IEEE Transactions on Systems, Man, and Cybernetics: Systems, Available online 9 January 2024
DOI:10.1109/TSMC.2023.3344017

Trajectory Inference of Unknown Linear Systems based on Partial States Measurements

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

***Abstract*—Proliferation of cheaper autonomous system prototypes has magnified the threat space for attacks across the manufacturing, transport, and smart living sectors. An accurate trajectory inference algorithm is required for monitoring and early detection of autonomous misbehaviour and to take relevant countermeasures. This paper presents a trajectory inference algorithm based on a closed-loop output error approach using partial states measurements. The approach is based on a physics informed state parameteterization that combines the main advantages of state estimation and identification algorithms. Noise attenuation and parameter estimates convergence are obtained if the output trajectories fulfil a persistent excitation condition. Known and unknown desired reference/destination cases are considered. The stability and convergence of the proposed approach are assessed via Lyapunov stability theory under the fulfilment of a persistent excitation condition. Simulation studies are carried out to verify the effectiveness of the proposed approach.**

***Index Terms*—Trajectory inference, Closed-loop Output error, State parameterization, Output measurements, Parameter identification, Excitation signal.**

I. Introduction

In the last decades, one of the main control and machine learning issues in many industrial and military applications [1], [2] involving autonomous systems is regarded to the trajectory inference problem [3], [4]. Its main objective is to infer the trajectory that follows a system from previous and/or current states measurements [5] and a prior knowledge of its physics [6], also known as model-based.

Trajectory inference is usually done off-line by searching a model that fits the data history of the system's trajectories to infer future destinations or mission profiles [7]. However, current detection technologies (e.g., radars) are non-cooperative, that is, they only provide noisy partial states measurements and, in the worst case, the measurements contain gaps. These problems hinder the accurate realization of the inference task. Therefore, the inference problem is not regarded to only predict the future trajectory, but also to find a model that is capable to reconstruct the complete past trajectory using the available measurements and to estimate the unmeasured ones.

State observers such as the Luenberger observer [8] or its variants [9] are one of the classical model-based approaches for state estimation that use partial states measurements. The performance of the state observer is degraded when there exists modelling error and noise at the output measurements. Model-free approaches such as high-gain observers [10] and sliding-mode observers [11] are used to overcome this issue by taking advantage of the robust properties of the sliding manifold. However, these models are sensitive to measurement noise and are affected by the "peaking" phenomena.

Kalman filter [12] is an optimal state observer that incorporates modelling error at the prior model and noise at the output measurements. Both the modelling error and the noise are assumed to have a Gaussian distribution [13], [14]. Kalman filter and its variants combine two different sources of knowledge: a prior model of the system dynamics and a posterior state estimation such that the mean between these two state estimations matches with the exact mean of the posterior state distribution. Nevertheless, Kalman filter can rapidly diverge when the prior model is not accurate or wrongly converge to different state estimates.

Whilst high-gain observers is the model-free version of the Luenberger observer, particle filter [15] is the model-free version of Kalman filter. In the particle filter approach, only sample points of the output measurements and a proposed distribution are used to generate normalized histograms that reconstruct the shape of the followed trajectory. However, the selection of the proposed distribution is not trivial and requires a large number of samples to achieve accurate results [16].

Neural networks are used as a model-free approach for state inference [17], [18]. Two different approaches can be distinguished direct-solution models and time-stepper models [19]. Whilst direct models [20] infer the trajectory from an initial value problem for a given initial state and set of inputs, the time-stepper models [21] learn the dynamics. Time-stepper methods are characterized by using a similar approach to numerical solvers [22], [23], that is, the current state is used to obtain the next state in the next time instance. However, these approaches require that the physics of the system to be normalized to avoid divergence of the neural weights. Furthermore, these networks require full states measurements. Alternatively, recurrent neural networks (RNNs) [24] are used as a complementary algorithm for Kalman filter algorithms [25], [26]. The idea is to incorporate memory [27] and adaptation to overcome the lack of knowledge of the prior model [28]. However, the RNN is data hungry and it is not able to capture the real physics of the system and, in consequence, biased estimates are obtained.

Reinforcement Learning (RL) has been used mainly for control purposes [29], [30] to find the optimal control policy that minimizes a cost function with or without a model 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 and Weisi Guo are with the School of Aerospace, Transport and Manufacturing, Cranfield University, Bedford, UK (e-mail: {Adolfo.Perrusquia-Guzman,weisi.guo}@cranfield.ac.uk).

