

Trajectory Intent Prediction of Autonomous Systems using Dynamic Mode Decomposition

Adolfo Perrusquía, *Member, IEEE*, Zhuangkun Wei *Member, IEEE*, Weisi Guo, *Senior Member, IEEE*

Abstract—Proliferation of autonomous systems have increased the threat space and the economic risk in several national infrastructures, e.g., at airports. Therefore, reliable detection of their intention is paramount to ensure smooth operation of national services and societal safety. This paper reports a data-driven trajectory intent prediction algorithm which is based on a linear model structure of the autonomous system dynamics obtained from a dynamic mode decomposition algorithm. The model computation is enhanced by two sources of physics informed knowledge associated to the energy functional. Two different prediction algorithms that consider fixed or time-varying references are designed in terms of the availability of control input measurements. Rigorous theoretical results are provided to support the approach using matrix decomposition and optimization techniques. Simulation and experimental studies are carried out to verify the effectiveness of the proposal.

Index Terms—Trajectory intent prediction, Dynamic Mode Decomposition, Physics Informed, Autonomous systems, Linear approximation.

I. INTRODUCTION

RESEARCH in predicting future behaviour and intent¹ has increased in recent years due to the high technological developments in autonomous systems. Whilst many autonomous systems pose no malicious intention, some of them carry out malicious capabilities that can compromise several national facilities [1] and hence, predicting their intention is mandatory to ensure societal safety. Two different tasks of intent prediction can be distinguished: high-level intent classification and trajectory intent prediction [2]. Intent classification task aims to predict the purpose of the autonomous system [3], [4], e.g., surveillance, delivery, inspection, pick and place, planning, etc. On the other hand, trajectory intent prediction task aims to predict the trajectory that the autonomous system will follow in future time steps for early detection and preventive control purposes [5]. In this paper, we focus on the trajectory intent prediction of autonomous systems using cooperative sensors (e.g., RF signals).

Research in trajectory intent prediction has been applied in the fields of robotics and smart navigation, e.g., in lane changing prediction of vehicles [6], pedestrian intent [7],

[8], maritime vessel intent forecasting [9], aircraft trajectory prediction [10], [11]. Machine learning techniques are the core methods used in the previous research studies, specifically: Bayesian approaches [3], [12], support vector regression [13], and linear regression [14], [15]. These methods require a large amount of data of similar trajectories to uncover the main features associated to the high-level intent. This is a strong restriction due to the complex nature and high variability of the intention.

Classic trajectory prediction techniques are model-based approaches [16], [17] that use state estimation techniques such as Kalman filters and their variants to make state and trajectory prediction. In most cases, a linear model is used for simplicity [18] which provides an unrealistic and low prediction capability in the long term. Actually, these models are good for short-term predictions since the model is used to predict one step prediction without regarding the control input. The development of intent-driven dynamical models [16] are used to enhance the previous approaches. These methods predict the future kinematics state and the inferred intended destination. However, they have high reliance on pre-defined trajectory plan and cannot be generalized for other similar systems and trajectories [14]. In terms of intent prediction of unmanned aerial vehicles (UAVs), genetic algorithms have been adopted to model dangerous drone intents using a support vector regressor (SVR) model of the UAV kinematics based on ADS-B flight data [13]. However, the above models require a large amount of labelled data.

Data-driven approaches have been designed for intent classification and prediction based on deep neural networks. Recent studies focus on human trajectory forecasting using deep generative models [19] and recurrent neural networks (RNNs) for trajectory modelling of robots and UAVs [20], [21] which are conditioned on long-term target goals [22]. These networks require a large amount of data and are usually unstable for small perturbations [23]. Other approaches use a human behaviour learning [24] approach to uncover the hidden utility or reward function associated to the main task of the autonomous system. The approach states that the reward function contains the necessary information about the desired reference or destination that the autonomous system wants to achieve. However, the approach is based on a linear model assumption which is a strict condition unless the linear model preserves the dynamic properties of the real system.

Gaussian processes [25], [26] combine the advantages of data-driven methods with kernel functions to estimate the posterior distribution associated to the future trajectory intent with high confidence prediction. The key idea is the design of

This work was supported by the Engineering and Physical Sciences Research Council under Grant EP/V026763/1 and by the Royal Academy of Engineering and the Office of the Chief Science Adviser for National Security under the UK Intelligence Community Postdoctoral Research Fellowship programme.

Adolfo Perrusquía, Zhuangkun Wei, and Weisi Guo are with the School of Aerospace, Transport and Manufacturing, Cranfield University, Bedford, UK (e-mail: Adolfo.Perrusquia-Guzman@cranfield.ac.uk).

¹In this paper, intent refers to the main task that the autonomous system aims to achieve either in a high-level purpose or the trajectory way-points.

adequate kernel functions that capture the autonomous system physics to infer its future trajectory. However, similarly to Kalman filter methods, these models are sensitive to the prior distribution and the selection of the kernel functions.

In the last decade, a data-driven approach known as Dynamic Mode Decomposition (DMD) [27] has gain popularity to model high-nonlinear systems as a linear-time invariant (LTI) system. The key idea of the DMD method is to solve a tractable optimization problem [28] using data collected through the system trajectories. In contrast to estimation algorithms that use simplistic linear models of the system, the linear model of the DMD saves the information about rotation, flipping, and scaling factors of the real-nonlinear system [29]. The algorithm has been modified to enhance its performance [30], [31] by considering nonlinear observables, delay coordinates, among others. These methods are usually applied to unforced systems that possess high nonlinear dynamical properties such as fluids, video, and biological systems. However, autonomous systems have not been studied in depth since most of them require external inputs to exhibit dynamic properties, e.g, mobile robots, drones, vessels, etc. DMD with control (DMDc) has been adopted to incorporate external inputs in the tractable optimization problem with good results. Here, DMD models and its variants have been focused on parameter identification tasks and dynamic model analysis. To best of our knowledge and the findings in the literature [27], [32], DMD has not been used for prediction purposes.

In view of the above, this paper reports a trajectory prediction algorithms for autonomous systems based on dynamic mode decomposition techniques. The algorithm is based on the local linear approximation of the autonomous system dynamics and enhanced by the control input structure to obtain reliable trajectory prediction with high accuracy and noise attenuation. The algorithms are explained theoretically in detail. Simulations and experiments in several case of studies are provided to observe the benefits, limitations, and future research opportunities.

The main features of this work are: 1) an on-line data-driven trajectory intent prediction algorithm based on DMD techniques and control input exploitation, 2) give evidence of the limitations of DMD models and future research directions, 3) the short-term prediction capabilities is enhanced by exploiting the control input components, and 4) the algorithm requires less data in comparison to machine and deep learning algorithms. The contributions of this work with respect to previous developments in trajectory intent prediction are the following: a) less computational complexity due to the low-dimensionality of autonomous systems, b) the obtained linear model preserved dynamic properties that are usually lost in linearization procedures, c) the algorithm can be performed on-line without imposing constraints and dynamic assumptions.

The paper outline is as follows: Section II establishes the preliminaries of the autonomous system dynamic equation. Section III defines three different DMD methods: the classical, with control, and physics informed. Section IV gives the trajectory prediction algorithm under two different scenarios: control-free and control-based. Section V and Section VI reports the simulation and experimental studies using differ-

ent autonomous platforms. The conclusions are presented in Section VII.

Throughout this paper, $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^{n \times m}$, $\mathbb{C}^{n \times m}$ denote the spaces of natural numbers, real numbers, real n -vectors, real $n \times m$ -matrices, and complex $n \times m$ -matrices, respectively; $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ denotes an identity matrix of $n \times n$; the norms $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top \mathbf{x}}$ and $\|\mathbf{X}\|_{\mathcal{F}} = \sqrt{\text{tr}\{\mathbf{X}^\top \mathbf{X}\}}$ stand for the Euclidean and Frobenius norms, respectively; $\text{tr}\{\cdot\}$ defines the trace function, where $x \in \mathbb{R}$ is a scalar, $\mathbf{x} \in \mathbb{R}^n$ is a vector, and $\mathbf{X} \in \mathbb{R}^{n \times m}$ is a matrix with $n, m \in \mathbb{N}$.

II. PRELIMINARIES

The general scheme of the proposed trajectory intent prediction problem is depicted in Fig. 1. The diagram shows three steps related to the approach: 1) online data collection of the autonomous system trajectories, 2) the application of the DMD method to obtain a local linear model from the collected data which will be used for prediction purposes, an 3) the design of the intention prediction algorithm based on the previous obtained model whose aim is to predict the trajectory intent associated to the reference signal. In the following sections, each step is explained in detail.

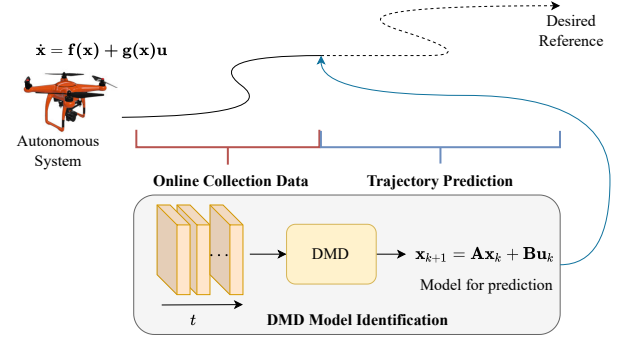


Fig. 1. Trajectory Intent Prediction problem

Assume the autonomous system fulfils the following second order differential equation

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{q})[\mathbf{L}(\mathbf{q})\mathbf{u} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q})], \quad (1)$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{m \times m}$ is a known symmetric and positive definite inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{m \times m}$ defines the Coriolis and centripetal forces matrix, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^m$ is the gravitational forces and torques vector, $\mathbf{L}(\mathbf{q}) \in \mathbb{R}^{m \times p}$ is a transformation matrix from $\mathbb{R}^p$ to $\mathbb{R}^m$, $\mathbf{u} \in \mathbb{R}^p$ stands to the control inputs, and $\mathbf{q} \in \mathbb{R}^m$ defines the generalized coordinates of the autonomous system. The nonlinear model (1) can be expressed in state space notation as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad (2)$$

where

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{\mathbf{q}} \\ -\mathbf{M}^{-1}(\mathbf{q})[\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})] \end{bmatrix} \in \mathbb{R}^n$$

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{m \times p} \\ \mathbf{M}^{-1}(\mathbf{q})\mathbf{L}(\mathbf{q}) \end{bmatrix} \in \mathbb{R}^{n \times p}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \in \mathbb{R}^n, \quad n = 2m.$$

Here, we want to construct an approximate local linear dynamical system from observed data along the system trajectories (2) which captures the performance of the autonomous system. This model will be further used to predict the future trajectory intent.

