Learning of Linear Dynamical Systems as a Non-Commutative Polynomial Optimization Problem

Quan Zhou*, Jakub Mareček‡
*Imperial College London, q.zhou22@imperial.ac.uk
‡Czech Technical University in Prague, jakub.marecek@fel.cvut.cz

***Abstract*—There has been much recent progress in time series forecasting and estimation of system matrices of linear dynamical systems (LDS). We present an approach to both problems based on an asymptotically convergent hierarchy of convexifications of a certain non-convex operator-valued problem, which is known as non-commutative polynomial optimization (NCPOP). We present promising computational results, including a comparison with methods implemented in Matlab System Identification Toolbox.**

I. Introduction

We consider the identification of vector autoregressive processes with hidden components from time series of observations, which is a key problem in system identification [32]. Its applications range from the identification of parameters in epidemiological models [3] and reconstruction of reaction pathways in other biomedical applications [10], to identification of models of quantum systems [5, 6]. Beyond this, one encounters either partially observable processes or questions of causality [44, 14] in almost any application domain. In the "prediction-error" approach to forecasting [32], it allows the estimation of subsequent observations in a time series.

To state the problem formally, let us define a linear dynamic system (G, F, V, W) as in [58].

$$\begin{aligned} \phi_t &= G\phi_{t-1} + \omega_t, \\ Y_t &= F'\phi_t + \nu_t, \end{aligned} \tag{1}$$

where $\phi_t \in \mathbb{R}^{n\times 1}$ is the hidden state, $Y_t \in \mathbb{R}^{m\times 1}$ is the observed output (measurements, observations), $G \in \mathbb{R}^{n\times n}$ and $F \in \mathbb{R}^{n\times m}$ are system matrices, and $\{\omega_t, \nu_t\}_{t\in\mathbb{N}}$ are normally distributed process and observation noises with zero mean and covariance of W and V respectively. The transpose of F is denoted as F'. Learning (or proper learning) refers to identifying the quadruple (G, F, V, W) given the output $\{Y_t\}_{t\in\mathbb{N}}$. We assume that the linear dynamical system (G, F, V, W) is observable [38], i.e., its observability matrix [58] has full rank. Note that a minimal representation is necessarily observable and controllable, cf. Theorem 4.1 in [49], so the assumption is not too strong.

There are three complications. First, the dimension n of the hidden state ϕ_t is not known, in general. Although [43] have shown that a lower-dimensional model can approximate a higher-dimensional one rather well, in many cases, it is hard to choose n in practice. Second, the corresponding optimization problem is non-convex, and guarantees of global convergence have been available only for certain special cases. Finally, the operators-valued optimization problem is non-commutative, and hence much work on general-purpose commutative non-convex optimization is not applicable without making assumptions [5, cf.] on the dimension of the hidden state.

Here, we aim to develop a method for proper learning of LDS that could also estimate the dimension of the hidden state and that would do so with guarantees of global convergence to the best possible estimate, given the observations. This would promote explainability beyond what forecasting methods without global convergence guarantees allow for. In particular, our contributions are:

- We cast learning of a linear dynamical system with an unknown dimension of the hidden state as a non-commutative polynomial optimization problem (NCPOP). This also makes it possible to utilize prior information as shape constraints in the NCPOP.
- We show how to use Navascules-Pironio-Acin (NPA) hierarchy [37] of convexifications of the NCPOP to obtain bounds and guarantees of global convergence. The runtime is independent of the (unknown) dimension of the hidden state.
- In two well-established small examples of [30, 18, 26], our approach outperforms standard subspace and least squares methods, as implemented in Matlab™ System Identification Toolbox™.

II. Background

First, we set our work in the context of related work. Next, we provide a brief overview of non-commutative polynomial optimization, pioneered by [41] and nicely surveyed by [8], which is our key technical tool. Prior to introducing our own results, we introduce some common notation, following [58].

A. Related Work in System Identification and Control

There is a long history of research within system identification [32]. In forecasting under LDS assumptions (improper learning of LDS), a considerable progress has been made in the analysis of predictions for the expectation of the next measurement using auto-regressive (AR) processes in Statistics and Machine Learning. In [2], first guarantees were presented for auto-regressive moving-average (ARMA) processes. In [30], these results were extended to a subset of autoregressive integrated moving average (ARIMA) processes. [26] have shown that up to an arbitrarily small error given in advance, AR(s) will perform as well as *any* Kalman filter on any bounded sequence. This has been extended by [52] to Kalman filtering with logarithmic regret.

Another stream of work within improper learning focuses on sub-space methods [23, 38] and spectral methods [18, 17]. [50, 51] presented the present-best guarantees for traditional sub-space methods. [48] utilize regularizations to improve sample complexity. Within spectral methods, [18] and [17] have considered learning LDS with input, employing certain eigenvalue-decay estimates of Hankel matrices in the analyses of an auto-regressive process in a dimension increasing over time. We stress that none of these approaches to improper learning are "prediction-error": They do *not* estimate the system matrices.

In proper learning of LDS, many state-of-the-art approaches consider the least squares method, despite complications encountered in unstable systems [12]. [47] have provided non-trivial guarantees for the ordinary least squares (OLS) estimator in the case of stable G and there being no hidden component, i.e., F' being an identity and $Y_t = \phi_t$. Surprisingly, they have also shown that more unstable linear systems are easier to estimate than less unstable ones, in some sense. [46] extended the results to allow for a certain prefiltering procedure. [42, 43] extended the results to cover stable, marginally stable, and explosive regimes. [39] provide a finite-horizon analysis of the Ho-Kalman algorithm. Most recently, [4] provided a detailed analysis of the use of the method of moments in learning linear dynamical systems, which could be seen as a polynomial-time algorithm for learning a LDS from a trajectory of polynomial length up to a polynomial error. Our work could be seen as a continuation of the work on the least squares method, with guarantees of global convergence.

