Online Subspace Learning on Flag Manifolds for System Identification

Dian Jin DJIN38@WISC.EDU
University of Wisconsin-Madison

Jeremy Coulson JEREMY.COULSON@WISC.EDU
University of Wisconsin-Madison

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

Data-driven control methods based on subspace representations are powerful but are often limited to linear time-invariant systems where the model order is known. A key challenge is developing online data-driven control algorithms for time-varying systems, especially when the system's complexity is unknown or changes over time. To address this, we propose a novel online subspace learning framework that operates on flag manifolds. Our algorithm leverages streaming data to recursively track an ensemble of nested subspaces, allowing it to adapt to varying system dimensions without prior knowledge of the true model order. We show that our algorithm is a generalization of the Grassmannian Recursive Algorithm for Tracking. The learned subspace models are then integrated into a data-driven simulation framework to perform prediction for unknown dynamical systems. The effectiveness of this approach is demonstrated through a case study where the proposed adaptive predictor successfully handles abrupt changes in system dynamics and consistently achieves competitive performance against several baselines.

Keywords: System identification; Subspace learning; Optimization on manifolds

1. Introduction

Subspace representations are becoming increasingly important in the field of systems and control, especially with the recent revival of behavioral systems theory (Willems, 1986) in data-driven control. A key concept in this area is Willems' Fundamental Lemma, which provides a direct data-driven non-parametric *subspace* representation for finite-length trajectories of linear time-invariant systems. This has spurred the development of effective subspace identification techniques (Favoreel et al., 1999; Van Overschee and De Moor, 2012), novel data-driven simulation and control methods (Markovsky and Rapisarda, 2008; Coulson et al., 2021; De Persis and Tesi, 2019; Berberich et al., 2020; Verhoek et al., 2021; Dörfler et al., 2022), which have been shown to perform remarkably well across a wide range of application domains (Elokda et al., 2021; Huang et al., 2021b,a; Kerkhof and Keviczky, 2021; Fawcett et al., 2022). See (Markovsky and Dörfler, 2021) and references therein.

Despite these advances, many existing identification and control methods are limited to Linear Time-Invariant (LTI) systems. A significant open challenge is the design of online adaptive approaches and recursive algorithms that can handle systems that change over time. While some subspace tracking methods have been developed for online state-space identification (Verhaegen and Verdult, 2007; Van Overschee and De Moor, 1994; Oku and Kimura, 2002; Zhang et al., 2019; Sasfi et al., 2025), they typically require the system order to be known and constant. In many real-world scenarios, however, the true system order is unknown and may vary, for instance, due to component failures. In this work, we propose a novel framework for adaptive, data-driven identification of unknown, time-varying dynamical systems based on subspace learning techniques.

Subspace learning (sometimes termed subspace tracking) has long been studied in signal processing (Yang, 1995; Delmas, 2010; Balzano et al., 2010; Balzano and Wright, 2015; He et al., 2012). Broadly speaking, subspace learning algorithms can be classified into algebraic and geometric methods (Balzano et al., 2018). The geometric methods optimize a certain loss function via gradient descent on a matrix manifold such as the Grassmannian, a set consisting of all subspaces of a particular dimension. Such methods have a solid theoretical background rooted in optimization on manifolds (Edelman et al., 1998; Absil et al., 2004; Boumal, 2023). Running these algorithms on Grassmannians requires choosing the dimensionality *a priori*. This poses a significant challenge in practical applications where the true underlying dimension is unknown or may change over time.

Contribution. We propose a novel framework for online system identification and data-driven simulation for unknown time-varying dynamical systems. The proposed framework consists of the Flag Recursive Online Tracking (FRONT) algorithm on flag manifolds that leverages streaming data and yields a nested hierarchy of subspaces. To the best of our knowledge, this is the first online algorithm that optimizes over flag manifolds with streaming data. We formally show that Grassmannian Recursive Algorithm for Tracking (GREAT) (Sasfi et al., 2025) is a special case of our algorithm when the true subspace dimension is known. When the dimension is unknown, we propose a dimensionality ensemble strategy inspired by ensemble learning (Dietterich, 2000). Finally, the FRONT algorithm is leveraged as a subspace model in data-driven simulation (Markovsky and Rapisarda, 2008) to perform trajectory prediction for a time and complexity-varying unknown system. The framework is showcased on an AutoRegressive model with eXogenous inputs (ARX) system simulation and is shown to outperform non-adaptive subspace prediction (Favoreel et al., 1999) and other adaptive benchmarks.

The rest of the paper is organized as follows. Section 2 contains preliminaries on subspace geometry. We formulate the problem in Section 3. Section 4 introduces the online subspace learning algorithm. Section 5 contains numerical studies. Section 6 concludes the paper.

2. Preliminaries

We use $\mathbb{Z}_{\geq 0}$ to denote non-negative integers. Given $i, j \in \mathbb{Z}_{\geq 0}$, with $i \leq j$, we use $[i, j]$ to denote the discrete interval $[i, j] \cap \mathbb{Z}_{\geq 0}$. We use $I_{p\times m}$ to denote the $p \times m$ matrix whose entries are 1 on the main diagonal (i.e., (i, i)-entry for $i = 1, \ldots, \min(p, m)$), and all other entries are 0. The diagonal matrix with diagonal elements $\gamma_1, \cdots, \gamma_d$ is denoted $\operatorname{diag}(\gamma_1, \cdots, \gamma_d)$. Given a function $v\colon \mathbb{Z}_{\geq 0} \to \mathbb{R}^q$ and $i, j \in \mathbb{Z}_{\geq 0}$, with $i \leq j$, we define $v_{[i,j]} = (v(i), v(i+1), \ldots, v(j)) \in \mathbb{R}^{q(j-i+1)}$. We use $\mathbf{U}$ to denote a subspace of $\mathbb{R}^p$ and U to denote a matrix such that $\operatorname{im} U = \mathbf{U}$. The orthogonal projector onto $\mathbf{U}$ is denoted by $\Pi_{\mathbf{U}}$, and is used interchangeably with $\Pi_U = UU^\top$, where U is an orthonormal matrix whose columns span $\mathbf{U}$. The Moore–Penrose inverse of a matrix A is denoted $A^\dagger$. The Gaussian distribution with mean 0 and covariance $\Sigma \in \mathbb{R}^{p\times p}$ is denoted $\mathcal{N}(0, \Sigma)$.

2.1. Subspace geometry

A Grassmannian is the set of all subspaces in $\mathbb{R}^p$ with dimension d denoted by $\operatorname{Gr}(p, d) := \{\mathbf{U} \text{ a linear subspace of } \mathbb{R}^p : \dim \mathbf{U} = d\}$. Flag manifolds generalize the Grassmannians by considering sequences of nested subspaces of increasing dimension. Let $p \geq 2$ be an integer and $q_{1:d} := (q_1, \ldots, q_d)$ with $0 < q_1 < \cdots < q_d < p$. A flag of signature $(p, q_{1:d})$ is a nested sequence of linear subspaces $\{\mathbf{U}_i\}_{i=1}^d$ of $\mathbb{R}^p$ satisfying $\{0\} \subset \mathbf{U}_1 \subset \cdots \subset \mathbf{U}_d \subset \mathbb{R}^p$ with $\dim \mathbf{U}_j = q_j, j = 1, \ldots, d$.