Assume the data from the autonomous system's trajectories are collected with a sample time $T > 0$ such that the nonlinear model (2) can be expressed as the following discrete-time system

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k) + \mathbf{G}(\mathbf{x}_k)\mathbf{u}_k, \quad (3)$$

where

$$\begin{aligned} \mathbf{F}(\mathbf{x}_k) &:= T\mathbf{f}(\mathbf{x}) + \mathbf{x}_k, \\ \mathbf{G}(\mathbf{x}_k) &:= T\mathbf{g}(\mathbf{x}). \end{aligned}$$

To obtain a local linear dynamical model from the autonomous system's trajectories, a constant input dynamics is required to avoid state dependent approximation. Hence, the nonlinear model (3) is rewritten as

$$\mathbf{x}_{k+1} \approx \mathbf{F}(\mathbf{x}_k) + \mathbf{B}\mathbf{u}_k, \quad (4)$$

where

$$\mathbf{B} = \max_{\mathbf{x}} \mathbf{G}(\mathbf{x}) \in \mathbb{R}^{n \times p}.$$

III. DYNAMIC MODE DECOMPOSITION

Traditional dynamic mode decomposition (DMD) [27] techniques model the nonlinear system (2) as an unforced system of the form

$$\dot{\mathbf{x}} = \mathbf{h}(\mathbf{x}), \quad (5)$$

where $\mathbf{h}(\mathbf{x}) \in \mathbb{R}^n$ is a nonlinear function. The equivalent discrete-form for a sample time $T > 0$ is

$$\mathbf{x}_{k+1} = \mathbf{H}(\mathbf{x}_k), \quad (6)$$

where $\mathbf{H}(\mathbf{x}_k) \in \mathbb{R}^n$ defines the autonomous system's dynamics under a hidden state dependent control input. In this scenario, DMD aims to find a tractable linear approximation [28] of (6) as

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k, \quad (7)$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is an unknown linear matrix that minimizes the following cost index

$$e = \|\mathbf{x}_{k+1} - \mathbf{A}\mathbf{x}_k\|_2 \quad (8)$$

across all points for $k = 1, \dots, \ell - 1$. The following matrices are constructed from the $\ell - 1$ samples

$$\begin{aligned} \mathbf{X} &= \begin{bmatrix} | & | & \cdots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{\ell-1} \\ | & | & \cdots & | \end{bmatrix} \in \mathbb{R}^{n \times \ell-1}, \\ \mathbf{X}' &= \begin{bmatrix} | & | & \cdots & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_\ell \\ | & | & \cdots & | \end{bmatrix} \in \mathbb{R}^{n \times \ell-1}, \end{aligned} \quad (9)$$

which verify the following linear system of equations

$$\mathbf{X}' = \mathbf{A}\mathbf{X}. \quad (10)$$

In this case the minimization problem is better represented by

$$E = \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_{\mathcal{F}}. \quad (11)$$

The exact matrix $\mathbf{A}$ that minimizes the error function E can be obtained in a least-squares sense as

$$\mathbf{A} = \mathbf{X}'\mathbf{X}^\dagger, \quad (12)$$

where † denotes the Moore-Penrose pseudoinverse. This solution typically has two main problems: i) the computational complexity increases as the number of data increases, and ii) noisy measurements yields to wrong estimations and hence the prediction accuracy will be compromised [33]. To overcome these issues, DMD executes a low-rank matrix decomposition [29] to consider the eigenvectors associated to the highest eigenvalues as

$$\begin{aligned} \mathbf{X} &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \\ &= \begin{bmatrix} \tilde{\mathbf{U}} & \tilde{\mathbf{U}}_r \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{\Sigma}} & \mathbf{0}_{r \times \ell-1-r} \\ \mathbf{0}_{n-r \times r} & \tilde{\mathbf{\Sigma}}_r \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{V}}^* \\ \tilde{\mathbf{V}}_r^* \end{bmatrix} \\ &\approx \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^*, \end{aligned} \quad (13)$$

where $\mathbf{U} \in \mathbb{C}^{n \times n}$ stands to the left singular vectors, $\mathbf{\Sigma} \in \mathbb{C}^{n \times (\ell-1)}$ define the singular values, and $\mathbf{V} \in \mathbb{C}^{\ell-1 \times \ell-1}$ are the right singular values. Here * denotes the complex conjugate transpose. The terms $\tilde{\mathbf{U}} \in \mathbb{C}^{n \times r}$, $\tilde{\mathbf{\Sigma}} \in \mathbb{C}^{r \times r}$, and $\tilde{\mathbf{V}} \in \mathbb{C}^{\ell-1 \times r}$ define a low-rank approximation of rank r associated to the r largest singular values. The terms $\tilde{\mathbf{U}}_r$, $\tilde{\mathbf{\Sigma}}_r$, and $\tilde{\mathbf{V}}_r$ indicate the remaining $\ell - 1 - r$ singular vectors and values. Therefore, the matrix $\mathbf{A}$ can be approximated as

$$\mathbf{A} \approx \mathcal{A} = \mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^*. \quad (14)$$

So, the linear approximation of the autonomous system's dynamics is given by

$$\mathbf{x}_{k+1} = \mathcal{A}\mathbf{x}_k. \quad (15)$$

Since the number of states of the autonomous system is small in comparison to the dimension of the sample data, then the SVD approximation has a good performance [29]. However, instead of using the n states, DMD can find a low-dimensional representation that encompass the dynamics of the autonomous system efficiently. This low-representation can be expressed as a projection mapping to the basis of eigenvectors associated to the largest eigenvalues, that is, $(\tilde{\mathbf{U}}^*\tilde{\mathbf{U}})^{-1}\tilde{\mathbf{U}}^*\mathbf{x} = \tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}}$ is a low-dimensional representation of $\mathbf{x}$ and recall that the singular vectors $\tilde{\mathbf{U}}$ are orthonormal, i.e., $(\tilde{\mathbf{U}}^*\tilde{\mathbf{U}})^{-1}\tilde{\mathbf{U}}^*\mathbf{x} = \tilde{\mathbf{U}}^*\mathbf{x} = \tilde{\mathbf{x}}$. Then, the linear approximation is rewritten as

$$\begin{aligned} \tilde{\mathbf{x}}_{k+1} &= \tilde{\mathbf{U}}^*\mathcal{A}\tilde{\mathbf{U}}\tilde{\mathbf{x}}_k \\ &= \tilde{\mathbf{U}}^*\mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{x}}_k \\ &= \tilde{\mathbf{A}}\tilde{\mathbf{x}}_k, \end{aligned} \quad (16)$$

where $\tilde{\mathbf{A}} \in \mathbb{R}^{r \times r}$ is a low-dimensional representation of the linear approximation mapping $\mathcal{A}$ given by

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}}^*\mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}. \quad (17)$$

The dynamical properties of this representation can be obtained by applying the eigendecomposition of matrix $\tilde{\mathbf{A}}$ which verifies the standard definition

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}, \quad (18)$$

where $\mathbf{W} \in \mathbb{C}^{r \times r}$ and $\mathbf{\Lambda} \in \mathbb{C}^{r \times r}$ denote the eigenvectors and eigenvalues of matrix $\tilde{\mathbf{A}}$, respectively. Here, we want to reconstruct the eigendecomposition of matrix $\mathbf{A}$ from the eigenvalues $\mathbf{\Lambda}$ and eigenvectors $\mathbf{W}$ of $\tilde{\mathbf{A}}$. For this purpose, $\mathbf{A}$ and $\tilde{\mathbf{A}}$ can be rewritten as

$$\begin{aligned} \mathbf{A} &= \mathbf{S}\tilde{\mathbf{U}}^* \\ \tilde{\mathbf{A}} &= \tilde{\mathbf{U}}^*\mathbf{S}, \end{aligned} \quad (19)$$

where $\mathbf{S} = \mathbf{X}'\tilde{\mathbf{V}}\tilde{\Sigma}^{-1}$. If (18) is verified with eigenvalues $\lambda \neq 0$, then we seek the eigenvectors Φ of $\mathbf{A}$ that fulfils the following eigenvalues equality

$$\mathbf{A}\Phi = \mathbf{S}\tilde{\mathbf{U}}^*\Phi = \Phi\mathbf{\Lambda}. \quad (20)$$

It is easy to verify that the eigenvectors Φ , also known as dynamic modes [30], that fulfils the above equality and the relations (19) are computed as

$$\Phi = \mathbf{S}\mathbf{W} = \mathbf{X}'\tilde{\mathbf{V}}\tilde{\Sigma}^{-1}\mathbf{W}. \quad (21)$$

In addition, the eigenvectors of $\mathbf{W}$ of $\tilde{\mathbf{A}}$ can be computed as the projection of the eigenvectors to the low-dimensional space as $\mathbf{W} = \tilde{\mathbf{U}}^*\Phi$. Then, the equivalence between the eigenvalues of $\mathbf{A}$ and $\tilde{\mathbf{A}}$ can be verified as

$$\tilde{\mathbf{A}}\mathbf{W} = \tilde{\mathbf{U}}^*\mathbf{S}\tilde{\mathbf{U}}^*\Phi = \tilde{\mathbf{U}}^*\mathbf{A}\Phi = \tilde{\mathbf{U}}^*\Phi\mathbf{\Lambda} = \mathbf{W}\mathbf{\Lambda}. \quad (22)$$

Remark 1: DMD modes preserve dynamic information in lower dimensions as observed in (22). Here, the dynamic model can be decomposed into dynamic modes Φ , where the eigenvalues $\mathbf{\Lambda}$ characterize the temporal nature of the associated dynamic modes Φ as shown in (20).

Remark 2: To ensure good estimation results in the linear approximation mapping $\mathbf{A}$, the collected data $\mathbf{X}$ and $\mathbf{X}'$ need to be rich enough or, equivalently, to fulfil a persistent of excitation condition [34], [35].

A. Dynamic Mode Decomposition with Control

In the previous section, the system (6) is used to model the autonomous system's dynamics as an unforced system where the control input is within the nonlinear function $\mathbf{H}(\mathbf{x}_k)$. In this scenario, the DMD algorithm is applied to obtain a local linear approximation of the autonomous system's dynamics which is encompassed by the matrix $\mathbf{A}$. However, this approximation lacks of generalization for different references. Therefore, an improvement of the DMD method known as DMD with control (DMDc) [36] was proposed to incorporate measurements of the control input sequences to enhance the generalization of the DMD model under different references and controllers.