B. Non-Commutative Polynomial Optimization

Our key technical tool is non-commutative polynomial optimization, first introduced by [41]. Here, we provide a brief summary of their results, and refer to [8] for a book-length introduction. NCPOP is an operator-valued optimization problem with a standard form in (2):

$$\begin{aligned} P^* = \min_{(\mathcal{H}, X, \psi)} \quad & \langle \psi, p(X)\psi \rangle \\ \text{s.t.} \quad & q_i(X) \succcurlyeq 0, i = 1, \dots, m, \\ & \langle \psi, \psi \rangle = 1, \end{aligned} \tag{2}$$

where $X = (X_1, \dots, X_n)$ is a tuple of bounded operators on a Hilbert space $\mathcal{H}$. In contrast to traditional scalar-valued, vector-valued, or matrix-valued optimization techniques, the dimension of variables X is unknown *a priori*. Let $[X, X^\dagger]$ denotes these $2n$ operators, with the †-algebra being conjugate transpose. The normalized vector ψ, i.e., $\|\psi\|^2 = 1$ is also defined on $\mathcal{H}$ with the inner product $\langle \psi, \psi \rangle = 1$. $p(X)$ and $q_i(X)$ are polynomials and $q_i(X) \succcurlyeq 0$ denotes that the operator $q_i(X)$ is positive semidefinite. Polynomials $p(X)$ and $q_i(X)$ of degrees $\deg(p)$ and $\deg(q_i)$, respectively, can be written as:

$$p(X) = \sum_{|\omega| \le \deg(p)} p_\omega \omega, \quad q_i(X) = \sum_{|\mu| \le \deg(q_i)} q_{i,\mu} \mu, \tag{3}$$

where $i = 1, \dots, m$. Monomials ω, μ, u and ν in following text are products of powers of variables from $[X, X^\dagger]$. The degree of a monomial, denoted by $|\omega|$, refers to the sum of the exponents of all operators in the monomial ω. Let $\mathcal{W}_k$ denote the collection of all monomials whose degrees $|\omega| \le k$, or less than infinity if not specified. Following [1], we can define the moments on field $\mathbb{R}$ or $\mathbb{C}$, with a feasible solution $(\mathcal{H}, X, \psi)$ of problem (2):

$$y_\omega = \langle \psi, \omega\ \psi \rangle, \tag{4}$$

for all $\omega \in \mathcal{W}$ and $y_1 = \langle \psi, \psi \rangle = 1$. Given a degree k, the moments whose degrees are less or equal to k form a sequence $y = (y_\omega)_{|\omega| \le 2k}$. We call k as the moment order. With a finite set of moments y of moment order k, we can define a corresponding k^{th}-order moment matrix $M_k(y)$:

$$M_k(y)(\nu, \omega) = y_{\nu^\dagger \omega} = \langle \psi, \nu^\dagger\ \omega\ \psi \rangle, \tag{5}$$

for any $|\nu|, |\omega| \le k$ and the localizing matrix $M_{k_i}(q_i y)$:

$$\begin{aligned} M_{k_i}(q_i y)(\nu, \omega) &= \sum_{|\mu| \le \deg(q_i)} q_{i,\mu} y_{\nu^\dagger \mu \omega} \\ &= \sum_{|\mu| \le \deg(q_i)} q_{i,\mu} \langle \psi, \nu^\dagger\ \mu\ \omega\ \psi \rangle, \end{aligned} \tag{6}$$

for any $|\nu|, |\omega| \le k_i$, where $k_i = k - \lceil \deg(q_i)/2 \rceil$, and $i = 1, \dots, m$.

If $(\mathcal{H}, X, \psi)$ is feasible, one can utilize the Sums of Squares theorem of [19] and [35] to derive semidefinite programming (SDP) relaxations. In particular, we can obtain a k^{th}-order SDP relaxation of the non-commutative polynomial optimization problem (2) by choosing a moment order k that satisfies the condition of $2k \ge \max\{\deg(p), \deg(q_i)\}$. In the Navascules-Pironio-Acin (NPA) hierarchy [37], the SDP relaxation of moment order k, has the following form:

$$\begin{aligned} P^k = \min_{y = (y_\omega)_{|\omega| \le 2k}} \quad & \sum_{|\omega| \le d} p_\omega y_\omega \\ \text{s.t.} \quad & M_k(y) \succcurlyeq 0, \\ & M_{k_i}(q_i y) \succcurlyeq 0, i = 1, \dots, m, \\ & y_1 = 1. \end{aligned} \tag{7}$$

Notice that there are variants [57, 56, 54] that exploit the sparsity and significantly reduce the computational burden.

Let us define the quadratic module, following [41]. Let $Q = \{q_i, i = 1, \dots, m\}$ be the set of polynomials determining the constraints. The *positivity domain* $\mathbf{S}_Q$ of Q are n-tuples of bounded operators $X = (X_1, \dots, X_n)$ on a Hilbert space $\mathcal{H}$ making all $q_i(X)$ positive semidefinite. The *quadratic module* $\mathbf{M}_Q$ is the set of $\sum_i f_i^\dagger f_i + \sum_i \sum_j g_{ij}^\dagger q_i g_{ij}$ where f_i and g_{ij} are polynomials from the same ring. As in [41], we assume:

Assumption 1 (Archimedean)**.** *Quadratic module* $\mathbf{M}_Q$ *of* (2) *is Archimedean, i.e., there exists a real constant* C *such that* $C^2 - (X_1^\dagger X_1 + \cdots + X_{2n}^\dagger X_{2n}) \in \mathbf{M}_Q$.