We denote this flag by $\mathbf{U}_{1:d}$. The set of all such flags forms a smooth manifold $\mathrm{Flag}(p, q_{1:d})$ (Ye et al., 2022). We also denote $q_0 = 0, q = q_d, q_{d+1} = p$. We can represent flags as points on the Stiefel manifold $\mathrm{St}(p, q) = \{U \in \mathbb{R}^{p\times q} : U^\top U = I\}$ consisting of orthonormal matrices. Define for each $k = 1, \ldots, d$, $U_k \in \mathrm{St}(p, q_k - q_{k-1})$ such that $[U_1 \cdots U_k]$ is an orthonormal basis of $\mathbf{U}_k$. We must note that under these notations, $\mathbf{U}_k$ is not the subspace spanned by U_k. Then $U_{1:d} \coloneqq [U_1 \cdots U_d]$ is a representative of $\mathbf{U}_{1:d}$, called Stiefel representative. Throughout this paper we will abuse the notation to write $U_{1:d} \in \mathrm{Flag}(p, q_{1:d})$. If the signature is (p, q_1), then $\mathrm{Flag}(p, (q_1)) = \mathrm{Gr}(p, q_1)$. The chordal distance of two subspaces $\mathbf{A} \in \mathrm{Flag}(p, (m)), \mathbf{B} \in \mathrm{Flag}(p, (n))$ is defined by $\mathrm{d}(\mathbf{A}, \mathbf{B}) = (\sum_{i=1}^{\min(m,n)} \sin^2 \theta_i)^{1/2}$ (Ye and Lim, 2016, Theorem 12), where θ_i is the i^{th} principal angle (Golub and Van Loan, 2013) between $\mathbf{A}$ and $\mathbf{B}$. We present a simple example to illustrate the difference between flags and subspaces.

Example 1 *Consider the canonical basis* $e_1 = (1, 0, 0, 0), e_2 = (0, 1, 0, 0), e_3 = (0, 0, 1, 0), e_4 = (0, 0, 0, 1)$ *of* $\mathbb{R}^4$. *Let* $\mathbf{U}_1 = \mathrm{span}\{e_1\}, \mathbf{U}_2 = \mathrm{span}\{e_1, e_2\}, \mathbf{U}_3 = \mathrm{span}\{e_1, e_2, e_3\}$, *then* $\mathbf{U}_{1:3} \in \mathrm{Flag}(4, (1, 2, 3))$, *and a Stiefel representation of this flag is the matrix* $U = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}$. *Note that even if two bases span the same subspace, the hierarchical structures of their flags may be different. Let* $V = \begin{bmatrix} e_1 & e_3 & e_2 \end{bmatrix}$, *then clearly* $\mathrm{im}\, V = \mathrm{im}\, U$. *However,* $\mathbf{V}_1 = \mathrm{span}\{e_1\}, \mathbf{V}_2 = \mathrm{span}\{e_1, e_3\}, \mathbf{V}_3 = \mathrm{span}\{e_1, e_2, e_3\}$, *which implies* $\mathbf{V}_2 \neq \mathbf{U}_2$, *hence* $\mathbf{V}_{1:d} \neq \mathbf{U}_{1:d}$.

3. Problem Setup & Motivation

The goal of this paper is to estimate an unknown, time-varying subspace $\mathbf{U}^t$ from streaming data. More precisely, for each $t \in \mathbb{Z}_{\geq 0}$, let $\mathbf{U}^t \in \bigcup_{j=1}^{d} \mathrm{Gr}(p, q_j)$, where $0 < q_1 < \cdots < q_d < p$. We assume that we have access to an upper bound q_d on the dimension of all subspaces $\mathbf{U}^t$. Note that the upper bound q_d may not be tight in the sense that there may be no time instance $t \in \mathbb{Z}_{\geq 0}$ where $\dim \mathbf{U}^t = q_d$. We assume that at each $t \in \mathbb{Z}_{\geq 0}$, we measure the data $w_t = v_t + \eta_t$, where $v_t \in \mathbf{U}^t$, and η_t is noise. We wish to obtain estimates $\widehat{\mathbf{U}}^t$ such that $d(\widehat{\mathbf{U}}^t, \mathbf{U}^t) \leq \epsilon_{\text{tol}}$ for some user-defined tolerance $\epsilon_{\text{tol}} > 0$, i.e., we seek estimates that are "close" to the true subspaces. We refer to this problem as *learning* $\mathbf{U}^t$ or *tracking* $\mathbf{U}^t$. In the case when $d = 1$ and q_d is known, there are efficient algorithms (Yang, 1995; Balzano et al., 2010; Sasfi et al., 2025) for robustly learning $\mathbf{U}^t$. However, in many practical settings the true dimension of the underlying subspaces is *unknown* and may even vary over time (e.g., online dynamic mode decomposition for time-varying dynamical systems (Zhang et al., 2019), direction of arrival analysis (Rabideau, 1996)). Solving this problem is critical for online system identification and control: a fixed, mis-specified subspace dimension yields biased estimates, slow adaptation, and brittle decisions (Ljung, 1998) (see Figure 3). Learning a nested hierarchy of subspaces (flag) online gives rank-on-demand models that track changing complexity, enabling efficient prediction, reliable change-point detection, without prior knowledge of the true order (Szwagier and Pennec, 2025).

Motivating Example: Online System Identification. Consider the Linear Time-Varying (LTV) dynamical system

$$\begin{aligned} x(t+1) &= A_t x(t) + B_t u(t), \\ y(t) &= C_t x(t) + D_t u(t), \end{aligned} \tag{1}$$

where $A_t \in \mathbb{R}^{n\times n}$, $B_t \in \mathbb{R}^{n\times m}$, $C_t \in \mathbb{R}^{r\times n}$, and $D_t \in \mathbb{R}^{r\times m}$ are time varying matrices. Assume that the model parameters (A_t, B_t, C_t, D_t) are unknown, and we only have access to input-output

data of (1). A typical problem in system identification is to use input-output data to find a model that captures the input-output behavior of (1). The main challenge in this setting is that the model parameters are *time-varying*. Furthermore, we do not have access to state measurements. Hence, the underlying state-space dimension (system complexity) is unknown. Let $L \geq 1$, for each $t \geq L-1$, the L-length input-output trajectories of (1) on the interval $[t-L+1, t]$ are given by

$$v_t := \begin{bmatrix} u_{[t-L+1,t]} \\ y_{[t-L+1,t]} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I_{mL} \\ \mathcal{O}_{[t-L+1,t]} & \mathcal{T}_{[t-L+1,t]} \end{bmatrix}}_{=:U^t} \begin{bmatrix} x(t-L+1) \\ u_{[t-L+1,t]} \end{bmatrix}, \text{ where}$$

$$\mathcal{O}_{[t-L+1,t]} = \begin{bmatrix} C_{t-L+1} \\ C_{t-L+2}A_{t-L+1} \\ \vdots \\ C_tA_{t-1}\cdots A_{t-L+1} \end{bmatrix}, \mathcal{T}_{[t-L+1,t]} = \begin{bmatrix} D_{t-L+1} & 0 & 0 & \cdots & 0 \\ C_{t-L+2}B_{t-L+1} & D_{t-L+2} & 0 & \cdots & 0 \\ C_{t-L+3}A_{t-L+2}B_{t-L+1} & C_{t-L+3}B_{t-L+2} & D_{t-L+3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_tA_{t-1}\cdots A_{t-L+2}B_{t-L+1} & \cdots & \cdots & \cdots & D_t \end{bmatrix}.$$