Here, the local linear approximation (15) is modified as a linear control system of the form

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k. \quad (23)$$

In this case, we assume that $\mathbf{B} \in \mathbb{R}^{n \times p}$ is known such that the problem is slightly modified to the minimization of

$$e = \|\mathbf{x}_{k+1} - \mathbf{B}\mathbf{u}_k - \mathbf{A}\mathbf{x}_k\|_2, \quad (24)$$

across all points for $k = 1, \dots, \ell - 1$. An additional matrix is constructed from the control sequences $\mathbf{u}_k$ as

$$\mathbf{\Upsilon} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{\ell-1} \\ | & | & \cdots & | \end{bmatrix} \in \mathbb{R}^{m \times \ell-1}. \quad (25)$$

So, the optimization problem is equivalent to find the matrix $\mathbf{A}$ that minimizes the following error function

$$E = \|\mathbf{X}' - \mathbf{B}\mathbf{\Upsilon} - \mathbf{A}\mathbf{X}\|_{\mathcal{F}}. \quad (26)$$

The exact matrix $\mathbf{A}$ is obtained in a least-squares sense as

$$\begin{aligned} \mathbf{X}' - \mathbf{B}\mathbf{\Upsilon} &= \mathbf{A}\mathbf{X} \\ \mathbf{A} \approx \mathbf{A} &= (\mathbf{X}' - \mathbf{B}\mathbf{\Upsilon})\mathbf{X}^\dagger. \end{aligned} \quad (27)$$

A similar low-dimensional representation can be obtained by projecting the matrix $\mathbf{A}$ onto the principal subspace as

$$\begin{aligned} \tilde{\mathbf{x}}_{k+1} &= \tilde{\mathbf{U}}^*\mathbf{A}\tilde{\mathbf{U}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{U}}^*\mathbf{B}\mathbf{v}_k \\ &= \tilde{\mathbf{U}}^*(\mathbf{X}' - \mathbf{B}\mathbf{\Upsilon})\tilde{\mathbf{V}}\tilde{\Sigma}^{-1}\tilde{\mathbf{x}}_k + \tilde{\mathbf{U}}^*\mathbf{B}\mathbf{v}_k \\ &= \tilde{\mathbf{A}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{B}}\mathbf{v}_k, \end{aligned} \quad (28)$$

where

$$\begin{aligned} \tilde{\mathbf{A}} &= \tilde{\mathbf{U}}^*(\mathbf{X}' - \mathbf{B}\mathbf{\Upsilon})\tilde{\mathbf{V}}\tilde{\Sigma}^{-1} \\ \tilde{\mathbf{B}} &= \tilde{\mathbf{U}}^*\mathbf{B}, \end{aligned} \quad (29)$$

and the dynamic modes are computed by

$$\Phi = (\mathbf{X}' - \mathbf{B}\mathbf{\Upsilon})\tilde{\mathbf{V}}\tilde{\Sigma}^{-1}\mathbf{W}. \quad (30)$$

In this formulation the addition of the control input sequences $\mathbf{\Upsilon}$ gives useful information to distinguish between the open-loop dynamics associated to the matrix $\mathbf{A}$ and the imposed dynamics given by the controller $\mathbf{u}_k$.

B. Physics Informed Dynamic Mode Decomposition

The methods discussed in previous sections are completely data driven which does not consider some useful properties of the autonomous system. This section provides a different DMD formulation known as physics informed DMD (PIDMD) [37], where we associate a physical meaning to the linear approximation matrix $\mathbf{A}$.

The DMDc method uses prior knowledge of matrix $\mathbf{B}$ to compute the matrix $\mathbf{A}$ that minimizes the error function E . This prior knowledge defines a basic physics informed model associated to the input dynamics that gives a constraint to the possible solutions of matrix $\mathbf{A}$. In this section, we aim to add an additional constraint to the matrix $\mathbf{A}$ to provide a more robust and stable linear physical representation [38], [39].

Consider the second order dynamic model (1) and assume $\mathbf{u} = \mathbf{0}$. This model represents an unforced dynamic model which is obtained from the Euler-Lagrange (EL) formulation [40]. The key idea of this methodology is the construction of the energy functions associated to each conservative element that the system possesses. For the autonomous systems that we focus on, these energy functions are the kinetic energy $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}$ and potential energy $\mathcal{U}(\mathbf{q}) \in \mathbb{R}$ [41].

Notice that the unforced dynamics of the autonomous system is in fact a conservative system that verifies the energy conservation law

$$\mathcal{E}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{U}(\mathbf{q}) = \mathcal{C}, \quad (31)$$

where $\mathcal{C} \in \mathbb{R}$ is a constant scalar value. The DMD model assumes that the energy of the local linear representation is quadratic in terms of the state as $\mathcal{E}(\mathbf{x}) = \|\mathbf{x}\|_2^2$ [42]. So, the problem can be viewed in finding the matrix $\mathbf{A}$ that minimizes the error function e and ensures energy conservation. Thus, the energy is preserved if

$$\mathcal{E}(\mathbf{A}\mathbf{x}) = \|\mathbf{A}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2 = \mathcal{E}(\mathbf{x}). \quad (32)$$

This implies that the linear matrix $\mathbf{A}$ is orthonormal, i.e., $\mathbf{A}^\top \mathbf{A} = \mathbf{A}\mathbf{A}^\top = \mathbf{I}_n$ and its main function is to distribute the energy between the states and keep it constant. Therefore, the problem (8) is modified to an optimization problem with an orthonormal constraint as

$$\begin{aligned} \min_{\mathbf{A}} \quad & \|\mathbf{x}_{k+1} - \mathbf{A}\mathbf{x}\|_2^2, \\ \text{s.t.} \quad & \mathbf{A}\mathbf{A}^\top = \mathbf{I}_n, \end{aligned} \quad (33)$$

or equivalently

$$\begin{aligned} \min_{\mathbf{A}} \quad & \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_{\mathcal{F}}^2 = \text{tr}\{(\mathbf{X}' - \mathbf{A}\mathbf{X})^\top (\mathbf{X}' - \mathbf{A}\mathbf{X})\}, \\ \text{s.t.} \quad & \mathbf{A}\mathbf{A}^\top = \mathbf{I}_n. \end{aligned} \quad (34)$$

This problem is also known as the orthogonal Procrustes problem [43] whose solution is obtained by constructing the Lagrangian

$$\mathcal{L} = \text{tr}\{(\mathbf{X}' - \mathbf{A}\mathbf{X})^\top (\mathbf{X}' - \mathbf{A}\mathbf{X}) + \Lambda_{\mathbf{A}}(\mathbf{A}\mathbf{A}^\top - \mathbf{I}_n)\}, \quad (35)$$

where $\Lambda_{\mathbf{A}} > 0 \in \mathbb{R}^{n \times n}$ is a matrix of Lagrange multipliers. Differentiating the Lagrangian with respect to the matrix $\mathbf{A}$ and equalling to zero gives

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{A}} = & -(\mathbf{X}'\mathbf{X}^\top + \mathbf{X}'\mathbf{X}^\top) + \mathbf{A}(\mathbf{X}^\top \mathbf{X} + \mathbf{X}\mathbf{X}^\top) \\ & + (\Lambda_{\mathbf{A}} + \Lambda_{\mathbf{A}}^\top)\mathbf{A} = \mathbf{0}. \end{aligned}$$

Define $\mathbf{T} = \mathbf{X}'\mathbf{X}^\top$, $\mathbf{P} = \frac{1}{2}(\mathbf{X}^\top \mathbf{X} + \mathbf{X}\mathbf{X}^\top)$, and $\mathbf{Q} = \frac{1}{2}(\Lambda_{\mathbf{A}} + \Lambda_{\mathbf{A}}^\top)$. Then,

$$\mathbf{A}\mathbf{P} + \mathbf{Q}\mathbf{A} = \mathbf{T}. \quad (36)$$

If we post-multiply (36) by $\mathbf{A}^\top$ and using the orthonormal constraint $\mathbf{A}\mathbf{A}^\top = \mathbf{I}_n$ gives

$$\mathbf{T}\mathbf{A}^\top - \mathbf{A}\mathbf{P}\mathbf{A}^\top = \mathbf{Q}. \quad (37)$$

Notice that both $\mathbf{Q}$ and $\mathbf{A}\mathbf{P}\mathbf{A}^\top$ are symmetric matrices. The above equality holds if and only if

$$\mathbf{T}\mathbf{A}^\top = \mathbf{A}\mathbf{T}^\top.$$

Applying the constraint $\mathbf{A}^\top \mathbf{A} = \mathbf{I}$ in the above equality of symmetry gives

$$\begin{aligned} \mathbf{T} &= \mathbf{A}\mathbf{T}^\top \mathbf{A} \\ \mathbf{T}\mathbf{T}^\top &= \mathbf{A}\mathbf{T}^\top \mathbf{T}\mathbf{A}^\top. \end{aligned} \quad (38)$$

The above equality relates two symmetric matrices $\mathbf{T}\mathbf{T}^\top$ and $\mathbf{T}^\top \mathbf{T}$ which contains only the data collected from the autonomous system trajectories. We can apply an eigendecomposition to the above matrices which yields

$$\begin{aligned} \mathbf{T}\mathbf{T}^\top &= \mathbf{U}_{\mathbf{T}}\mathbf{\Lambda}_{\mathbf{T}}\mathbf{U}_{\mathbf{T}}^*, \\ \mathbf{T}^\top \mathbf{T} &= \mathbf{V}_{\mathbf{T}}\mathbf{\Lambda}_{\mathbf{T}}\mathbf{V}_{\mathbf{T}}^*, \end{aligned}$$

where $\mathbf{U}_{\mathbf{T}}, \mathbf{V}_{\mathbf{T}} \in \mathbb{R}^{n \times n}$ are the eigenvectors of matrices $\mathbf{T}\mathbf{T}^\top$ and $\mathbf{T}^\top \mathbf{T}$, respectively; both matrices share the same matrix of eigenvalues $\mathbf{\Lambda}_{\mathbf{T}}$. Both $\mathbf{U}_{\mathbf{T}}$ and $\mathbf{V}_{\mathbf{T}}$ are orthonormal, that is, $\mathbf{U}_{\mathbf{T}}^* \mathbf{U}_{\mathbf{T}} = \mathbf{U}_{\mathbf{T}} \mathbf{U}_{\mathbf{T}}^* = \mathbf{V}_{\mathbf{T}}^* \mathbf{V}_{\mathbf{T}} = \mathbf{V}_{\mathbf{T}} \mathbf{V}_{\mathbf{T}}^* = \mathbf{I}_n$. Substituting the eigendecomposition in (38) gives

$$\mathbf{U}_{\mathbf{T}}\mathbf{\Lambda}_{\mathbf{T}}\mathbf{U}_{\mathbf{T}}^* = \mathbf{A}\mathbf{V}_{\mathbf{T}}\mathbf{\Lambda}_{\mathbf{T}}\mathbf{V}_{\mathbf{T}}^*\mathbf{A}^\top.$$