If the Archimedean assumption is satisfied, Pironio et al. [41] have shown that $\lim_{k \to \infty} P^k = P^*$ and how to use the so-called rank-loop condition [41] to detect global optimality. We refer to an extended version online [60] for further details.

C. Minimizer Extraction and Gelfand-Naimark-Segal Construction

Notice that the solution of the SDP relaxation makes it possible to read out the value of the objective function $\langle \psi, p(X)\psi \rangle$ of (2) easily, by looking up the correct entries of the moment matrix (5). To extract the optimizer with this objective-function value, one may utilize a variant of the singular-value decomposition of the moment matrix pioneered by [20], which can be construed [25] as the Gelfand–Naimark–Segal (GNS) construction [15, 45, 11]. (The GNS construction essentially produces a *-representation from a positive linear functional of a C*-algebra on a Hilbert space. Under the Archimedean assumption, this method could be applied to non-commutative polynomials, which are not C*-algebras otherwise. We refer to an extended version online [60] for further details.) These SVD-based approaches do not require the rank-loop condition to be satisfied, as is well explained in Section 2.2 of [25]. Once global optimality is detected (cf. the previous section), it is possible to extract the global optimum (H^*, X^*, ψ^*) from the solution of the SDP relaxation of (2) by Gram decomposition; cf. Theorem 2 in [41].

III. The Main Result

Given a trajectory of observations Y_1,...,Y_{t-1}, loss is a one-step error function at time t that compares an estimate f_t with the actual observation Y_t. Within the least squares estimator, we aim to minimize the sum of quadratic loss functions, i.e.,

$$\min_{f_t, t \ge 1} \sum_{t \ge 1} \|Y_t - f_t\|^2, \tag{8}$$

where the estimates $f_t, t \ge 1$ are decision variables. The properties of the optimal least squares estimate are well understood: it is consistent, cf. Mann and Wald [34] and Ljung [31], and has favorable sample complexity, cf. Theorem 4.2 of Campi and Weyer [9] in the general case, and to Jedra and Proutiere [22] for the latest result parameterized by the size of a certain epsilon net. We stress, however, that *it has not been understood* how to solve the non-convex optimization problem, in general, outside of some special cases [16] and recent, concurrent work of [4]. In contrast to [16], we focus on a method achieving global convergence under mild assumptions, and specifically without assuming the dimension of the hidden state is known.

When the dimension of the hidden state is not known, we need operator-valued variables m_t to model the state evolution, and some additional scalar-valued variables. We denote the process noise and the observation noise at time t by ω_t and ν_t, respectively. We also denote as such the decision variables corresponding to the estimates thereof, if there is no risk of confusion. If we add the sum of the squares of ω_t and the sum of the squares of ν_t as regularizers to the objective function with sufficiently large multipliers and minimize the resulting objective, we should reach a feasible solution with respect to the system matrices with the process noise ω_t and observation noise ν_t being close to zero.

Overall, such a formulation has the form in Equations (9) subject to (10–11). The inputs are $Y_t, t \geq 1$, i.e., the time series of the actual measurements, of a time window T thereof, and multipliers c_1, c_2. Decision variables are system matrices G, F; noisy estimates f_t, realizations ω_t, ν_t of noise, for $t \geq 1$; and state estimates m_t, for $t \geq 0$, which include the initial state m_0. We minimize the objective function:

$$\min_{f_t, m_t, G, F, \omega_t, \nu_t} \sum_{t\geq 1} \|Y_t - f_t\|^2 + c_1 \sum_{t\geq 1} \nu_t^2 + c_2 \sum_{t\geq 1} \omega_t^2 \tag{9}$$

for a 2-norm $\|\cdot\|$ over the feasible set given by constraints for $t \geq 1$:

$$m_t = G m_{t-1} + \omega_t \tag{10}$$

$$f_t = F' m_t + \nu_t. \tag{11}$$

We call the term $F'm_t$ noise-free estimates, which are regarded as our simulated/ predicted outputs. Equations (9) subject to (10–11) give us the least squares model. We can now apply the techniques of non-commutative polynomial optimization to the model so as to recover the system matrices of the underlying linear system.

Theorem 2. *For any observable linear system (G, F, V, W), for any length T of a time window, and any error $\epsilon > 0$, under Assumption 1, there is a convex optimization problem whose objective function value is at most ϵ away from* (9) *subject to (10–11). Furthermore, an estimate of (G, F, V, W) can be extracted from the solution of the same convex optimization problem.*

Proof. First, we need to show the existence of a sequence of convex optimization problems, whose objective function approaches the optimum of the non-commutative polynomial optimization problem. As explained in Section II-B above, [41] show that there is a sequence of natural semidefinite-programming relaxations of (2). The convergence of the sequence of their objective-function values is shown by Theorem 1 of [41], which requires Assumption 1. The translation of a problem involving multiple scalar- and operator-valued variables $f_t, m_t, G, F, \omega_t, \nu_t$ in (9–11) to $(\mathcal{H}, X, \psi)$ of (2), also known as the product-of-cones construction, is somewhat tedious, but routine and implemented in multiple software packages [59, 55, e.g.]. Second, we need to show that extraction of an estimate of (G, F, V, W) from the SDP relaxation of order $k(\epsilon)$ in the series is possible. There, one utilizes the Gelfand–Naimark–Segal (GNS) construction [15, 45], as explained in Section 2.2 of [25] or in Section II-C above. Notice that [29, cf.] the estimate of (G, F, V, W) may have a higher error than ϵ. □

This reasoning can be applied to more complicated formulations, involving shape constraints. For instance, in quantum systems [5], density operators are Hermitian and this constraint can be added to the least squares formulation.