Instead of learning the state-space matrices (A_t, B_t, C_t, D_t) through (recursive) state-space system identification methods (Verhaegen and Dewilde, 1992; Van Overschee and De Moor, 1994; Oku and Kimura, 2002), we directly learn the subspaces $\mathbf{U}^t \in \bigcup_{j=1}^{d} \mathrm{Gr}(p, q_j)$ based on streaming data $w_t = v_t + \eta_t$, where η_t is measurement noise. Note that $\dim \mathbf{U}^t$ is unknown and may vary depending on the rank of $\mathcal{O}_{[t-L+1,t]}$. The subspace $\mathbf{U}^t$ is known in behavioral system theory (Willems, 1986; Markovsky and Dörfler, 2021) as the *restricted behavior* (Markovsky and Dörfler, 2022) and serves as the backbone of several powerful nonparametric control methods (Coulson et al., 2019; De Persis and Tesi, 2019; Markovsky and Dörfler, 2021; Markovsky and Rapisarda, 2008). This online learning problem is studied in (Sasfi et al., 2025) under the assumption that $\dim \mathbf{U}^t$ is constant, which may not hold in practical settings in which the complexity of the system varies over time (see Section 5.1). In what follows, we propose a method for learning $\mathbf{U}^t$ without assuming $\dim \mathbf{U}^t$ is constant, and use the learned flags for a data-driven prediction task (see Section 5.2). While we do not address control here, the learned subspaces can be incorporated into data-driven control frameworks, which is a direction for future work.

4. Online subspace learning on flag manifolds

Inspired by recent advances in subspace learning for online system identification (Sasfi et al., 2025), we propose a method for solving the learning problem in Section 3, which consists of estimating the evolving subspace $\mathbf{U}^t$ using optimization on the flag manifold. Each element of the flag manifold is an ensemble of subspaces that can be used for downstream tasks such as data-driven prediction. Thus, our algorithm contributes to a data-to-prediction pipeline that continuously adapts to streaming data and the changing complexity of the system (see Section 5.2).

4.1. Gradient descent on flag manifolds with streaming data

Let $\{\mathbf{U}^t\}_{t\in\mathbb{Z}_{\geq 0}}$ be a sequence of subspaces with each $\mathbf{U}^t \in \bigcup_{j=1}^{d} \mathrm{Gr}(p, q_j)$. At each time $t \in \mathbb{Z}_{\geq 0}$ we observe a data sample $w_t = v_t + \eta_t$, where $v_t \in \mathbf{U}^t$ and η_t is noise. Let $T \geq 1$. For each $t \geq T-1$, let $W_t = [w_{t-T+1} \ \cdots \ w_t] \in \mathbb{R}^{p\times T}$ be the window of the most recent T data samples. For each $t \in \mathbb{Z}_{\geq 0}$ and $j \in \{1, \ldots, d\}$, we define the cost function $g^j_{W_t} \colon \mathrm{Gr}(p, q_j) \to \mathbb{R}$ by

$$g^j_{W_t}(\mathbf{U}) = \|W_t - \Pi_{\mathbf{U}} W_t\|_F^2, \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm. Minimizing (2) over $\mathbf{U} \in \mathrm{Gr}(p, q_j)$ is known as principal component analysis (PCA) (Boumal, 2023, Section 2.4) and entails finding the q_j-dimensional subspace that best fits the data in the sense of minimizing the projection error. While alternative norms (e.g., ℓ_1) could provide improved robustness under non-Gaussian noise (Candès et al., 2011), our focus in this work is on the geometric and algorithmic aspects of flag-based subspace tracking. It is well-known that PCA admits a closed-form solution (Eckart and Young, 1936) given by the top q_j singular vectors (principal vectors) of W_t. One approach to solving the proposed problem would be running d separate optimizations on each $\mathrm{Gr}(p, q_k)$ and select a suitable dimension, but this is computationally expensive and would yield optimal subspaces $\mathbf{V}_j^t := \arg\min_{\mathbf{U}\in\mathrm{Gr}(p,q_j)} g_{W_t}^j(\mathbf{U})$ that are not nested for $j = 1, \ldots, d$. The lack of nestedness has been shown (Szwagier and Pennec, 2025) to cause poor performance when using subspaces for tasks such as robust subspace recovery. Instead, we propose optimizing over $\mathrm{Flag}(p, q_{1:d})$. For each $t \in \mathbb{Z}_{\geq 0}$ we define

$$f_{W_t}(\mathbf{U}_{1:d}) = \left\| W_t - \frac{1}{d}\sum_{i=1}^{d} \Pi_{\mathbf{U}_i} W_t \right\|_F^2 \tag{3}$$

for all $\mathbf{U}_{1:d} \in \mathrm{Flag}(p, q_{1:d})$. One key difference of optimizing (3) instead of (2) is that we obtain a nested sequence of subspaces which can be leveraged for downstream tasks (e.g., data-driven prediction) via ensembling methods (Dietterich, 2000) (see Section 5.2). The operator $\frac{1}{d}\sum_{i=1}^{d} \Pi_{\mathbf{U}_i}$ demonstrates an ensemble of projections onto subspaces of different dimensions, which is called the "flag trick" in (Szwagier and Pennec, 2025, Definition 5). Similar to (2), minimizers of (3) can be computed in closed form by computing the top-q_d principal vectors of W_t (Szwagier and Pennec, 2025, Theorem 4). However, recomputing the full sample covariance $W_t W_t^\top$ and its leading eigenspaces at every time t can be computationally expensive, motivating recursive subspace tracking methods (Balzano et al., 2018). Based on results in optimization on flag manifolds (Ye et al., 2022; Zhu and Shen, 2024), we perform Riemannian gradient descent (RGD) (Boumal, 2023, Section 4.3) for (3). The goal of RGD is to seek a minimizer of the function (3) by iterative updates. In particular, for each new data sample w_t, we perform $K > 0$ steps along the gradient of (3). Suppose that the initial point is $\widehat{\mathbf{U}}_{1:d}^{T-1}$ (we use the superscript $T-1$ since we have to collect T data points to form the first window W_{T-1}). For each fixed $t \geq T-1$, we update