Therefore, the orthonormal matrix $\mathbf{A}$ that minimizes the optimization problem and preserve the energy is

$$\mathbf{A} = \mathbf{U}_{\mathbf{T}}\mathbf{V}_{\mathbf{T}}^*. \quad (39)$$

The matrix (39) defines the orthogonal linear approximation of the unforced system. We can obtain a low-dimensional representation of matrix $\mathbf{A}$ by first obtaining the singular vectors $\tilde{\mathbf{U}}_{\mathbf{X}} \in \mathbb{C}^{n \times r}$ of $\mathbf{X}$ to project both $\mathbf{X}$ and $\mathbf{X}'$ to the principal subspace. Let denote the projected matrices as $\tilde{\mathbf{X}} \in \mathbb{R}^{r \times \iota-1}$ and $\tilde{\mathbf{X}}' \in \mathbb{R}^{r \times \iota-1}$, then we can construct the matrix $\tilde{\mathbf{T}} = \tilde{\mathbf{X}}'\tilde{\mathbf{X}}^\top \in \mathbb{R}^{r \times r}$. The SVD method is applied in $\tilde{\mathbf{T}}$ to obtain the new singular matrices $\tilde{\mathbf{U}}_{\mathbf{T}} \in \mathbb{R}^{r \times r}$ and $\tilde{\mathbf{V}}_{\mathbf{T}} \in \mathbb{R}^{r \times r}$. Hence, the low-dimensional orthogonal matrix is

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}}_{\mathbf{T}}\tilde{\mathbf{V}}_{\mathbf{T}}^* \in \mathbb{R}^{r \times r}. \quad (40)$$

The linear representation can be recovered as

$$\begin{aligned} \mathbf{x}_{k+1} &= \tilde{\mathbf{U}}_{\mathbf{X}}\mathcal{A}\tilde{\mathbf{U}}_{\mathbf{X}}^*\mathbf{x}_k \\ &= \tilde{\mathbf{U}}_{\mathbf{X}}\tilde{\mathbf{U}}_{\mathbf{T}}\tilde{\mathbf{V}}_{\mathbf{T}}^*\tilde{\mathbf{U}}_{\mathbf{X}}^*\mathbf{x}_k \\ &\approx \mathcal{A}\mathbf{x}_k, \end{aligned} \quad (41)$$

where $\mathbf{A} \approx \mathcal{A} = \tilde{\mathbf{U}}_{\mathbf{X}}\tilde{\mathbf{U}}_{\mathbf{T}}\tilde{\mathbf{V}}_{\mathbf{T}}^*\tilde{\mathbf{U}}_{\mathbf{X}}^* \in \mathbb{R}^{n \times n}$.

To obtain the dynamical properties of the orthogonal representation we need to follow a similar procedure to the traditional DMD method, that is, the eigenvalues of $\tilde{\mathbf{A}}$ verifies

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}.$$

The dynamic modes Φ of $\mathbf{A}$ require that

$$\mathbf{A}\Phi = \tilde{\mathbf{U}}_{\mathbf{X}}\tilde{\mathbf{U}}_{\mathbf{T}}\tilde{\mathbf{V}}_{\mathbf{T}}^*\tilde{\mathbf{U}}_{\mathbf{X}}^*\Phi = \Phi\mathbf{\Lambda},$$

so, the dynamic modes Φ of $\mathbf{A}$ are computed as

$$\Phi = \tilde{\mathbf{U}}_{\mathbf{X}}\mathbf{W}, \quad (42)$$

and the eigenvectors $\mathbf{W}$ of $\mathbf{A}$ are computed as $\mathbf{W} = \tilde{\mathbf{U}}_{\mathbf{X}}^*\Phi$ such that

$$\tilde{\mathbf{A}}\mathbf{W} = \tilde{\mathbf{U}}_{\mathbf{T}}\tilde{\mathbf{V}}_{\mathbf{T}}^*\tilde{\mathbf{U}}_{\mathbf{X}}^*\Phi = \tilde{\mathbf{U}}_{\mathbf{X}}^*\mathbf{A}\Phi = \tilde{\mathbf{U}}_{\mathbf{X}}^*\Phi\mathbf{\Lambda} = \mathbf{W}\mathbf{\Lambda}.$$

For the forced case, the PIDMD methodology cannot be used since the control input, in most cases, adds dissipative terms that violates the energy conservation law.

Remark 3: This approach is sensitive to the potential energy function $\mathcal{U}(\mathbf{q})$ because it does not appear in some autonomous system's models or contains partial terms of the state $\mathbf{x}$, e.g., unmanned autonomous vehicles (UAVs). In addition, $\mathcal{U}(\mathbf{q})$ cannot be written as a quadratic function using a symmetric matrix, e.g., the gravitational terms; in consequence it hinders the design of the orthonormal matrix $\mathbf{A}$.

IV. TRAJECTORY PREDICTION

The trajectory prediction module is based on the local linear approximation model obtained from any of the aforementioned DMD algorithms. Two different trajectory prediction algorithms are identified in terms of the structure of the linear model: control-free and control based which are discussed in the following sections.

A. Control-free prediction algorithm

This is the simple prediction algorithm technique which is based on the classic DMD linear model. Here, we can easily compute N time steps in the future by noticing that

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{A}\mathbf{x}_0 \\ \mathbf{x}_2 &= \mathbf{A}\mathbf{x}_1 = \mathbf{A}^2\mathbf{x}_0 \\ &\vdots \\ \mathbf{x}_N &= \mathbf{A}\mathbf{x}_{N-1} = \mathbf{A}^N\mathbf{x}_0. \end{aligned}$$

Therefore, the prediction algorithm is computed as

$$\mathbf{x}_i = \mathbf{A}^i\mathbf{x}_0, \quad i = 1, \dots, N. \quad (43)$$

Algorithm 1 summarizes the pseudo-code of the trajectory prediction algorithm based on the classic DMD method.

Algorithm 1 Dynamic Mode Decomposition for Trajectory Prediction

Input: Number of dimensions for projection r , State measurements $\mathbf{x}_k$.

- 1: Collect $\iota \geq n^2 + 1$ samples and construct the matrices $\mathbf{X}, \mathbf{X}'$.
- 2: Apply the classic DMD (14) or (17) to obtain $\mathbf{A}$ or $\tilde{\mathbf{A}}$, respectively.
- 3: Predict the trajectory $\mathbf{x}_N^{pred}$ in N steps using (43).

Output: $\mathbf{x}_N^{pred}$.

The main drawback of this technique is that matrix $\mathbf{A}$ contains within the hidden destination $\mathbf{x}^d \in \mathbb{R}^n$ and hence, we cannot generalize for different references. To consider new destinations, we need to collect again data and obtain a local linear model using the DMD method. However, this is useful for iterative trajectories, e.g., sinusoidal trajectories.

B. Control-based prediction algorithm

The matrix $\mathbf{A}$ obtained from the DMDc method is based on the measurements of $\mathbf{x}_k$ and $\mathbf{u}_k$ which contains all the non-linearities of the autonomous system model. Assume that the control input in (4) of the autonomous system has a linear structure of the form

$$\mathbf{u}_k = \mathbf{K}(\mathbf{x}^d - \mathbf{x}_k), \quad (44)$$

where $\mathbf{K} \in \mathbb{R}^{p \times n}$ is an unknown stabilizing gain and $\mathbf{x}^d \in \mathbb{R}^n$ is the hidden reference. Here, $\mathbf{u}_k$ is a stabilizing controller that ensures the reference tracking.

Remark 4: The linear controller assumption is not always met in different real-world implementations. However, we can

always compute a linear stabilizing controller from the matrices $\mathbf{A}$ and $\mathbf{B}$ that approximates the closed-loop performance of the real system and apply the proposed prediction algorithm.

Consider the following one-step prediction equations

$$\begin{aligned} \mathbf{u}_k &= -\mathbf{K}\mathbf{x}_k + \mathbf{K}\mathbf{x}^d, \\ \mathbf{u}_{k+1} &= -\mathbf{K}\mathbf{x}_{k+1} + \mathbf{K}\mathbf{x}^d. \end{aligned} \quad (45)$$

The difference between the above equations yields $\mathbf{u}_{k+1} - \mathbf{u}_k = -\mathbf{K}(\mathbf{x}_{k+1} - \mathbf{x}_k)$. Define $\mathbf{y}_k := \mathbf{u}_{k+1} - \mathbf{u}_k$ and $\mathbf{z}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$. Then we can construct two new matrices from the collected data

$$\begin{aligned} \mathbf{Y} &= \begin{bmatrix} | & | & & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_{\kappa-1} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times \kappa-1} \\ \mathbf{Z} &= \begin{bmatrix} | & | & & | \\ \mathbf{z}_1 & \mathbf{z}_2 & \cdots & \mathbf{z}_{\kappa-1} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times \kappa-1}, \end{aligned} \quad (46)$$

to form the next system of equations $\mathbf{Y} = -\mathbf{K}\mathbf{Z}$. So, we can obtain $\mathbf{K}$ using a batch least squares formulation, that is,

$$\mathbf{K} = -\mathbf{Y}\mathbf{Z}^\dagger. \quad (47)$$

Since the reference $\mathbf{x}^d$ is within the control input (44), then for any pair $\{\mathbf{x}_k, \mathbf{u}_k\}$ we have that

$$\mathbf{K}\mathbf{x}^d = \mathbf{u}_k + \mathbf{K}\mathbf{x}_k. \quad (48)$$

So, following a similar procedure to the control-free prediction algorithm we obtain