Crucially for the practical applicability of the method, one should like to exploit the sparsity in the NCPOP (9–11). Notice that one can decompose the problem (9–11) into t subsets of variables involving $f_t, m_{t-1}, m_t, G, F, \omega_t, \nu_t$, which satisfy the so-called running intersection property [55]. We refer to [24] for a seminal paper on trace optimization exploiting correlative sparsity, and to [55] for the variant exploiting term sparsity. We also present a brief summary online [60].

Also, note that the extraction of the minimizer using the GNS procedure, as explained in Section II-C above, is stable to errors in the moment matrix, for *any* NCPOP, including the pre-processing above. See Theorem 4.1 in [25]. That is: it suffices to solve the SDP relaxation with a fixed error, in order to extract the minimizer.

One can also utilize a wide array of reduction techniques on the resulting SDP relaxations. Notable examples include facial reduction [7, 40] and exploiting sparsity [13]. Clearly, these can be applied to any SDPs, irrespective of the non-commutative nature of the original problem, but can also introduce [27] numerical issues. We refer to [33] for an up-to-date discussion.

IV. Numerical illustrations

Let us now present the implementation of the approach using the techniques of non-commutative polynomial optimization [41, 8] and to compare the results with traditional system identification methods. Our implementation is available online [1]. We present our experimental settings in more detail online [60].

A. The general setting

a) Our formulation and solvers: For our formulation, we use Equations (9) subject to (10–11), where we need to specify the values of c_1 and c_2. To generate the SDP relaxation of this formulation as in (7), we need to specify the moment order k. Because the degrees of objective (9) and constraints in (10–11) are all less than or equal to 2, the moment order k within the respective hierarchy can start from $k = 1$.

In our implementation, we use a globally convergent Navascués-Pironio-Acín (NPA) hierarchy [41] of SDP relaxations, as utilized in the proof of Theorem 2, and its sparsity-exploiting variant, known as the non-commutative variant of the term-sparsity exploiting moment/SOS (TSSOS) hierarchy [57, 56, 55]. (See [60] for a summary.) The SDP of a given moment order within the NPA hierarchy is constructed using `ncpol2sdpa` 1.12.2[2] of Wittek [59]. The SDP of a given moment order within the non-commutative variant of the TSSOS hirarchy is constructed using the `nctssos`[3] of Wang et al. [55]. Both SDP relaxations are then solved by `mosek` 9.2 [36].

b) Baselines: We compare our method against leading methods for estimating state-space models, as implemented in MathWorks™ Matlab™ System Identification Toolbox™. Specifically, we test against a combination of least squares algorithms implemented in routine `ssest` ("least squares auto"), subspace methods of [38] implemented in routine `n4sid` ("subspace auto"), and a subspace identification method of [21] with an ARX-based algorithm to compute the weighting, again utilized via `n4sid` ("ssarx").

To parameterize the three baselines, we need to specify the dimension d of the estimated state-space model. We would set $d = n$ directly or alternatively, iterate from 1 to the highest number allowed in the toolbox when the underlying system is unknown, e.g., in real-world stock-market data. Then, we need to specify the error to be minimized in the loss function during estimation. In fairness to the

[1] https://github.com/Quan-Zhou/Proper-Learning-of-LDS
[2] https://github.com/peterwittek/ncpol2sdpa
[3] https://github.com/wangjie212/NCTSSOS

baselines, we use the one-step ahead prediction error when comparing prediction performance and simulation error between measured and simulated outputs when comparing simulation performance.

c) The performance index: To measure the goodness of fit between the ground truth $\{Y_t\}_{t=1}^T$ (actual measurements) and the noise-free simulated/ predicted outputs $\{F'm_t\}_{t=1}^T$, using different system identification methods, we introduce the *normalized root mean squared error (nrmse)* fitness value:

$$\text{nrmse} := \left(1 - \frac{\|Y - F'm\|^2}{\|Y - \text{mean}(Y)\|^2}\right) \times 100\%, \tag{12}$$

where Y and $F'm$ are the vectors consisting of the sequence $\{Y_t\}_{t=1}^T$ and $\{F'm_t\}_{t=1}^T$ respectively. A higher nrmse fitness value indicates better simulation or prediction performance.

B. Experiments on the example of Hazan et al.

Experiments in Sections IV-B–IV-C utilize synthetic time series of T observations generated using LDS of the form in (1), with the tuple (G, F, V, W) and the initial hidden state ϕ_0 detailed next. We use the dimension n to indicate that the time series of observations were generated using $n \times n$ system matrices, while we use operator-valued variables to estimate these. The standard deviations of process noise and observation noise W, V are chosen from $0.1, 0.2, \ldots, 0.9$. Note that W is an $n \times n$ matrix in general, while we consider the spherical case of $W = 0.1 \times I_n$, where I_n is the n-dimensional identity matrix, which we denote by $W = 0.1$.

In our first experiment, we explore the statistical performance of feasible solutions of the SDP relaxation using the example of Hazan et al. [18, 26]. We performed one experiment on each combination of standard deviations of process W and observation noise V from the discrete set $0.1, 0.2, \ldots, 0.9$, i.e., 81 runs in total.