$$\widehat{\mathbf{U}}_{1:d}^{t,k+1} = \mathrm{Exp}_{\widehat{\mathbf{U}}_{1:d}^{t,k}}\left(-\alpha_t \mathrm{grad}\, f_{W_t}(\widehat{\mathbf{U}}_{1:d}^{t,k})\right), \tag{4}$$

where $k = 0, \ldots, K-1$, $\alpha_t > 0$ is the step size, and we denote $\widehat{\mathbf{U}}_{1:d}^{T-1,0} = \widehat{\mathbf{U}}_{1:d}^{T-1}$, $\widehat{\mathbf{U}}_{1:d}^{t+1} := \widehat{\mathbf{U}}_{1:d}^{t,K}$, $\widehat{\mathbf{U}}_{1:d}^{t+1,0} = \widehat{\mathbf{U}}_{1:d}^{t+1}$, for all $t \geq T-1$. The Riemannian gradient $\mathrm{grad}\, f_{W_t}$ is a vector field on the manifold, and $\mathrm{Exp}_{\widehat{\mathbf{U}}_{1:d}^{t,k}}$ is the exponential map, which maps the tangent vector $-\alpha_t \mathrm{grad}\, f_{W_t}(\widehat{\mathbf{U}}_{1:d}^{t,k})$ back to the manifold. Riemannian gradient and exponential map can be computed using only matrix algebra (Ye et al., 2022).

4.2. Learning time- and dimension-varying subspaces

Having established the necessary foundations, we now present the **F**lag **R**ecursive **ON**line **T**racking (FRONT) algorithm given by Algorithm 1. The output of FRONT is a sequence of flags, rather than selecting a single subspace from $\mathbf{U}_{1:d}^t$. We aggregate information across subspaces of different dimensions, with their usage depending on the specific task. Since a flag is represented by an

Algorithm 1 FRONT

Require: Window length $T \geq 1$; initial estimate $\widehat{\mathbf{U}}_{1:d}^{T-1}$; samples $\{w_t\}_{t\in\mathbb{Z}_{\geq 0}}$; step sizes α_t

1: **for** $t = T-1, T, \dots$ **do**
2: Construct $W_t = [w_{t-T+1} \cdots w_t]$
Initialize gradient descent: $\widehat{\mathbf{U}}_{1:d}^{t,0} = \widehat{\mathbf{U}}_{1:d}^{t}$
3: **for** $k = 0, 1, \dots, K-1$ **do**
Gradient step:
$\widehat{\mathbf{U}}_{1:d}^{t,k+1} = \mathrm{Exp}_{\widehat{\mathbf{U}}_{1:d}^{t,k}}\left(-\alpha_t \mathrm{grad} f_{W_t}(\widehat{\mathbf{U}}_{1:d}^{t,k})\right)$
4: **Update:** $\widehat{\mathbf{U}}_{1:d}^{t+1} = \widehat{\mathbf{U}}_{1:d}^{t,K}$

Algorithm 2 GREAT

Require: Window length $T \geq 1$; initial estimate $\widehat{\mathbf{U}}^{T-1}$; samples $\{w_t\}_{t\in\mathbb{Z}_{\geq 0}}$; step sizes α_t

1: **for** $t = T-1, T, \dots$ **do**
2: Construct $W_t = [w_{t-T+1} \cdots w_t]$
Initialize gradient descent: $\widehat{\mathbf{U}}^{t,0} = \widehat{\mathbf{U}}^{t}$
3: **for** $k = 0, 1, \dots, K-1$ **do**
Gradient step:
$\widehat{\mathbf{U}}^{t,k+1} = \mathrm{Exp}_{\widehat{\mathbf{U}}^{t,k}}\left(-\alpha_t \mathrm{grad} f_{W_t}(\widehat{\mathbf{U}}^{t,k})\right)$
4: **Update:** $\widehat{\mathbf{U}}^{t+1} = \widehat{\mathbf{U}}^{t,K}$

orthonormal matrix in $\mathbb{R}^{p\times q_d}$, the per-time-step complexity of FRONT is $O(Kp^2q_d)$, by the same covariance-based complexity argument as in (Sasfi et al., 2025, Remark 5). FRONT is similar in structure to GREAT (Sasfi et al., 2025), but generalizes GREAT by learning an ensemble of models with different complexity while not assuming that the true data-generating subspaces have fixed dimension. Moreover, when $d = 1$ so that $\mathrm{Flag}(p, (q_1)) = \mathrm{Gr}(p, q_1)$, FRONT is equivalent to GREAT. To prove this, we make a definition on algorithm equivalence.

Definition 1 *Let M be a Riemannian manifold. We define an algorithm $\mathcal{A}$ on M as an update mapping: $\mathcal{A} : M \to M$ by $U^{t+1} = \mathcal{A}(U^t)$.*

Definition 2 *Let $\mathcal{A}_1, \mathcal{A}_2$ be two algorithms on M such that $U^{t+1} = \mathcal{A}_1(U^t)$ and $V^{t+1} = \mathcal{A}_2(V^t)$, respectively. We say that $\mathcal{A}_1$ is equivalent to $\mathcal{A}_2$ if $U^0 = V^0$ implies $U^t = V^t$ for all $t \geq 1$.*

Proposition 3 *Let $\mathcal{A}_1 : \mathrm{Flag}(p, (q)) \to \mathrm{Flag}(p, (q))$ be defined by (4), and $\mathcal{A}_2 : \mathrm{Gr}(p, q) \to \mathrm{Gr}(p, q)$ be defined by the update rule in GREAT. Then, $\mathcal{A}_1$ is equivalent to $\mathcal{A}_2$.*

Proof By definition, $\mathrm{Flag}(p, (q)) = \mathrm{Gr}(p, q)$, so $\mathcal{A}_1$ and $\mathcal{A}_2$ are defined on the same manifold. Let $\mathbf{U}^0$ be the initial point of $\mathcal{A}_1$ and $\mathbf{V}^0$ be the initial point of $\mathcal{A}_2$. Suppose $\mathbf{U}^0 = \mathbf{V}^0$. The cost function (3) becomes $f_{W_t}(\widehat{\mathbf{U}}) = \|W_t - \Pi_{\widehat{\mathbf{U}}} W_t\|_F^2$, which coincides with $g_{W_t}(\widehat{\mathbf{U}})$. For each $t = T-1, T, \dots$ and $k \in \{0, \dots, K-1\}$, let $\mathbf{U}^{t,k}$ and $\mathbf{V}^{t,k}$ be the points obtained by the gradient step in FRONT and GREAT, respectively. Since $\mathrm{Flag}(p, (q))$ and $\mathrm{Gr}(p, q)$ are the same manifold, their intrinsic geometric structures coincide. Thus, the exponential map and the Riemannian gradients are identical. This can also be verified by the formulas in (Ye et al., 2022) for $\mathrm{Flag}(p, (q))$ with those in (Edelman et al., 1998) for $\mathrm{Gr}(p, q)$. Therefore, we have $\widehat{\mathbf{U}}^{t,k} = \widehat{\mathbf{V}}^{t,k}$ for all $t \in \{T-1, T, \dots\}$ and $k \in \{0, \cdots, K-1\}$, hence $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent. ■