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{A}\mathbf{x}_0 + \mathbf{B}\mathbf{K}(\mathbf{x}^d - \mathbf{x}_0) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}_0 + \mathbf{B}\mathbf{K}\mathbf{x}^d \\ \mathbf{x}_2 &= \mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{K}(\mathbf{x}^d - \mathbf{x}_1) = (\mathbf{A} - \mathbf{B}\mathbf{K})^2\mathbf{x}_0 \\ &\quad + (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{B}\mathbf{K}\mathbf{x}^d + \mathbf{B}\mathbf{K}\mathbf{x}^d \\ &\vdots \\ \mathbf{x}_N &= \mathbf{A}\mathbf{x}_{N-1} + \mathbf{B}\mathbf{K}(\mathbf{x}^d - \mathbf{x}_{N-1}) = (\mathbf{A} - \mathbf{B}\mathbf{K})^N\mathbf{x}_0 \\ &\quad + \left[(\mathbf{A} - \mathbf{B}\mathbf{K})^{N-1} + \cdots + (\mathbf{A} - \mathbf{B}\mathbf{K}) + \mathbf{I}_n \right] \mathbf{B}\mathbf{K}\mathbf{x}^d. \end{aligned}$$

Therefore, the prediction is given by

$$\mathbf{x}_i^{pred} = (\mathbf{A} - \mathbf{B}\mathbf{K})^i\mathbf{x}_0 + \left(\sum_{j=1}^i (\mathbf{A} - \mathbf{B}\mathbf{K})^{j-1} \right) \mathbf{B}\mathbf{K}\mathbf{x}^d. \quad (49)$$

Notice that the proposed trajectory prediction algorithm is valid for a single reference point. However, if there are multiple points then we need to compute $\mathbf{K}\mathbf{x}^d$ each time the point changes. New points can be considered each time that $\|\mathbf{B}\mathbf{x}_{j+1}^d - \mathbf{B}\mathbf{x}_j^d\| \neq 0$. This condition implies that when the autonomous system reaches to the desired point, then we need to measure the states and control input to determine if the destination changes and predict the number of time steps that the autonomous system will require to reach the new point. For this purpose we need to check if $\|\mathbf{B}\mathbf{K}\mathbf{x}_k - \mathbf{B}\mathbf{K}\mathbf{x}^d\| \leq \varepsilon$ for a small $\varepsilon > 0$. This procedure is applied iteratively until the experiment concludes. Algorithm 2 shows the pseudo-code of the proposed control-based trajectory prediction algorithm.

Standard prediction algorithms aim to predict the future trajectory in accordance with the trajectory history of the autonomous system. The proposed approach, in contrast, uses the

Algorithm 2 Physics Informed Dynamic Mode Decomposition for Trajectory Prediction

Input: Matrix $\mathbf{B}$, state measurements $\mathbf{x}_k$, control input measurements $\mathbf{u}_k$, number of dimensions for projection r , length of the experiment t , threshold ε .

- 1: Collect $\iota \geq n^2 + 1$ samples and construct the matrices $\mathbf{X}, \mathbf{X}', \mathbf{Y}, \mathbf{Y}$, and $\mathbf{Z}$.
- 2: Apply either the DMDc (27) or PIDMD (39) to obtain $\mathbf{A}$.
- 3: Compute the control gain $\mathbf{K}$ using (47).
- 4: **while** $k \leq t$ **do**
- 5: Measure $\mathbf{x}_k$ and $\mathbf{u}_k$. Compute $\mathbf{K}\mathbf{x}^d$ using (48).
- 6: Set $i = 0$
- 7: **while** $\|\mathbf{K}\mathbf{x}_k - \mathbf{K}\mathbf{x}^d\| \leq \varepsilon$ **do**
- 8: Predict the trajectory $\mathbf{x}_i^{pred}$ in i steps using (49)
- 9: $i \leftarrow i + 1$
- 10: **end while**
- 11: **end while**

Output: $\mathbf{x}_i^{pred}$, steps i .

linear model obtained from the DMD method and control input information to predict the time that the autonomous system requires to converge to the desired reference. Therefore, the prediction duration can be arbitrarily large since the reference is constant and does not change over time.

Algorithm 2 is valid when the reference is constant. For time-varying references we assume the control gain $\mathbf{K}$ is known, such that we can estimate the desired reference in each iteration i . This defines one of the main disadvantages of the proposed approach due to its small prediction time window for time-varying references. Future research will analyse the use of model predictive control as a potential tool to alleviate this issue.

Remark 5: For Algorithm 1 and Algorithm 2, the involved computation is constituted by two main elements: the estimation of matrix $\mathbf{A}$ and the prediction algorithm in N steps. If one does the computation of the complete DMD or DMDc, the computational complexity has growth $n(\iota - 1)^2$, and for the predictive model (43) or (49) grows exponentially in n each step. Thus, the complexity of the overall algorithm in N steps is $\mathcal{O}(n^{2+N} + n(\iota - 1)^2)$ with the term n^{2+N} dominating for large N . Here, the computational complexity is consistent with classical decomposition methods such as SVD and iterative learning control [44].

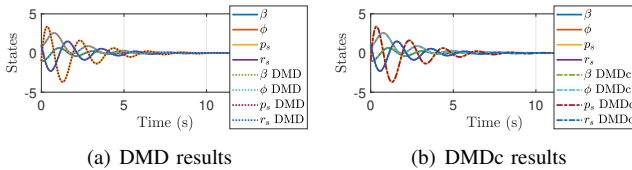


Fig. 2. DMD approximation of the F-16 aircraft dynamics

V. SIMULATION STUDIES

Several mechanical systems are tested to verify the proposed methodology and provide some insights of its limitations.

A. F-16 Aircraft dynamics

Consider a linear discrete-time F-16 aircraft model [45] defined by the following matrices

$$\mathbf{A} = \begin{bmatrix} 0.9963 & 0.0006 & 0.0004 & -0.0099 \\ -0.0015 & 1 & 0.0098 & 0.0004 \\ -0.3001 & -0.0001 & 0.9638 & 0.008 \\ 0.0851 & 0 & -0.0002 & 0.9948 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -0.0073 & 0.0013 \\ -0.0003 & -0.0006 \end{bmatrix}.$$

These matrices correspond to a continuous F-16 model sampled each $T = 0.01$ s. Assume the F-16 aircraft model is controlled by a discrete-time linear quadratic regulator (LQR) [46] whose gain is

$$\mathbf{K} = \begin{bmatrix} 5.9732 & -2.9021 & -1.4979 & -3.1341 \\ -0.1580 & -0.0681 & -0.0873 & -1.7660 \end{bmatrix}.$$

In this example, we aim to obtain the linear matrices and observe the qualities of the DMD method for system identification and prediction. Here, we only compare the DMD and DMDc methods since we don't have a physical relationship of the F-16 aircraft states $\mathbf{x} = [\beta, \phi, p_s, r_s]^T$, where β is the sideslip, ϕ is the roll angle, p_s and r_s define the stability axis roll and yaw angles. Data are collected in the first 2 seconds of the simulation to construct matrices $\mathbf{X}, \mathbf{X}'$, and $\mathbf{Y}$ to apply the DMD and the DMDc methods. The estimated matrices are

$$\mathbf{A}_{\text{DMD}} = \begin{bmatrix} 0.9963 & 0.0006 & 0.0004 & -0.0099 \\ -0.0015 & 1.0000 & 0.0098 & 0.0004 \\ -0.2942 & -0.0064 & 0.9615 & 0.0015 \\ 0.0853 & -0.0003 & -0.0003 & 0.9943 \end{bmatrix},$$

$$\mathbf{A}_{\text{DMDc}} = \begin{bmatrix} 0.9963 & 0.0006 & 0.0004 & -0.0099 \\ -0.0015 & 1 & 0.0098 & 0.0004 \\ -0.3001 & -0.0001 & 0.9638 & 0.008 \\ 0.0851 & 0 & -0.0002 & 0.9948 \end{bmatrix}.$$

Notice that the obtained matrices $\mathbf{A}_{\text{DMD}}$ and $\mathbf{A}_{\text{DMDc}}$ are different. The matrix obtained from the DMD method is in fact the closed-loop matrix between the real matrix $\mathbf{A}$ and the dynamics given by the controller $\mathbf{BK}$, that is, $\mathbf{A}_{\text{DMD}} \approx \mathbf{A} - \mathbf{BK}$. On the other hand, the DMDc estimates accurately the matrix $\mathbf{A}$. Here both the control-free and control-based methods are used for the trajectory prediction. Here, the desired reference is the origin which is a straightforward reference that both methods can achieve accurately using the estimated matrices $\mathbf{A}_{\text{DMD}}$ and $\mathbf{A}_{\text{DMDc}}$.

Fig. 2 shows the trajectory prediction results using the DMD and DMDc methods. Since the F-16 model is linear and noise-free then an exact representation of its dynamics is obtained and therefore, the estimated trajectories matches with the real ones. Algorithm 2 is applied to predict the future time step where the trajectories converge to the desired reference. In this example, we assume convergence when the norm $\|\mathbf{x}_k - \mathbf{x}^d\|_2 \leq 0.01$. Both the DMD and the DMDc converge to the desired reference $\mathbf{x}^d$ in 12.06 seconds. One interesting conclusion obtained from this experiment is that the prediction time can be modified by increasing or decreasing

the threshold, e.g., for a threshold of 0.001, the prediction time is 17.16 seconds. Despite a small threshold gives more confidence in the accuracy of the prediction, the prediction time is overestimated which can be dangerous in counter system applications.

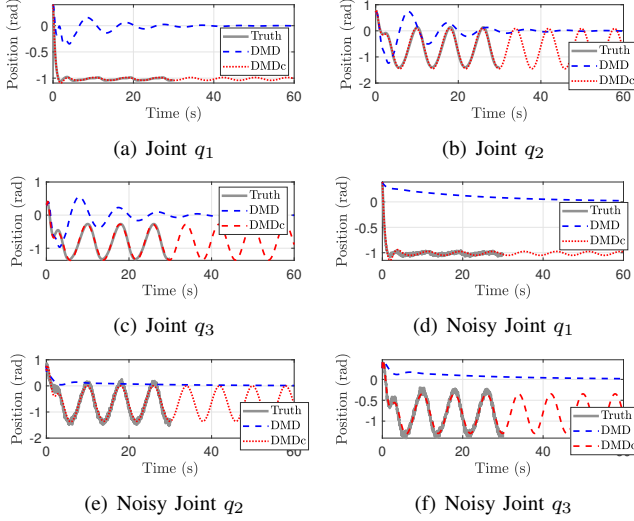


Fig. 3. Trajectory Prediction of the robot dynamics

It is important to clarify the following points: i) DMD method incorporates the desired reference in the estimation of the matrix $\mathbf{A}_{\text{DMD}}$ which is useful for stabilization tasks but not suitable for time-varying references, ii) DMD works well in stable open-loop systems where the matrix $\mathbf{A}_{\text{DMD}}$ captures the information associated to the real autonomous system dynamics; on the other hand, in closed-loop systems the estimation performance can be degraded by the desired reference.