Figure 1 illustrates the nrmse values of the 81 runs of our method in different combinations of standard deviations of process noise W and observation noise V (upper), and another 81 experiments in different combinations of c_1 and c_2 (lower). In the upper subplot of Figure 1, we consider: $n = 2$, $G = \left(\begin{smallmatrix} 0.9 & 0.2 \\ 0.1 & 0.1 \end{smallmatrix}\right)$, $F' = \left(\begin{smallmatrix} 1 & 0.8 \end{smallmatrix}\right)$, the starting point $\phi_0' = \left(\begin{smallmatrix} 1 & 1 \end{smallmatrix}\right)$, and $T = 20$. In the lower subplot of Figure 1, we have the same settings as in the upper one, except for $W = V = 0.5$ and the parameters c_1, c_2 being chosen from $10^{-4}, \ldots, 1$. It seems clear the highest nrmse is to be observed for the standard deviation of both process and observation noises close to 0.5. While this may seem puzzling at first, notice that higher standard deviations of noise make it possible to approximate the observations by an auto-regressive process with low regression depth [26, Theorem 2]. The observed behavior is therefore in line with previous results [26, e.g., Figure 3].

C. Comparisons against the baselines

Next, we investigate the simulation performance of our method in comparison with other system identification methods, for varying LDS used to generate the time series. Our method and the three baselines described in Section IV-A are run 30 times for each choice of the standard deviations of the noise, with all methods using the same time series.

Figure 2 illustrates the results, with methods distinguished by colors: blue for "least squares auto", purple for "subspace auto", pink for "ssarx", and yellow for our method. The upper subplot presents the mean (solid lines) and mean ± one standard deviations (dashed lines) of nrmse as standard deviation of both process noise and observation noise ("noise std") increasing in lockstep from 0.1 to 0.9. The underlying system is the same as in the upper subplot of Figure 1, except for $W = V = 0.1, 0.2, \ldots, 0.9$. The middle subplot is similar, except the time series are generated by systems of a higher differential order:

$$\begin{aligned} \phi_t &= G\phi_{t-1} + \omega_t \\ Y_t &= F_1'\phi_t + F_2'(\phi_t - \phi_{t-1}) + \nu_t, \end{aligned} \tag{13}$$

and the formulation of our method is changed accordingly. In the lower subplot of Figure 2, we consider the mean (solid dots) and mean ± one standard deviations (vertical error bars) of nrmse at different dimensions $n = 2, 3, 4$ of the underlying system (1).

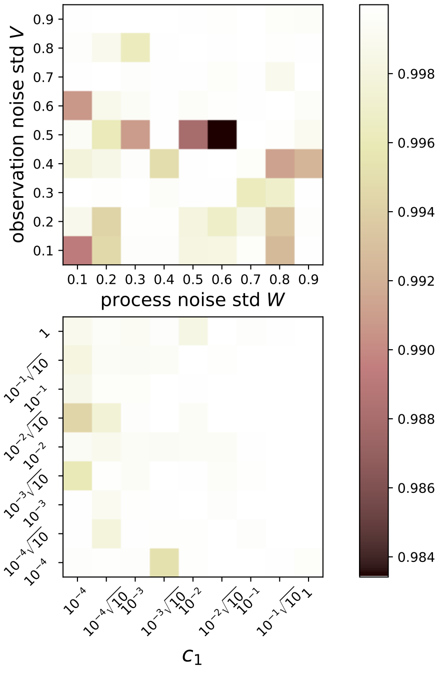

Fig. 1: **Upper:** The nrmse fitness values (12) of 81 experiments of our method at different combinations of noise standard deviations of process noise W and observation noise V and **Lower:** at different combinations of parameters c_1 and c_2. Both use the data generated from systems in (1). Lighter colors indicate higher nrmse and thus better simulation performance.

As Figure 2 suggests, the nrmse values of our method on this example are almost 100%, while other methods rarely reach 50% despite the fact that the dimensions used by the baselines are the true dimensions of the underlying system ($d = n$). (We will use "least squares auto", which seems to work best within the other methods, in the following experiment on stock-market data.) Additionally, our method shows better stability; the gap between the yellow dashed lines in the upper or middle subplot, which suggests the width of two standard deviations, is relatively small.

D. Experiments with stock-market data

Our approach to proper learning of LDS could also be used in a "prediction-error" method for improper learning of LDS, i.e., forecasting its next observation (output, measurement). As such, it can be applied to any time series. To exhibit this, we consider real-world stock-market data first used in [30]. In particular, we predict the evolution of the stock price from the 21st period to the 121st period, where each prediction is based on the 20 immediately preceding observations ($T = 20$). For our method, we use the same formulation (9) subject to (10)-(11), but with the variable F' removed. For comparison, the combination of least squares algorithms "least

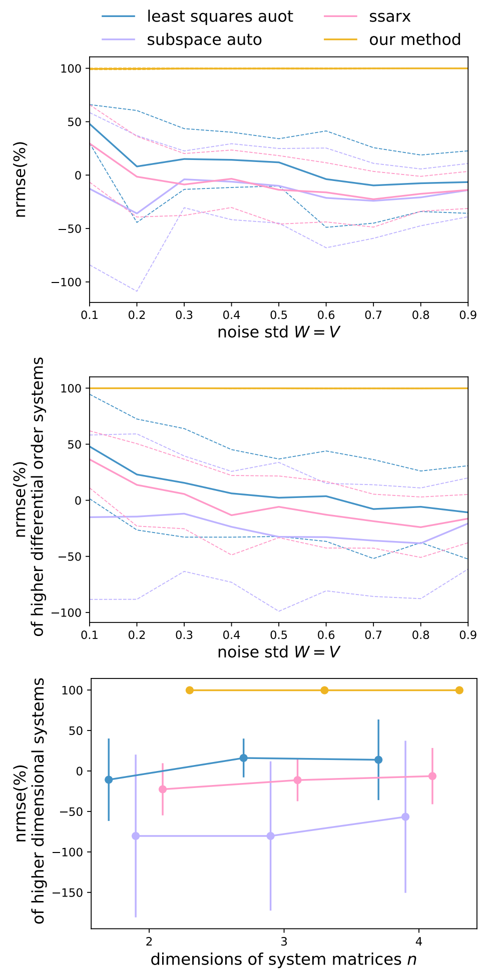

Fig. 2: The nrmse fitness values (12) of our method compared to the leading system identification methods implemented in Matlab™ System Identification Toolbox™. **Upper & middle:** the mean (solid lines) and mean ± one standard deviations (dashed lines) of nrmse as standard deviation of both process noise and observation noise increasing in lockstep from 0.1 to 0.9. The time series used for simulation are generated from systems in (1) (upper) and higher differential order systems in (13) (middle), with the dimensions n of both systems being 2. **Lower:** the mean (solid dots) and mean ± one standard deviations (vertical error bars) of nrmse at different dimensions n of the underlying systems in (1). Higher nrmse indicates better simulation performance.