Under assumptions on quantitative persistency of excitation of the data and bounded subspace variation (Sasfi et al., 2025, Assumptions 1-4), all convergence guarantees and finite-sample analysis of (Sasfi et al., 2025, Theorem 1) directly extend to FRONT on $\mathrm{Flag}(p, (q))$. In particular, (Sasfi et al., 2025) derives finite-sample upper bounds on the distance $d(\widehat{\mathbf{U}}^t, \mathbf{U}^t)$ in terms of subspace change rate, and singular values of the data window W_t. These results provide guaranteed convergence rates to the true underlying subspace and therefore give a quantitative method for determining conditions on data needed to satisfy $d(\widehat{\mathbf{U}}^t, \mathbf{U}^t) < \epsilon_{\text{tol}}$ for a given ϵ_{tol}. Beyond this trivial signature, we do not claim convergence guarantees for FRONT in this paper. In the future work, we wish to study

finite-sample convergence guarantees for FRONT in the more general case when the flag does not have a trivial signature $(p, (q))$. The nontrivial flag-manifold setting is supported by the following numerical case studies.

5. Numerical case studies

Throughout all numerical examples[1], when implementing FRONT, we use the Python class `Flag` developed in (Szwagier and Pennec, 2025) and use the `SteepestDescent` optimizer from Pymanopt (Townsend et al., 2016), which uses backtracking line-search (Boumal, 2023) to specify the step sizes α_t used for the gradient steps in FRONT. The number of gradient steps is $K = 5$.

5.1. Geodesic tracking

We generate a sequence of time-varying ground-truth subspaces on $\mathrm{Flag}(10, (1, \dots, 5))$. The initial flag $\mathbf{U}^0$ has Stiefel representation $U^0 = I_{10\times 5}$. For each $t \in \mathbb{Z}_{\geq 0}$ we update $\mathbf{U}^{t+1} = \mathrm{Exp}_{\mathbf{U}^t}(\alpha \mathbf{H}^t)$, where $\mathbf{H}^t$ is a random tangent vector (Boumal, 2023, Section 8.3) at $\mathbf{U}^t$ and $\alpha = 5 \cdot 10^{-5}$. At time $T_{\text{switch}} = 100$, we increase the subspace dimension by setting $U^{T_{\text{switch}}} = I_{10\times 6}$ and generate subsequent flags in the same way on $\mathrm{Flag}(10, (1, \dots, 6))$. At each time t we observe a data point $w_t = U^t a_t + \eta_t$, where $a_t \sim \mathcal{N}(0, I)$ and $\eta_t \sim \mathcal{N}(0, 10^{-4} I)$ is noise. For each $t \geq T - 1$, the data window is constructed as $W_t = [w_{t-T+1} \dots w_t]$. The subspace estimate is updated online by running FRONT on $\mathrm{Flag}(10, (5, 6))$. We set the initial estimate $\widehat{\mathbf{U}}^T$ to be random and run with data window size $T \in \{1, 20, 50\}$. For each $T \in \{1, 20, 50\}$, we run 100 experiments across different random data sets $\{w_t\}_{t=1}^{200}$.

Figure 1 shows how $d(\widehat{\mathbf{U}}^t, \mathbf{U}^t)$ evolves over time $t = T - 1, \dots, 200$ for different window sizes $T \in \{1, 20, 50\}$. For all $T \in \{1, 20, 50\}$, the distance decreases as more data are available, indicating that the estimates gradually align with the true subspaces. At time T_{switch}, the distance sharply increases due to the sudden change in subspace dimension, after which FRONT readapts and the distance decreases again. The plateau for $T = 1$ after switching arises because a single-sample update does not provide sufficient information to identify the new subspace, whereas larger data window sizes ($T = 20, 50$) yield better estimates. Although this example considers an increase in subspace dimension, the problem setup in Section 3 does not assume increasing complexity. Decreasing subspace dimension can likewise be accommodated, since each flag contains all lower-dimensional nested subspaces.

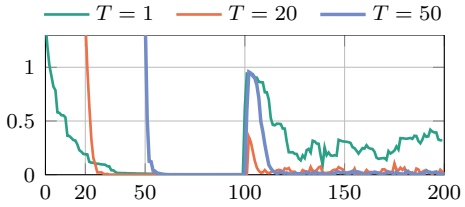

Figure 1: Average chordal distance for different window sizes across 100 experiments.

5.2. Online data-driven prediction for a switched system

The data-driven prediction problem posed in (Markovsky and Rapisarda, 2008, Section 4) aims to find the output sequence of an unknown dynamical system corresponding to an input sequence and a past input-output trajectory collected from the system. For LTI systems, data-driven prediction can be performed using a matrix representation of the set of finite-length trajectories (restricted behavior) (Favoreel et al., 1999). Motivated by the subspace representations of dynamical systems in Section 3, we use FRONT to learn the subspace of finite-length trajectories of a time- and

1. The code is available at https://github.com/DianJin-Frederick/FRONT-experiments.

complexity-varying system and perform data-driven prediction. We consider the data-driven prediction task for a single-input single-output (SISO) ARX system, whose dynamics is given by

$$\begin{cases} y_t = 0.3y_{t-1} - 0.02y_{t-2} + 0.6u_{t-1} + 0.2u_{t-2}, & t < T_{\text{switch}}, \\ y_t = 1.5y_{t-1} - 0.74y_{t-2} + 0.12y_{t-3} + 0.6u_{t-1} + 0.2u_{t-2} + 0.05u_{t-3}, & t \geq T_{\text{switch}}, \end{cases} \tag{5}$$

The input dimension is $m = 1$ and output dimension is $n = 1$. The main purpose of this example is to test adaptation to a change in system complexity, since the underlying system order increases at T_{switch}. This is precisely the setting that motivates the use of a flag, as it incorporates information from subspaces across all candidate dimensions. Let $T_{\text{sim}} = 300$ and $T_{\text{switch}} = 100$. Given a random input sequence $u_{[0,T_{\text{sim}}-1]}$ and a prediction horizon $T_{\text{f}} \in \mathbb{Z}_{\geq 0}$, we wish to predict the corresponding future output trajectory $y_{[0,T_{\text{sim}}-1]}$ of (5). We assume that the measurement of output at time t is corrupted by the independent noise η_t for $t \in [0, T_{\text{sim}} - 1]$. The noise-to-signal ratio (NSR) at time $t \in [0, T_{\text{sim}} - 1]$ is defined as $\sigma_t := \|\eta_t\|_2 / \|y_t\|_2$. In our experiments, we fix this ratio to a constant $\sigma > 0$ for all t. The measurement noise is generated as $\eta_t \sim \mathcal{N}\big(0,\ (\sigma\|y_t\|_2)^2 I\big)$.