B. 3-DOF Planar Robot

Consider a 3-DOF planar robot proposed in [47]. The state vector is defined by the position and velocities of each joint respectively, i.e., $\mathbf{x} = [q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3]^\top$. This model is high-nonlinear and it will test the representation capabilities of DMD methods. The PIDMD cannot be used for this approach since the controller incorporates dissipative terms which violates the energy conservation law. In addition, the open-loop dynamics of this particular robot is unstable and hence, cannot be used for identification purposes. The controller $\mathbf{u}$ is designed to ensure stable reference tracking of the robot trajectories. The data trajectories are sampled each $T = 0.03\text{s}$ to construct the matrices $\mathbf{X}, \mathbf{X}'$. The matrix $\mathbf{B}$ is computed as

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_3 \\ \left(T \max_{\mathbf{q}} \mathbf{M}(\mathbf{q}) \right)^{-1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.03 & -0.02 & 0 \\ -0.02 & 0.022 & -0.007 \\ 0 & -0.007 & 0.009 \end{bmatrix}.$$

We collect the first 20 seconds of the trajectories to obtain the linear matrix $\mathbf{A}$. Here, the robot is following a time-varying reference which implies that the matrix gain $\mathbf{K}$ cannot

be computed using (47). So, for this example we assume knowledge of the control gain $\mathbf{K}$ for prediction purposes. The control gain is $\mathbf{K} = [\text{diag}\{100, 50, 30\}, \text{diag}\{50, 30, 20\}]$.

Fig. 3 reports the prediction results of the planar robot for the next 30 seconds. Two scenarios are tested: 1) measurements without noise and 2) measurements with Gaussian noise. Notice that the DMD method has poor estimation and prediction performance in both scenarios because the robot is following a time-varying reference whose dynamics is not well represented by the linear matrix approximation. When the Gaussian noise is incorporated into the trajectories, the DMD method obtains an underdamped linear system with complex eigenvalues that cannot capture the real behaviour of the robot. In other words, the linear approximation of the DMD loses information due to the negative real part of its eigenvalues. On the other hand, the DMDc shows accurate inference results with noise attenuation capabilities by incorporating the control input information. The error norm $\|\tilde{\mathbf{q}}\| = \|\mathbf{q} - \mathbf{q}_i\|$ is used as performance metric, where $\mathbf{q} = [q_1, q_2, q_3]^\top$ and $\mathbf{q}_i$ is the estimated joint angle of method $i = \{\text{DMD}, \text{DMDc}\}$. The results are summarized in Table I where it is clear that the standard DMD method suffers to represent the robot trajectories. Conversely, the incorporation of the control part enhances the prediction capabilities.

TABLE I
ERROR NORM $\|\tilde{\mathbf{q}}\|$ OF THE ROBOT JOINTS. BEST RESULTS ARE IN BOLD

Error Norm	$\ \tilde{\mathbf{q}}_1\ $		$\ \tilde{\mathbf{q}}_2\ $		$\ \tilde{\mathbf{q}}_3\ $	
	DMD	DMDc	DMD	DMDc	DMD	DMDc
Without noise	0.0307	3.9364e^{-4}	0.0273	0.0017	0.0273	0.0013
With noise	0.037	0.0011	0.029	0.0038	0.0314	0.0026

C. Drone's Attitude Dynamics

Consider the attitude dynamics of a drone [48], [49] given by the following matrices

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} I_x & 0 & -I_x s_\theta \\ 0 & I_y c_\phi^2 + I_z s_\phi^2 & (I_y - I_z) c_\phi s_\phi c_\theta \\ -I_x s_\theta & (I_y - I_z) c_\phi s_\phi c_\theta & I_x s_\theta^2 c_\theta^2 + I_z c_\phi^2 c_\theta^2 \end{bmatrix},$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^\top \frac{\partial \mathbf{M}(\mathbf{q})}{\partial \mathbf{q}}$$

where $\mathbf{q} = [\phi, \theta, \psi]^\top$ stand to the roll, pitch, and yaw angles. The state vector is $\mathbf{x} = [\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^\top$. I_x, I_y, I_z define the inertia terms, s_j and c_j refer to the $\sin(j)$ and $\cos(j)$ functions with $j = \phi, \theta, \psi$. For this model we cannot use the PIDMD because there is no potential energy that associates the roll, pitch, and yaw angles.

The matrix $\mathbf{B}$ is computed with a sample period of $T = 0.01$ and is given by

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0.4633 & 0 & -0.4633 \\ 0.4633 & 0 & -0.4633 & 0 \\ -0.0013 & 0.0013 & -0.0013 & 0.0013 \end{bmatrix}.$$

Data is collected from the first 15 seconds of the experiment to construct the matrices $\mathbf{X}$ and $\mathbf{X}'$. Here, the collected data do not contain noise. The drone follows a time-varying reference and hence we assume knowledge of the control gain which is given by

$$\mathbf{K} \approx \begin{bmatrix} 0 & 0.51 & -0.498 & 0 & 0.52 & -1.47 \\ 0.51 & 0 & 0.498 & 0.52 & 0 & 1.47 \\ 0 & -0.51 & -0.498 & 0 & -0.52 & -1.47 \\ -0.51 & 0 & 0.498 & -0.52 & 0 & 1.47 \end{bmatrix}.$$

Fig. 4 shows the trajectory prediction results for the next 20 seconds. In this experiment, the DMD cannot capture adequately the dynamics of the attitude dynamics despite the absence of noise in the data. This is because two main factors: i) the collected data does not have enough richness and ii) the signals are independent to each other causing deviation in the values of the estimated matrix $\mathbf{A}$. The richness problem is usually addressed by fulfilling a persistent of excitation (PE) condition [34]. On the other hand, the DMDc is able to predict accurately the trajectory of the drone similarly to the 3-DOF robot case. Here, the incorporation of the control information helps to couple the dynamics to improve the prediction capabilities of the DMD model. It is important to mention that for highly variable references, then this approach will not be suitable due to the data sampling time. This issue is topic for future work.

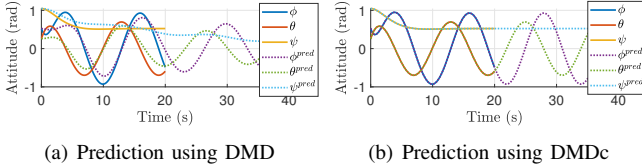


Fig. 4. Trajectory Prediction of the drone's attitude dynamics

Table II summarizes the norm of the prediction error. The DMD exhibits large error due to the lack on complex conjugates eigenvalues. The DMDc method shows small prediction error due to the incorporation of the control input that adds a constraint in the predictive manifold.

TABLE II
ERROR NORM $\|\mathbf{q}\|$ OF THE DRONE ATTITUDE STATES. BEST RESULTS ARE IN BOLD.

$\ \phi\ $		$\ \theta\ $		$\ \psi\ $	
DMD	DMDc	DMD	DMDc	DMD	DMDc
0.0043	1.7738e⁻¹⁵	0.0039	1.2095e⁻¹⁵	0.0022	4.2826e⁻¹⁵

D. Conservative System

Consider a one link robot model [50] without friction defined by

$$(ml^2 + J)\ddot{q} + mgl \sin(q) = 0,$$

where J is the inertia of the link, m is the mass, l is the length of the link, and q is the joint angle. This model is an ideal representation of the dynamics of the robot that verifies the energy conservation law. The energy of the system is

$$\mathcal{E}(q, \dot{q}) = \frac{1}{2}(ml^2 + J)\dot{q}^2 - mgl \cos(q),$$

which verifies $\frac{d}{dt}\mathcal{E}(q, \dot{q}) = 0$. In this example we compare the performance of the classic DMD (the DMDc is the same as the DMD for this case) with the PIDMD under noise-free and noisy measurements. Fig. 5 shows the results of the reconstruction. It can be observed that the PIDMD has good results in both scenarios with noise attenuation capabilities. Here, the addition of the energy of the system as a constraint regularizes the learning manifold of the DMD method. This helps to capture the dynamic properties of the robot such that the noise is neglected in the estimation. In contrast, the classic DMD is sensitive to the noise and exhibits a similar performance to the 3-DOF robot, that is, an underdamped linear system is obtained that cannot capture the open-loop behaviour of the robot.

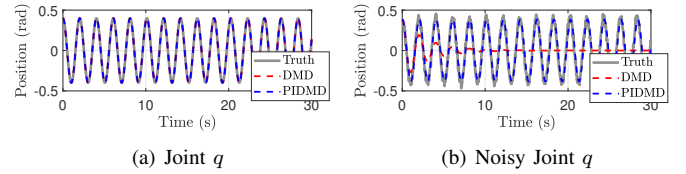


Fig. 5. Dynamics Reconstruction using DMD and PIDMD

One issue observed in the implementation is the definition of the potential energy. In the previous formulation, the joint angle is measured with respect to the negative y -axis, such that $\mathcal{U}(q) = -mgl \cos(q)$. However, if we measure the joint angle respect to the x -axis then the potential energy is defined by $\mathcal{U} = mgl \sin(q)$ and thus, the dynamics is modified to

$$(ml^2 + J)\ddot{q} + mgl \cos(q) = 0,$$

that also verifies $\frac{d}{dt}\mathcal{E}(q, \dot{q}) = 0$. The reconstruction results under this model are shown in Fig. 6. Notice that both the DMD and PIDMD cannot obtain good results. When the joint angle is measured along the x -axis then the robot natural dynamics becomes unstable since the origin is not a stable attractor point. Conversely, when the joint angle is measured along the y -axis then the trajectories of the robot near the origin are completely stable and useful for both the DMD and PIDMD methods.