squares auto" is used again. Since we are using the stock-market data, the dimension n of the underlying system is unknown. Hence, the dimensions d of the "least squares auto" are iterated from 1 to 4, wherein 4 is the highest setting allowed in the toolbox for 20-period observations.

Figure 3 shows in the left subplot the results obtained by our method (a yellow curve), and the "least squares auto" of varying dimensions $d = 1, 2, 3, 4$ (four blue curves). The true stock price "origin" is displayed by a dark curve. The percentages in the legend correspond to nrmse values (12). Both from the nrmse and the shape of these curves, we notice that "least squares auto" performs poorly when the stock prices are volatile. This is highlighted in the right subplot, which zooms in on the 66th-101st period.

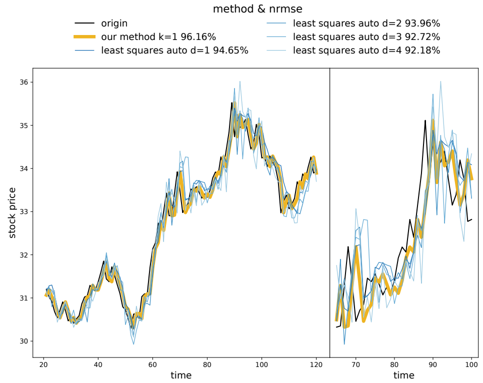

Fig. 3: **Left:** The time series of stock price (dark) for the 21st-121st period used in [30], and the predicted outputs of our method (yellow) compared against "least squares auto" (blue) implemented in Matlab™ System Identification Toolbox™. The dimension d of "least squares auto" is iterated from 1 to the highest number of 4. The percentages in legend are corresponding nrmse values of one-step predictions. **Right:** a zoom-in for the 66th-101st period.

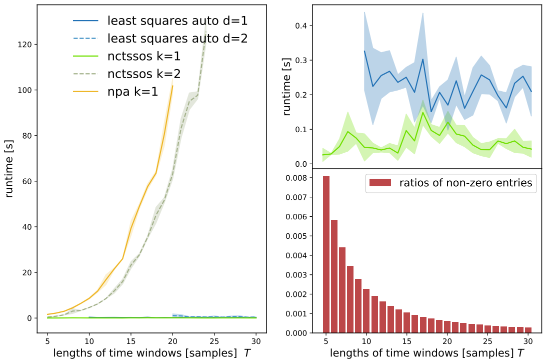

Fig. 4: **Left:** The (solid or dashed) curves show the mean runtime of the SDP relaxation of the baseline "least squares auto" (blue), the TSSOS hierarchy (green) and the NPA hierarchy (yellow), at different moment orders k or dimensions d. The mean ± one standard deviation of runtime is displayed by shaded error bands. **Upper-right:** The mean and mean ± one standard deviation of runtime of the SDP relaxation of TSSOS hierarchy at moment order $k = 1$ and the "least squares auto" with dimension $d = 1$. **Lower-right:** The red bars display the sparsity of NPA hierarchy of the experiment on stock-market data against the length of time window, by ratios of non-zero coefficients out of all coefficients in the SDP relaxations

E. Runtime

Next, we consider the runtime of two implementations of solvers for (9) subject to (10)-(11). The first implementation constructs the SDP relaxation of NPA hierarchy via `ncpol2sdpa` 1.12.2 with moment order $k = 1$. The second implementation constructs the non-commutative variant of the TSSOS hierarchy via `nctssos`, with moment order $k = 1, 2$. For comparison purposes, we include the baseline "least squares auto" at dimensions $d = 1, 2$. We randomly select a time series from the stock-market data, with the length of time window T chosen from $5, 6, \ldots, 30$, and run these three methods three times for each T.

Figure 4 illustrates the runtime of the SDP relaxations and the baseline "least squares auto" as a function of the length of the time window. These implemented methods are distinguished by colors: blue for "least squares auto", green for the non-commutative variant of the TSSOS hierarchy ("nctssos"), and yellow for the NPA hierarchy ("npa"). The mean and mean ± one standard deviation of runtime are displayed by (solid or dashed) curves and shaded error bands. The upper-right subplot compares the runtime of our method with "nctssos" at moment order $k = 1$ against "least squares auto" with dimension $d = 1$. The red bars in the lower-right subplot display the sparsity of NPA hierarchy of the experiment on stock-market data against the length of time window, by ratios of non-zero coefficients out of all coefficients in the SDP relaxations.

As in most primal-dual interior-point methods [53], runtime of solving the relaxation to ϵ error is polynomial in its dimension and logarithmic in $1/\epsilon$, but it should be noted that the dimension of the relaxation grows fast in the length T of the time window and the moment order k. It is clear that the runtime of solvers for SDP relaxations within the non-commutative variant of the TSSOS hierarchy exhibits a modest growth with the length of time window, much slower than that of the plain-vanilla NPA hierarchy.

