling and optimization in matlab," in *2004 IEEE International Symposium on Computer Aided Control Systems Design*, 2004, pp. 284–289.
- [26] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999.