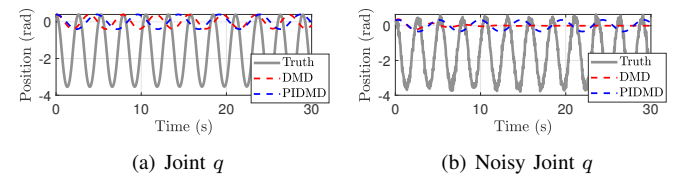


Fig. 6. Dynamics Reconstruction under different potential energy

Table III summarizes the norm of the prediction error of the DMD and the PIDMD methods. We can observe that the definition of the potential energy plays a major role in the results of the DMD and the PIDMD methods. By using an adequate definition of the potential energy, then the PIDMD outperforms the DMD method with a more stable and robust performance.

In terms of prediction the PIDMD method can be used in a three steps procedure, that is, first the linear approximation

matrix $\mathbf{A}$ is obtained under a zero control input. Then, the control input is applied to the system to compute the control gain $\mathbf{K}$, and finally make the prediction in N steps using (49). However, many autonomous systems do not present dynamics under zero control input, e.g., UAVs. Furthermore, real autonomous systems have many dissipative terms related to friction, damping, heat, etc., that hinders the application of the PIDMD.

TABLE III
ERROR NORM $\|\tilde{\mathbf{q}}\|$ OF THE ONE LINK ROBOT. BEST RESULTS ARE IN BOLD.

$\ \tilde{\mathbf{q}}\ $	$\mathcal{U} = -mgl \cos(q)$		$\mathcal{U} = mgl \sin(q)$	
	DMD	PIDMD	DMD	PIDMD
Without noise	1.1551e⁻⁵	1.6875e ⁻⁵	0.0672	0.0673
With noise	0.0083	0.0017	0.067	0.0674

E. Discussions

The results obtained from the previous cases show the effectiveness of the DMD methods under different scenarios for trajectory prediction. We observe that the DMDc has better results than the classic DMD in presence of noise or time-varying references. This is achieved by separating the control input from the states measurements. The prediction algorithm is effective under constant references, however if we deal with time-varying references we need to know beforehand the control gain $\mathbf{K}$. PIDMD is a promising architecture that exploits the physical laws of the autonomous system to obtain a linear approximation of the nonlinear dynamics. This allows to capture the dynamic properties of the system and regularize the learning manifold of the DMD for noise attenuation. However, the previous results show that is highly sensitive to: i) dissipative terms, ii) potential energy definition, and iii) control inputs. Further work will focus on the design of PIDMD that considers different energy functions that can effectively incorporate the control input and/or dissipative terms to obtain a more realistic linear mapping $\mathbf{A}$. The proposed methods under different sampling times show that the models are robust. However, it is show that the richness of the data plays a crucial role in the estimation of the linear model.

VI. EXPERIMENTAL VALIDATION

We further test the proposed approach using a personal UAV with mass $m = 0.5$ kg. We use the nonlinear backstepping method proposed in [51] to control the UAV. The control algorithm is designed in Matlab with an interface with Beagle-Bone-Blue (BBB) chip processor. In this experiment the states are defined by the linear positions and velocities of the UAV, whilst the control input is given by the Euler angles roll and pitch, and the total thrust. The VICON camera system, composed of 25 well-distributed cameras with different resolutions, is used to track the position of the drone. The VICON measurements and transmitting frequency is 120 Hz with a localization error of 0.01 m. Here the measurement noise is small such that the linear system estimation is not compromised. However, we still need to verify that the data

is rich enough and diverse to avoid the issues observed in the drone's attitude model.

The collected data belongs from a helix trajectory to ensure enough richness of the training data. We use the DMDc method to obtain a linear model of the drone and test it under unseen trajectories of way points to demonstrate its prediction capabilities. The following linear matrix $\mathbf{B}$, obtained around the hover flight condition, is used

$$\mathbf{B} = \begin{bmatrix} 0 & -0.002 & 0 \\ 0.002 & 0 & 0 \\ 0 & 0 & 0.0004 \\ 0 & -0.1962 & 0 \\ 0.1962 & 0 & 0 \\ 0 & 0 & 0.0428 \end{bmatrix}.$$

The estimated linear matrix $\mathbf{A}$ is

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0.02 & 0 & -2e^{-4} \\ 0 & 1 & 0 & 0 & 0.02 & 2e^{-4} \\ 0 & 0 & 1 & 1e^{-4} & 0 & 0.0192 \\ 1e^{-4} & 1e^{-4} & -2e^{-4} & 0.9972 & -6e^{-4} & 0.0159 \\ 1e^{-4} & 1e^{-4} & -2e^{-4} & -0.0021 & 0.9991 & 0.0143 \\ 0 & 0 & 0 & 0.0033 & 0.0011 & 0.9758 \end{bmatrix}.$$

In this particular model, we observe that the gain $\mathbf{K}$ cannot be computed using (47) since the estimated gain creates a linear combination between the linear positions and velocities of the UAV. This is caused due to the lack of relations between them in the real UAV dynamics. Hence, in this experiment we assume knowledge of the matrix $\mathbf{K}$ given by

$$\mathbf{K} = \begin{bmatrix} 0 & 0.447 & 0 & 0 & 1.045 & 0 \\ -0.447 & 0 & 0 & -1.045 & 0 & 0 \\ 0 & 0 & 0.447 & 0 & 0 & 1.191 \end{bmatrix}.$$

Two point-to-point trajectories are tested: a square trajectory and a trapezoidal trajectory. In these trajectories we change the desired reference point each 30 seconds. Each experiment lasts 2 minutes. Fig. 7 shows the trajectory prediction results.

Here, we observe that the estimated linear model is able to infer accurately the trajectory of both testing trajectories. For the square trajectory, the norm of the prediction error per axis are: $\|\tilde{\mathbf{x}}\| = 5.6069e^{-5}$, $\|\tilde{\mathbf{y}}\| = 7.0035e^{-5}$ and $\|\tilde{\mathbf{z}}\| = 3.4655e^{-4}$, which clearly demonstrates the accuracy of the linear model predictor. In terms of the prediction time, we use a threshold of $\varepsilon = 0.01$. The convergence time of the UAV into each way point are: $t_1 = 12.94$ s, $t_2 = 41.54$ s, $t_3 = 72.52$ s, and $t_4 = 101.32$ s. For the trapezoidal trajectory, the norm of the prediction error per axis are: $\|\tilde{\mathbf{x}}\| = 5.375e^{-5}$, $\|\tilde{\mathbf{y}}\| = 7.8973e^{-5}$, and $\|\tilde{\mathbf{z}}\| = 4.8180e^{-4}$. The convergence time of the UAV into each way point are $t_1 = 13.04$ s, $t_2 = 40.08$ s, $t_3 = 73.1$ s and $t_4 = 101.32$ s. Here, we can observe that the prediction times are consistent with the results of Fig. 7.

It is important to mention that the prediction time can be further improved by improving the sampling time of the data collection. This impact in the accuracy of the model and the selection of an adequate threshold.

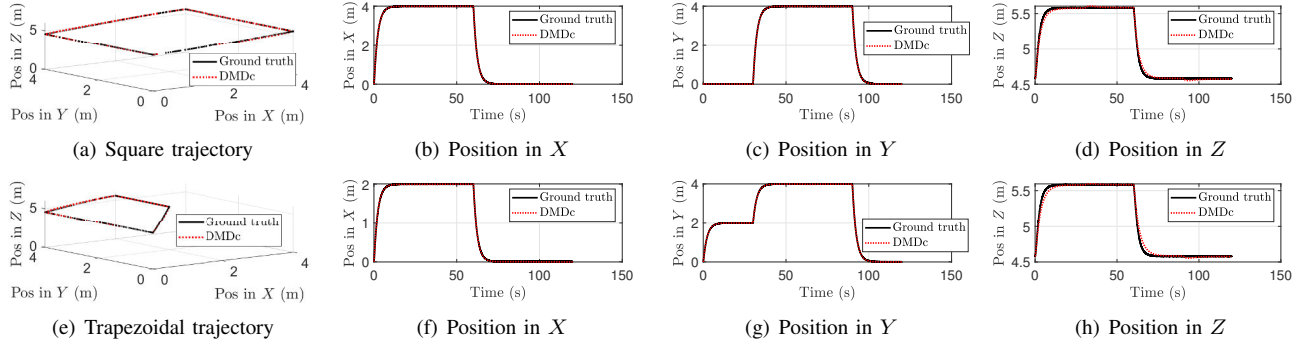


Fig. 7. UAV trajectory prediction. First row shows the results of the square trajectory. Second row shows the results of the trapezoidal trajectory.

VII. CONCLUSIONS

This paper reports a trajectory intent prediction algorithm based on dynamic mode decomposition techniques. Three algorithms are used in the development of the approach: classic DMD, DMD with control (DMDc), and physics informed DMD (PIDMD). When the control input is available, then the gain and reference can be extracted using a simple batch-least squares rule algorithm which helps to improve the accuracy of the prediction. Simulation studies inform that classic DMD is not enough to predict future trajectories due to the complex nature of the intended reference. The DMDc shows good prediction results by using the control input knowledge. Here, the control input gives the information about the desired reference that is helpful to provide of stability and robustness in the prediction. Experiment studies using a personal UAV are conducted under closed-form trajectories. The results demonstrate the effectiveness of the proposed approach to infer the final destination of the point to point trajectory and the approximate convergence time based in a proposed threshold. Good trajectory prediction results are given when the reference is constant, otherwise we need knowledge of the control input gain to guarantee good future prediction results. The proposed models are robust to the sampling time and only require data with enough richness for the adequate estimation of the linear model. The PIDMD is a promising technique but possesses many limitations due to its idealistic formulation and issues with the energy function definition. However, it gives an interesting insight for further work which consists in the analysis of non-conservative systems based on different energy functions. In addition, further work includes the analysis and design of DMD methods based on nonlinear observables and time-varying references, as well multi-agent systems for control input estimation.