V. Conclusions

We have presented an alternative approach to the recovery of hidden dynamic underlying a time series, without assumptions on the dimension of the hidden state. For the first time in system identification and machine learning, this approach utilizes non-commutative polynomial programming (NCPOP), which has been recently developed within Mathematical Optimization [41, 59, 25, 55]. NCPOP can accommodate a variety of other objectives and constraints [61, e.g. in fairness]. This builds upon a long history of work on the method of moments [1, 19] and its applications in Machine Learning [28], as well as recent progress [33] in the scalability of semidefinite programming.

Acknowledgements

We would like to thank two anonymous reviewers, Denys Bondar, and Rostislav Horcik for their kind comments, which have helped us improve the draft substantially. J.M. acknowledges support of the OP RDE funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" and the Czech Science Foundation (23-07947S). This work has received funding from the European Unions Horizon Europe research and innovation programme under grant agreement No. 101084642. This work was also supported by Innovate UK under the Horizon Europe Guarantee; UKRI Reference Number: 101084642 (Codiet).

References

[1] Naum Ilich Akhiezer and M.G. Krein. *Some questions in the theory of moments*, volume 2. American Mathematical Society, 1962.

[2] Oren Anava, Elad Hazan, Shie Mannor, and Ohad Shamir. Online learning for time series prediction. In *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, 2013.

[3] Roy M Anderson and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992.

[4] Ainesh Bakshi, Allen Liu, Ankur Moitra, and Morris Yau. A new approach to learning linear dynamical systems. *arXiv preprint arXiv:2301.09519*, 2023.

[5] Denys Bondar, Kurt Jacobs, Georgios Korpas, Jakub Marecek, Zakhar Popovych, and Jiri Vala. A globally convergent approach for quantum control and system identification. *Bulletin of the American Physical Society*, 2023.

[6] Denys I Bondar, Zakhar Popovych, Kurt Jacobs, Georgios Korpas, and Jakub Marecek. Recovering models of open quantum systems from data via polynomial optimization: Towards globally convergent quantum system identification. *arXiv preprint arXiv:2203.17164*, 2022.

[7] Jon M Borwein and Henry Wolkowicz. Facial reduction for a cone-convex programming problem. *Journal of the Australian Mathematical Society*, 30(3):369–380, 1981.

[8] Sabine Burgdorf, Igor Klep, and Janez Povh. *Optimization of polynomials in non-commuting variables*. Springer, 2016.

[9] Marco C Campi and Erik Weyer. Finite sample properties of system identification methods. *IEEE Transactions on Automatic Control*, 47(8):1329–1334, 2002.

[10] I-Chun Chou and Eberhard O Voit. Recent developments in parameter estimation and structure identification of biochemical and genomic systems. *Mathematical biosciences*, 219(2):57–83, 2009.

[11] Jacques Dixmier. *Les C*-algèbres et leurs représentations*. Gauthier-Villars, Paris, France, 1969. English translation: C*-algebras (North-Holland, 1982).

[12] Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Finite time identification in unstable linear systems. *Automatica*, 96:342–353, 2018.

[13] Mituhiro Fukuda, Masakazu Kojima, Kazuo Murota, and Kazuhide Nakata. Exploiting sparsity in semidefinite programming via matrix completion i: General framework. *SIAM Journal on Optimization*, 11(3):647–674, 2001.

[14] Philipp Geiger, Kun Zhang, Bernhard Schoelkopf, Mingming Gong, and Dominik Janzing. Causal inference by identification of vector autoregressive processes with hidden components. In *International Conference on Machine Learning*, pages 1917–1925, 2015.

[15] Israel Gelfand and Mark Neumark. On the imbedding of normed rings into the ring of operators in Hilbert space. *Rec. Math. [Mat. Sbornik] N.S.*, 12(2):197–217, 1943.

[16] Moritz Hardt, Tengyu Ma, and Benjamin Recht. Gradient descent learns linear dynamical systems. *Journal of Machine Learning Research*, 19:1–44, 2018.

[17] Elad Hazan, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. Spectral filtering for general linear dynamical systems. In *Advances in Neural Information Processing Systems*, pages 4634–4643, 2018.

[18] Elad Hazan, Karan Singh, and Cyril Zhang. Learning linear dynamical systems via spectral filtering. In *Advances in Neural Information Processing Systems*, pages 6702–6712, 2017.

[19] J William Helton. "Positive" noncommutative polynomials are sums of squares. *Annals of Mathematics*, 156(2):675–694, 2002.

[20] Didier Henrion and Jean-Bernard Lasserre. Detecting global optimality and extracting solutions in gloptipoly. In *Positive polynomials in control*, pages 293–310. Springer, 2005.

[21] Magnus Jansson. Subspace identification and ARX modeling. In *Proceedings of the 13th IFAC SYSID Symposium*, pages 1625–1630, 2003.

[22] Yassir Jedra and Alexandre Proutiere. Finite-time identification of stable linear systems optimality of the least-squares estimator. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 996–1001. IEEE, 2020.

[23] Tohru Katayama. *Subspace methods for system identification*. Springer Science & Business Media, 2006.

[24] Igor Klep, Victor Magron, and Janez Povh. Sparse noncommutative polynomial optimization. *Mathematical Programming*, 193(2):789–829, 2022.

[25] Igor Klep, Janez Povh, and Jurij Volcic. Minimizer extraction in polynomial optimization is robust. *SIAM Journal on Optimization*, 28(4):3177–3207, 2018.

[26] Mark Kozdoba, Jakub Marecek, Tigran Tchrakian, and Shie Mannor. On-line learning of linear dynamical systems: Exponential forgetting in Kalman filters. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, pages 4098–4105, 2019. arXiv preprint arXiv:1809.05870.