Offline Data Collection. Assume that we have access to a noisy input-output trajectory of (5) of length $T_{\text{d}} = 30$ denoted by $(u^{\text{d}}_{[0,T_{\text{d}}-1]}, y^{\text{d}}_{[0,T_{\text{d}}-1]} + \eta^{\text{d}}_{[0,T_{\text{d}}]})$, where $\eta^{\text{d}}_t \sim \mathcal{N}\big(0,\ (\sigma\|y^{\text{d}}_t\|_2)^2 I\big)$ for $t \in [0, T_{\text{d}} - 1]$. Let $T_{\text{ini}} = 4$, $T_{\text{f}} = 4$, and $L = T_{\text{ini}} + T_{\text{f}} = 8$. We sort the offline input-output data into a Hankel matrix

$$H := \begin{bmatrix} u^{\text{d}}_{[0,L-1]} & \cdots & u^{\text{d}}_{[T_{\text{d}}-L,T_{\text{d}}-1]} \\ y^{\text{d}}_{[0,L-1]} + \eta^{\text{d}}_{[0,L-1]} & \cdots & y^{\text{d}}_{[T_{\text{d}}-L,T_{\text{d}}-1]} + \eta^{\text{d}}_{[T_{\text{d}}-L,T_{\text{d}}-1]} \end{bmatrix}$$

with the thin singular value decomposition (SVD) being $H = U\Sigma V^\top$, where $U \in \mathbb{R}^{(m+n)L \times r}$ is orthonormal with $r = \operatorname{rank} H$. Let $U = [u_1 \dots u_r]$, we set the initial point of FRONT to be $\mathbf{V}^0_{1:d} = \{\mathbf{V}^0_k\}_{k=1}^d \in \operatorname{Flag}((m+n)L, (q_1, \dots, q_d))$, where $\mathbf{V}^0_k = \operatorname{span}\{u_1, \cdots, u_{q_k}\}$ for $k \in [1, d]$, and $q_d = r$.

Subspace Predictor. Let $V \in \mathbb{R}^{(m+n)L \times r}$ be any orthonormal matrix with the partition $V = [V_{\text{p}}^\top\ V_{\text{f}}^\top\ Y_{\text{p}}^\top\ Y_{\text{f}}^\top]$, where $V_{\text{p}} \in \mathbb{R}^{mT_{\text{ini}} \times r}, V_{\text{f}} \in \mathbb{R}^{mT_{\text{f}} \times r}, Y_{\text{p}} \in \mathbb{R}^{nT_{\text{ini}} \times r}, Y_{\text{f}} \in \mathbb{R}^{nT_{\text{f}} \times r}$. The T_{f}-length predicted output at time t is given by

$$y^{\text{pred}}_{[t,t+T_{\text{f}}-1]}(V) = \begin{bmatrix} y^{\text{pred}}_t(V) \\ \vdots \\ y^{\text{pred}}_{t+T_{\text{f}}-1}(V) \end{bmatrix} = Y_{\text{f}} \begin{bmatrix} V_{\text{p}} \\ V_{\text{f}} \\ Y_{\text{p}} \end{bmatrix}^\dagger \begin{bmatrix} u_{[t-T_{\text{ini}},t-1]} \\ u_{[t,t+T_{\text{f}}-1]} \\ y_{[t-T_{\text{ini}},t-1]} + \eta_{[t-T_{\text{ini}},t-1]} \end{bmatrix} \in \mathbb{R}^{nT_{\text{f}}}, \tag{6}$$

which is known as the orthonormal subspace predictor (Jin and Coulson, 2026). In order to cope with the changing dynamics of (5), we use FRONT to update the subspace (restricted behavior of (5): see Section 3) based on streaming data $w_t := (u_{[t-T+1,t]}, y_{[t-T+1,t]} + \eta_{[t-T+1,t]})$ and initial point $\mathbf{V}^0_{1:d}$, where $\eta_t \sim \mathcal{N}(0, (\sigma\|y_t\|_2)^2 I)$. We then update the matrix $Y_{\text{f}}([V_{\text{p}}^\top\ V_{\text{f}}^\top\ Y_{\text{p}}^\top]^\top)^\dagger$ in (6) based on the learned flag. Let $\widehat{\mathbf{V}}^t_{1:d} = \{\widehat{\mathbf{V}}^t_k\}_{k=1}^d$ be the t^{th} flag estimate produced by FRONT. The Stiefel representative $\widehat{V}^t_{1:k}$ of $\widehat{\mathbf{V}}^t_k$ is obtained by taking the first q_k columns of the Stiefel representative $\widehat{V}^t_{1:d}$ of $\widehat{\mathbf{V}}^t_{1:d}$. Then we partition the rank-q_k matrix $\widehat{V}^t_{1:k}$ as $\widehat{V}^t_{1:k} = [V_{\text{p},k}^\top\ V_{\text{f},k}^\top\ Y_{\text{p},k}^\top\ Y_{\text{f},k}^\top]^\top$ where $V_{\text{p},k} \in \mathbb{R}^{mT_{\text{ini}} \times q_k}$, $V_{\text{f},k} \in \mathbb{R}^{mT_{\text{f}} \times q_k}$, $Y_{\text{p},k} \in \mathbb{R}^{nT_{\text{ini}} \times q_k}$, $Y_{\text{f},k} \in \mathbb{R}^{nT_{\text{f}} \times q_k}$. For each k, we get the prediction $y^{\text{pred}}_{[t,t+T_{\text{f}}-1]}(\widehat{V}^t_{1:k})$ defined in (6), and keep only its first component $y^{\text{pred}}_t(\widehat{V}^t_{1:k})$, simplifying

its notation as $\widehat{y}_t(k) \in \mathbb{R}^n$. Thus, by leveraging the nestedness of a flag, at each time t, we obtain d different predictions $\widehat{y}_t(1), \ldots, \widehat{y}_t(d)$ from matrices of different ranks based on only one matrix $\widehat{V}^t_{1:d}$. Linking back to the motivation made in Section 4.1, we compare against separately running FRONT on $\mathrm{Gr}(16, q_k)$ for $k \in [1, d]$ and use the individually learned subspace $\widehat{\mathbf{U}}^t_k \in \mathrm{Gr}(16, q_k)$ for prediction. In contrast, we use each $\widehat{\mathbf{V}}^t_k$ with $k \in [1, d]$ from a single flag $\widehat{\mathbf{V}}^t_{1:d}$ for prediction, requiring only one optimization instead of d separate runs. We measure the prediction performance by the cumulative prediction error over the whole simulation horizon $T_{\mathrm{sim}} - T_{\mathrm{f}}$ given by $\sum_{t=0}^{T_{\mathrm{sim}}-T_{\mathrm{f}}-1} \|y_t - \widehat{y}_t\|_2^2$, where $\widehat{y}_t$ is the prediction at time t from the corresponding model. We set $T_{\mathrm{ini}} = T_{\mathrm{f}} = 4$ and the window size $T = 20$.