REFERENCES

- [1] P. Wendt, A. Voltes-Dorta, and P. Suau-Sanchez, "Estimating the costs for the airport operator and airlines of a drone-related shutdown: an application to frankfurt international airport," *Journal of Transportation Security*, vol. 13, no. 1, pp. 93–116, 2020.
- [2] B. Fraser, A. Perrusquía, D. Panagiotakopoulos, and W. Guo, "Hybrid deep neural networks for drone high level intent classification using non-cooperative radar data," in *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. IEEE, 2023, pp. 1–6.
- [3] J. Liang, B. I. Ahmad, M. Jahangir, and S. Godsill, "Detection of malicious intent in non-cooperative drone surveillance," in *2021 Sensor Signal Processing for Defence Conference (SSPD)*. IEEE, 2021, pp. 1–5.
- [4] C. Anagnostopoulos and S. Hadjiefthymiades, "Intelligent trajectory classification for improved movement prediction," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 10, pp. 1301–1314, 2014.
- [5] S. Samaras, E. Diamantidou, D. Ataloglou, N. Sakellariou, A. Vafeiadis, V. Magoulaniotis, A. Lalas, A. Dimou, D. Zarpalas, K. Votis *et al.*, "Deep learning on multi sensor data for counter uav applications—a systematic review," *Sensors*, vol. 19, no. 22, p. 4837, 2019.
- [6] G. He, X. Li, Y. Lv, B. Gao, and H. Chen, "Probabilistic intention prediction and trajectory generation based on dynamic bayesian networks," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019.
- [7] T. Su, Y. Meng, and Y. Xu, "Pedestrian trajectory prediction via spatial interaction transformer network," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE, 2021, pp. 154–159.
- [8] K. Saleh, M. Hossny, and S. Nahavandi, "Intent prediction of pedestrians via motion trajectories using stacked recurrent neural networks," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 414–424, 2018.
- [9] Y. Cho, J. Kim, and J. Kim, "Intent inference of ship collision avoidance behavior under maritime traffic rules," *IEEE Access*, vol. 9, pp. 5598–5608, 2021.
- [10] A. Perrusquía and W. Guo, "Physics informed trajectory inference of a class of nonlinear systems using a closed-loop output error technique," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.
- [11] J. L. Yepes, I. Hwang, and M. Rotea, "New algorithms for aircraft intent inference and trajectory prediction," *Journal of guidance, control, and dynamics*, vol. 30, no. 2, pp. 370–382, 2007.
- [12] R. Rezaie and X. R. Li, "Destination-directed trajectory modeling, filtering, and prediction using conditionally markov sequences," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 2, pp. 820–833, 2020.
- [13] H. Zhang, Y. Yan, S. Li, Y. Hu, and H. Liu, "Uav behavior-intention estimation method based on 4-d flight-trajectory prediction," *Sustainability*, vol. 13, no. 22, p. 12528, 2021.
- [14] Q. Fu, X. Liang, J. Zhang, and X. Fan, "Intent inference based trajectory prediction and smooth for uas in low-altitude airspace with geofence," *CMC-Computers Materials & Continua*, vol. 63, no. 1, pp. 417–444, 2020.
- [15] G. Singh, A. Perrusquía, and W. Guo, "A two-stages unsupervised/ supervised statistical learning approach for drone behaviour prediction," in *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2023, pp. 1–6.
- [16] J. Liang, B. I. Ahmad, and S. Godsill, "Simultaneous intent prediction and state estimation using an intent-driven intrinsic coordinate model," in *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2020, pp. 1–6.
- [17] G. G. Rigatos, "Nonlinear kalman filters and particle filters for integrated navigation of unmanned aerial vehicles," *Robotics and Autonomous Systems*, vol. 60, no. 7, pp. 978–995, 2012.
- [18] A. Perrusquía, R. Garrido, and W. Yu, "Stable robot manipulator parameter identification: A closed-loop input error approach," *Automatica*, vol. 141, p. 110294, 2022.
- [19] T. Kacker, A. Perrusquía, and W. Guo, "Multi-spectral fusion using generative adversarial networks for uav detection of wild fires," in *2023*

- International Conference on Artificial Intelligence in Information and Communication (ICAIC)*. IEEE, 2023, pp. 182–187.
- [20] A. Perrusquía and W. Yu, “Neural H_2 control using continuous-time reinforcement learning,” *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 4485–4494, 2022.
- [21] Z. Yang, R. Tang, J. Bao, J. Lu, and Z. Zhang, “A real-time trajectory prediction method of small-scale quadrotors based on gps data and neural network,” *Sensors*, vol. 20, no. 24, p. 7061, 2020.
- [22] K. Mangalam, Y. An, H. Girase, and J. Malik, “From goals, waypoints & paths to long term human trajectory forecasting,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 233–15 242.
- [23] N. M. Gottschling, V. Antun, B. Adcock, and A. C. Hansen, “The troublesome kernel: why deep learning for inverse problems is typically unstable,” *arXiv preprint arXiv:2001.01258*, 2020.
- [24] A. Perrusquía, “A complementary learning approach for expertise transference of human-optimized controllers,” *Neural Networks*, vol. 145, pp. 33–41, 2022.
- [25] H. Oh, H.-S. Shin, S. Kim, A. Tsourdos, and B. A. White, “Airborne behaviour monitoring using gaussian processes with map information,” *IET Radar, Sonar & Navigation*, vol. 7, no. 4, pp. 393–400, 2013.
- [26] R. Rezaie and X. R. Li, “Gaussian conditionally markov sequences: Modeling and characterization,” *Automatica*, vol. 131, p. 109780, 2021.
- [27] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.
- [28] P. Héas and C. Herzet, “Low-rank dynamic mode decomposition: An exact and tractable solution,” *Journal of Nonlinear Science*, vol. 32, no. 1, p. 8, 2022.
- [29] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for machine learning*. Cambridge University Press, 2020.
- [30] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.
- [31] P. J. Schmid, “Dynamic mode decomposition and its variants,” *Annual Review of Fluid Mechanics*, vol. 54, pp. 225–254, 2022.
- [32] M. A. Gultekin and A. Bazzi, “Real-time data-driven system identification of motor drive systems using online dmdc,” in *2022 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, 2022, pp. 1–7.
- [33] A. Perrusquía and W. Guo, “A closed-loop output error approach for physics-informed trajectory inference using online data,” *IEEE Transactions on Cybernetics*, 2022.
- [34] K. G. Vamvoudakis and F. L. Lewis, “Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem,” *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.
- [35] A. Perrusquía and W. Yu, “Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview,” *Neurocomputing*, vol. 438, pp. 145–154, 2021.
- [36] J. L. Proctor, S. L. Brunton, and J. N. Kutz, “Dynamic mode decomposition with control,” *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.
- [37] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. Nathan Kutz, and S. L. Brunton, “Physics-informed dynamic mode decomposition,” *Proceedings of the Royal Society A*, vol. 479, no. 2271, p. 20220576, 2023.
- [38] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.
- [39] R. Wang, D. Maddix, C. Faloutsos, Y. Wang, and R. Yu, “Bridging physics-based and data-driven modeling for learning dynamical systems,” in *Learning for Dynamics and Control*. PMLR, 2021, pp. 385–398.
- [40] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction*. Princeton university press, 2011.
- [41] W. Yu and A. Perrusquía, *Human-Robot Interaction Control Using Reinforcement Learning*. Wiley-IEEE Press, 2022, ch. Robot Kinematics and Dynamics.
- [42] B. Herrmann, P. J. Baddoo, R. Semaan, S. L. Brunton, and B. J. McKeon, “Data-driven resolvent analysis,” *Journal of Fluid Mechanics*, vol. 918, 2021.
- [43] P. H. Schönemann, “A generalized solution of the orthogonal procrustes problem,” *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.
- [44] D. A. Bristow, M. Tharayil, and A. G. Alleyne, “A survey of iterative learning control,” *IEEE control systems magazine*, vol. 26, no. 3, pp. 96–114, 2006.
- [45] H. Vo and S. Seshagiri, “Robust control of f-16 lateral dynamics,” in *2008 34th annual conference of IEEE industrial electronics*. IEEE, 2008, pp. 343–348.
- [46] A. Perrusquía and W. Guo, “Cost inference of discrete-time linear quadratic control policies using human-behaviour learning,” in *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1. IEEE, 2022, pp. 165–170.
- [47] R. Srikant and C. S. Rao, “Application of backpropagation neural networks in calculation of robot kinematics,” in *Case Studies in Intelligent Computing*. Auerbach Publications, 2014, pp. 406–417.
- [48] A. Perrusquía and W. Guo, “Drone’s objective inference using policy error inverse reinforcement learning,” *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [49] T. Luukkonen, “Modelling and control of quadcopter,” *Independent research project in applied mathematics, Espoo*, vol. 22, p. 22, 2011.
- [50] A. Perrusquía, “Robust state/output feedback linearization of direct drive robot manipulators: A controllability and observability analysis,” *European Journal of Control*, vol. 64, p. 100612, 2022.
- [51] Y. Kartal, P. Kolaric, V. Lopez, A. Dogan, and F. Lewis, “Backstepping approach for design of pid controller with guaranteed performance for micro-air uav,” *Control Theory and Technology*, vol. 18, pp. 19–33, 2020.



Adolfo Perrusquía received the B.Eng. degree in Mechatronic Engineering from the National Polytechnic Institute (UPIITA-IPN) in 2014, and the M.S. and Ph.D. degrees, both in Automatic Control from the Automatic Control Department at the CINVESTAV-IPN in 2016 and 2020, respectively. He is currently a lecturer at the School of Aerospace, Transport and Manufacturing, Cranfield University, and a former UK-IC Postdoctoral Research Fellow. He is a member of the IEEE Computational Intelligence Society. His main research of interest focuses

on robotics, mechanisms, machine learning, reinforcement learning, nonlinear control, system modeling and system identification.



Zhuangkun Wei received the bachelor’s and master’s degrees in electronic engineering from Beijing University of Posts and Telecommunications (BUPT), China, in 2014 and 2018, respectively, and the Ph.D. degree in engineering from the University of Warwick, U.K., in 2021. His research interests cover physical layer security, graph signal processing, molecular communications, and Explainable Artificial Intelligence (XAI). He is currently a research fellow in the School of Aerospace, Transport and Manufacturing, Cranfield University.



Weisi Guo received the M.Eng. degree in engineering, and the M.A. and Ph.D. degrees in computer science from the University of Cambridge, Cambridge, U.K., in 2005, 2011, and 2011, respectively. He is a Chair Professor of Human Machine Intelligence with Cranfield University, Cranfield, U.K. He has published over 180 papers and is a PI on a number of molecular communication research grants.

Prof. Guo’s research has won him several international awards. He was a Turing Fellow at the Alan Turing Institute.



DSPACE

<https://dspace.org/>

Trajectory intent prediction of autonomous systems using dynamic mode decomposition

Perrusquía, Adolfo; Wei, Zhuangkun; Guo, Weisi

2024-12-01

IEEE

<https://doi.org/10.1109/tsmc.2024.3462790>

Perrusquía A, Wei Z, Guo W. (2024) Trajectory intent prediction of autonomous systems using dynamic mode decomposition. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Volume 54, Issue 12, December 2024, pp. 7897-7908