[27] Vyacheslav Kungurtsev and Jakub Marecek. A two-step preprocessing for semidefinite programming. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 384–389. IEEE, 2020.

[28] Jean-Bernard Lasserre and Victor Magron. Optimal data fitting: A moment approach. *SIAM Journal on Optimization*, 28(4):3127–3144, 2018.

[29] Yunseok Lee, Holger Boche, and Gitta Kutyniok. Computability of optimizers. *arXiv preprint arXiv:2301.06148*, 2023.

[30] Chenghao Liu, Steven CH Hoi, Peilin Zhao, and Jianling Sun. Online arima algorithms for time series prediction. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

[31] Lennart Ljung. Consistency of the least-squares identification method. *IEEE Transactions on Automatic Control*, 21(5):779–781, 1976.

[32] Lennart Ljung. *System Identification: Theory for the User*. Pearson Education, 1998.

[33] Anirudha Majumdar, Georgina Hall, and Amir Ali Ahmadi. Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 3, 2019.

[34] Henry B Mann and Abraham Wald. On the statistical treatment of linear stochastic difference equations. *Econometrica, Journal of the Econometric Society*, pages 173–220, 1943.

[35] Scott McCullough. Factorization of operator-valued polynomials in several non-commuting variables. *Linear Algebra and its Applications*, 326(1-3):193–203, 2001.

[36] MOSEK, ApS. The MOSEK Optimizer API for Python 9.2. 2020.

[37] Miguel Navascués, Stefano Pironio, and Antonio Acín. SDP relaxations for non-commutative polynomial optimization. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 601–634. Springer, 2012.

[38] Peter Van Overschee and Bart De Moor. *Subspace identification for linear systems: TheoryImplementationApplications*. Springer Science & Business Media, 2012.

[39] Samet Oymak and Necmiye Ozay. Non-asymptotic identification of LTI systems from a single trajectory. In *2019 American Control Conference (ACC)*, pages 5655–5661. IEEE, 2019.

[40] Frank Permenter and Pablo Parrilo. Partial facial reduction: simplified, equivalent sdps via approximations of the psd cone. *Mathematical Programming*, 171(1-2):1–54, 2018.

[41] S. Pironio, M. Navascués, and A. Acín. Convergent relaxations of polynomial optimization problems with noncommuting variables. *SIAM Journal on Optimization*, 20(5):2157–2180, 2010.

[42] Tuhin Sarkar and Alexander Rakhlin. Near optimal finite time identification of arbitrary linear dynamical systems. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5610–5618, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[43] Tuhin Sarkar, Alexander Rakhlin, and Munther A. Dahleh. Finite time LTI system identification. *J. Mach. Learn. Res.*, 22(1), jul 2022.

[44] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[45] Irving E Segal. Irreducible representations of operator algebras. *Bulletin of the American Mathematical Society*, 53(2):73–88, 1947.

[46] Max Simchowitz, Ross Boczar, and Benjamin Recht. Learning linear dynamical systems with semi-parametric least squares. In *Conference on Learning Theory*, pages 2714–2802. PMLR, 2019.

[47] Max Simchowitz, Horia Mania, Stephen Tu, Michael I Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *Conference On Learning Theory*, pages 439–473, 2018.

[48] Yue Sun, Samet Oymak, and Maryam Fazel. Finite sample system identification: Optimal rates and the role of regularization. In *Learning for Dynamics and Control*, pages 16–25. PMLR, 2020.

[49] Arun K Tangirala. *Principles of system identification: theory and practice*. Crc Press, 2014.

[50] Anastasios Tsiamis, Nikolai Matni, and George Pappas. Sample complexity of Kalman filtering for unknown systems. In *Learning for Dynamics and Control*, pages 435–444. PMLR, 2020.

[51] Anastasios Tsiamis and George J Pappas. Finite sample analysis of stochastic system identification. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3648–3654. IEEE, 2019.

[52] Anastasios Tsiamis and George J. Pappas. Online learning of the Kalman filter with logarithmic regret. *IEEE Transactions on Automatic Control*, pages 1–16, 2022.

[53] Levent Tunçel. Potential reduction and primal-dual methods. In *Handbook of semidefinite programming*, pages 235–265. Springer, 2000.

[54] Jie Wang, Martina Maggio, and Victor Magron. Sparsejsr: A fast algorithm to compute joint spectral radius via sparse sos decompositions. In *2021 American Control Conference (ACC)*, pages 2254–2259. IEEE, 2021.

[55] Jie Wang and Victor Magron. Exploiting term sparsity in noncommutative polynomial optimization. *Computational Optimization and Applications*, 80:483–521, 2021.

[56] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. Chordal-tssos: a moment-sos hierarchy that exploits term sparsity with chordal extension. *SIAM Journal on Optimization*, 31(1):114–141, 2021.

[57] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. Tssos: A moment-sos hierarchy that exploits term sparsity. *SIAM Journal on Optimization*, 31(1):30–58, 2021.

[58] Mike West and Jeff Harrison. *Bayesian Forecasting and Dynamic Models (2nd ed.)*. Springer-Verlag, Berlin, Heidelberg, 1997.

[59] Peter Wittek. Algorithm 950: Ncpol2sdpasparse semidefinite programming relaxations for polynomial optimization problems of noncommuting variables. *ACM Transactions on Mathematical Software (TOMS)*, 41(3):1–12, 2015.

[60] Quan Zhou and Jakub Marecek. Proper learning of linear dynamical systems as a non-commutative polynomial optimisation problem. *arXiv preprint arXiv:2002.01444*, 2023. Version 4.

[61] Quan Zhou, Jakub Mareček, and Robert Shorten. Fairness in forecasting of observations of linear dynamical systems. *Journal of Artificial Intelligence Research*, 76:1247–1280, 2023.