Figure 2 shows the median cumulative prediction errors of both approaches, where we set $d = 8$ and $(q_1, \ldots, q_d) = (8, \ldots, 15)$. The NSR is set to be $\sigma = 0.02$. All models were evaluated over 100 independent trials, and in each trial, all the models use the same offline input-output data $(u^{\mathrm{d}}_{[0,T_{\mathrm{d}}-1]}, y^{\mathrm{d}}_{[0,T_{\mathrm{d}}-1]} + \eta^{\mathrm{d}}_{[0,T_{\mathrm{d}}]})$, input sequence $u_{[0,T_{\mathrm{sim}}-1]}$ and measurement noise $\eta_{[0,T_{\mathrm{sim}}-1]}$. The results demonstrate that the nested subspace predictions from a single flag achieve median prediction errors comparable to those of individually optimized Grassmannians across many dimensions. We observe that the choice of subspace dimension can significantly affect performance. Optimal choice of subspace dimension is an area of future work. In the next part, we propose an ensembling strategy for mitigating the risk of poor subspace dimension selection.

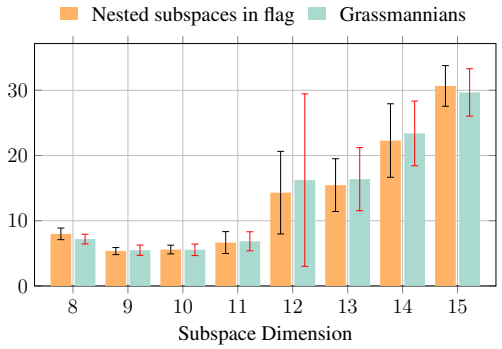

Figure 2: Median cumulative prediction errors (y-axis): a single flag evaluated at different ranks (orange) versus individually learned Grassmann subspaces (green). The whiskers span 30$^{\text{th}}$ to 70$^{\text{th}}$ percentiles.

Ensemble of Nested Subspace Prediction. Since the true system order is unknown, relying on a single subspace from the flag estimate may lead to biased predictions. To mitigate this, inspired by ensembling methods (Dietterich, 2000), we aggregate information from all candidate subspaces $\{\widehat{\mathbf{V}}^t_k\}_{k=1}^d$. Specifically, we compute an ensemble of predictions by averaging $\widehat{y}_t(k)$ obtained from (6) across $k = 1, \ldots, d$, yielding $\widehat{y}_t := \frac{1}{d}\sum_{k=1}^d \widehat{y}_t(k)$. We use uniform averaging as a simple baseline ensemble rule. More sophisticated weighting schemes, for example based on the spectrum of the data covariance, may further improve performance and are an interesting direction for future work. This approach leverages the hierarchical structure of the flag to balance the contribution of lower- and higher-dimensional subspaces, yielding more robust online predictions. We name this model as **average-$(q_1, \ldots, q_d)$ prediction.** We choose an ensemble of dimension $q_1 = 9$ and $q_2 = 10$ based on validation and evaluate the average-$(9, 10)$ prediction against several baseline models: no learning, N4SID, projection approximation subspace tracking (PAST). The *no learning* model predicts future outputs without any adaptive updates to the subspace predictor. The N4SID method (Van Overschee and De Moor, 1994) is implemented in an online fashion: at each time t we re-identify a state-space model (A_t, B_t, C_t, D_t) from the current input-output data window $W_t = [w_{t-T+1} \cdots w_t]$. The state is estimated by a Kalman filter. Then the estimated state-space model is used to compute $\widehat{y}_t$. The PAST algorithm (Yang, 1995) performs recursive subspace tracking based on each new w_t, where the estimated subspace is used in (6) for prediction. In addition, we evaluate the average-$(8, \ldots, 11)$ prediction to examine the performance when combining subspaces with relatively high cumulative prediction errors (see Figure 2). We vary the NSR level as $\sigma \in$

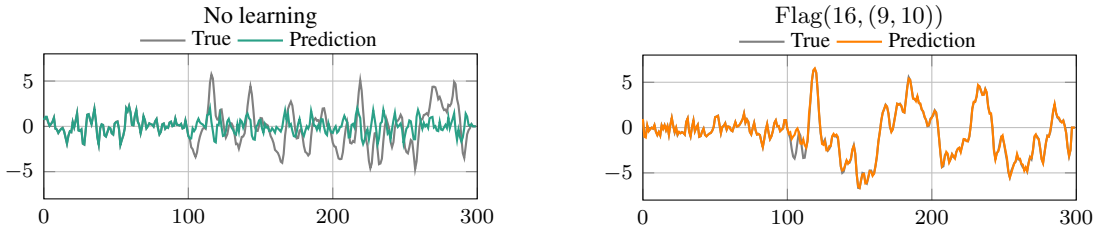

Figure 3: True and predicted trajectories. Vertical axis: output. Horizontal axis: time steps. Left: no learning baseline. Right: the average-$(9, 10)$ prediction model adapts and tracks the system after $t = T_{\text{switch}}$ quickly.

$\{0.01, 0.02, 0.05, 0.1\}$. For each NSR level, we conduct 100 independent trials. All models are evaluated using the same 100 offline input-output data sets, identical input sequences, and the same noise realizations across all NSR levels to ensure a fair comparison.

Results. Figure 3 shows, for the no learning baseline and the average-$(9, 10)$ prediction model, the predicted trajectory and the corresponding ground truth from the trial whose cumulative prediction error is closest to the median over 100 trials. Figure 4 reports the median cumulative prediction error versus the NSR for all models. The average-$(9, 10)$ prediction model achieves nearly the same median error as the Grassmann models $\mathrm{Gr}(16, 9)$ and $\mathrm{Gr}(16, 10)$, while requiring only a single optimization over one flag rather than separate optimizations for each dimension. Since the range $(9, 10)$ is chosen a posteriori, this experiment mainly illustrates the computational advantage of a single flag optimization. At all NSR levels, the average-$(8, \ldots, 11)$ model yields larger errors than the average-$(9, 10)$ model. Since the dimensions in the average-$(8, \ldots, 11)$ model are chosen from a broader range without assuming knowledge of the true system order, this experiment reveals a limitation of the proposed method: including dimensions that underfit or overfit the system can degrade prediction performance. The results highlight a trade-off between robustness to unknown model order and prediction accuracy, as flags reduce the need for dimension selection but may degrade when the chosen dimension range is not well aligned with the true system complexity.

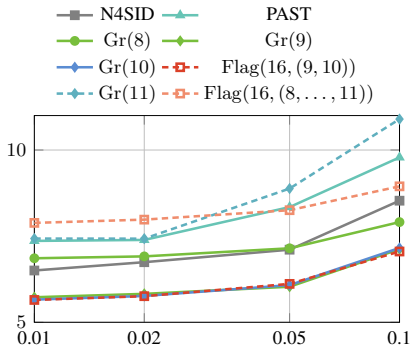

Figure 4: Median cumulative prediction error (y-axis) versus varying NSR (x-axis).

6. Conclusion

We proposed the FRONT algorithm for online system identification and data-driven prediction that adapts to time-varying systems by recursively updating a nested hierarchy of subspaces. This enables the predictor to accommodate unknown or changing system dimensions without prior model information. We proved that GREAT is recovered as a special case of FRONT. Numerical studies demonstrated that the proposed predictor effectively tracks abrupt changes in dynamics and consistently achieves better prediction error than most baselines even under high NSR. Future work focuses on finite sample convergence of FRONT for nontrivial flag signatures and integrating the proposed flag-based learning framework into an adaptive data-to-control pipeline.

References

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematica*, 80:199–220, 2004.

Laura Balzano and Stephen J Wright. Local convergence of an algorithm for subspace identification from partial data. *Foundations of Computational Mathematics*, 15:1279–1314, 2015.

Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *2010 48th Annual allerton conference on communication, control, and computing (Allerton)*, pages 704–711. IEEE, 2010.

Laura Balzano, Yuejie Chi, and Yue M Lu. Streaming PCA and subspace tracking: The missing data case. *Proceedings of the IEEE*, 106(8):1293–1310, 2018.

Julian Berberich, Johannes Köhler, Matthias A Müller, and Frank Allgöwer. Data-driven model predictive control with stability and robustness guarantees. *IEEE transactions on automatic control*, 66(4):1702–1717, 2020.

Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-enabled predictive control: In the shallows of the DeePC. In *2019 18th European control conference (ECC)*, pages 307–312. IEEE, 2019.

Jeremy Coulson, John Lygeros, and Florian Dörfler. Distributionally robust chance constrained data-enabled predictive control. *IEEE Transactions on Automatic Control*, 67(7):3289–3304, 2021.

Claudio De Persis and Pietro Tesi. Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Transactions on Automatic Control*, 65(3):909–924, 2019.

Jean-Pierre Delmas. Subspace tracking for signal processing. *Adaptive signal processing: next generation solutions*, pages 211–270, 2010.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

Florian Dörfler, Jeremy Coulson, and Ivan Markovsky. Bridging direct and indirect data-driven control formulations via regularizations and relaxations. *IEEE Transactions on Automatic Control*, 68(2):883–897, 2022.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

Ezzat Elokda, Jeremy Coulson, Paul N Beuchat, John Lygeros, and Florian Dörfler. Data-enabled predictive control for quadcopters. *International Journal of Robust and Nonlinear Control*, 31 (18):8916–8936, 2021.

Wouter Favoreel, Bart De Moor, and Michel Gevers. SPC: Subspace predictive control. *IFAC Proceedings Volumes*, 32(2):4004–4009, 1999.

Randall T. Fawcett, Kereshmeh Afsari, Aaron D. Ames, and Kaveh Akbari Hamed. Toward a data-driven template model for quadrupedal locomotion. *IEEE Robotics and Automation Letters*, 7 (3):7636–7643, 2022. doi: 10.1109/LRA.2022.3184007.

Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

Jun He, Laura Balzano, and Arthur Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1568–1575. IEEE, 2012.

Linbin Huang, Jeremy Coulson, John Lygeros, and Florian Dörfler. Decentralized data-enabled predictive control for power system oscillation damping. *IEEE Transactions on Control Systems Technology*, 30(3):1065–1077, 2021a.

Linbin Huang, Jianzhe Zhen, John Lygeros, and Florian Dörfler. Quadratic regularization of data-enabled predictive control: Theory and application to power converter experiments. *IFAC-PapersOnLine*, 54(7):192–197, 2021b.

Dian Jin and Jeremy Coulson. On the sensitivity of the subspace predictor to behavioral perturbations. *arXiv preprint arXiv:2603.17256*, 2026.

L Kerkhof and Tamás Keviczky. Predictive control of autonomous greenhouses: a data-driven approach. In *2021 European Control Conference (ECC)*, pages 1229–1235. IEEE, 2021.

Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.

Ivan Markovsky and Florian Dörfler. Behavioral systems theory in data-driven analysis, signal processing, and control. *Annual Reviews in Control*, 52:42–64, 2021.

Ivan Markovsky and Florian Dörfler. Identifiability in the behavioral setting. *IEEE Transactions on Automatic Control*, 68(3):1667–1677, 2022.

Ivan Markovsky and Paolo Rapisarda. Data-driven simulation and control. *International Journal of Control*, 81(12):1946–1959, 2008.

Hiroshi Oku and Hidenori Kimura. Recursive 4SID algorithms using gradient type subspace tracking. *Automatica*, 38(6):1035–1043, 2002.

Daniel J Rabideau. Fast, rank adaptive subspace tracking and applications. *IEEE Transactions on Signal Processing*, 44(9):2229–2244, 1996.

András Sasfi, Alberto Padoan, Ivan Markovsky, and Florian Dörfler. Great: Grassmannian recursive algorithm for tracking & online system identification. *IEEE Transactions on Automatic Control*, 2025.

Tom Szwagier and Xavier Pennec. Nested subspace learning with flags, 2025. URL https://arxiv.org/abs/2502.06022.

James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research*, 17(137):1–5, 2016. URL http://jmlr.org/papers/v17/16-177.html.

Peter Van Overschee and Bart De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

Peter Van Overschee and Bart De Moor. *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

Michel Verhaegen and Patrick Dewilde. Subspace model identification part 2. analysis of the elementary output-error state-space model identification algorithm. *International journal of control*, 56(5):1211–1241, 1992.

Michel Verhaegen and Vincent Verdult. *Filtering and system identification: a least squares approach*. Cambridge university press, 2007.

Chris Verhoek, Hossam S Abbas, Roland Tóth, and Sofie Haesaert. Data-driven predictive control for linear parameter-varying systems. *IFAC-PapersOnLine*, 54(8):101–108, 2021.

Jan C Willems. From time series to linear system—Part I. Finite dimensional linear time invariant systems. *Automatica*, 22(5):561–580, 1986.

Bin Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal processing*, 43(1):95–107, 1995.

Ke Ye and Lek-Heng Lim. Schubert varieties and distances between subspaces of different dimensions. *SIAM Journal on Matrix Analysis and Applications*, 37(3):1176–1197, 2016.

Ke Ye, Ken Sze-Wai Wong, and Lek-Heng Lim. Optimization on flag manifolds. *Mathematical Programming*, 194(1):621–660, 2022.

Hao Zhang, Clarence W Rowley, Eric A Deem, and Louis N Cattafesta. Online dynamic mode decomposition for time-varying systems. *SIAM Journal on Applied Dynamical Systems*, 18(3): 1586–1609, 2019.

Xiaojing Zhu and Chungen Shen. Practical gradient and conjugate gradient methods on flag manifolds. *Computational Optimization and Applications*, 88(2):491–524, 2024.