© 2024 IEEE. This is the Author Accepted Manuscript issued with: Creative Commons Attribution License (CC:BY 4.0).
The final published version (version of record) is available online at DOI:10.1109/TSMC.2023.3344017. Please refer to any applicable publisher terms of use.

the system [31], [32]. Here, the dynamic model of the system is inferred from the system trajectories to solve recursively a Hamilton-Jacobi-Bellman equation. However, RL techniques have not been used for online trajectory estimation problems and require access to the control input which is not available in non-cooperative detection systems.

The best performance of the aforementioned models is achieved when an accurate dynamic model is used, also known as physics informed model [19]. System identification methods aim to estimate the parameters [24], [33], [34] of a given system in order to design physics informed models for either model-based controllers and gain tunings. The most popular methods for parameter identification are least-squares (LS) [35], [36] and gradient algorithms and their respective variants. Parameter convergence can be achieved under the fulfilment of a persistent of excitation (PE) condition [37] at the regressor matrix. However, these kind of algorithms require complete states measurements. Moreover, if the measurements are noisy then biased estimates are obtained [38].

Novel architectures have been developed for parameter identification [39]–[41]. These methods are the Closed-Loop Input Error (CLIE) [42] and the Closed-Loop Output Error (CLOE) [43], [44]. Both the CLIE and CLOE algorithms have a state observer structure based on the real system and an estimated model which use either the input error or output error to update an identification law that subsequently updates the estimated model [45]. On the one hand, the CLIE algorithm assumes knowledge of the control input and complete state output to estimate the parameters of a system. On the other hand, the CLOE algorithm requires complete output measurements to estimate the parameters of the system. In addition, the CLOE algorithm requires initial parameter estimates close to their real values to achieve good results.

Therefore, there exists a trade-off between state estimation and system identification. On the one hand, state estimation requires an accurate physics-informed model to achieve good trajectory inference. Conversely, system identification requires noise-free state measurements to obtain unbiased parameter estimates. In addition, to the best of our knowledge there are no parameter identification algorithms that consider partial states measurements. Furthermore, there are no theoretical approaches that combine the main advantages of state estimation and identification algorithms and verify both the convergence of the parameters estimates and states.

Inspired by the previous comments, this paper reports a CLOE algorithm for physics informed trajectory inference using partial states measurements. The approach extends the scope of standard CLOE algorithms and combines the advantages of both estimation and identification algorithms in a complementary mechanism. Parameter and states convergence is analysed using Lyapunov stability theory. The main contributions of the work are the following:

- The CLOE algorithm is extended for partial states measurements based on a physics-informed parameterization in terms of the desired reference/destination and the output measurements of an estimated model.
- Both the parameters and states estimates are computed simultaneously using the output of an estimated model and a proposed excitation signal instead of the noisy-output measurements.
- A rigorous stability analysis of the closed-loop output error dynamics using Lyapunov stability theory is provided.
- The regressor matrices associated to the desired reference and the estimated output measurements are noise-free and consequently the estimated states are also noise-free.

The paper outline is as follows: Section II presents the problem formulation. Section III defines the state parameterization based on the desired trajectory and the output measurements. Section IV presents the CLOE trajectory inference algorithm. Section V covers a CLOE architecture for unknown desired reference/destination. Section VI reports the simulations studies using different autonomous system models. The conclusions are presented in Section VII.

Throughout this paper, $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{R}^n$, $\mathbb{R}^{n\times m}$ denote the spaces of natural numbers, real numbers, positive real numbers, real n-vectors, and real $n\times m$-matrices, respectively; $I_n \in \mathbb{R}^{n\times n}$ denotes an identity matrix; $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denotes the minimum and maximum eigenvalues of matrix A, respectively; $\det(A)$ and $\text{adj}(A)$ stand to the determinant and the adjoint matrix of matrix A; $\mathcal{L}$ denotes the Laplace transform, $\otimes$ and $\text{vec}(A)$ defines the Kronecker product and the matrix stretch, the norms $\|A\| = \sqrt{\lambda_{\max}(A^\top A)}$ and $\|x\|$ stand for the induced matrix and vector Euclidean norms, respectively; where $x \in \mathbb{R}^n$, $A, B \in \mathbb{R}^{n\times n}$ and $n, m \in \mathbb{N}$.

II. Problem Formulation

Consider a linear time invariant continuous-time system [8]

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx, \quad x(0) = x_0, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input, $y \in \mathbb{R}^p$ is the output vector with $p < n$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$ denote the plant, input, and output matrices, respectively. The matrices A and B are unknown.

Assumption 1: The pair (A, B) is controllable and the pair (A, C) is observable.

Assumption 2: Only measurements of the output y are available, that is, we have partial-states measurements.

Assumption 3: The control input u has an unknown linear structure that ensures stability of the closed-loop trajectories and tracking of a desired reference x_d.

Consider that the control input of system (1) has the following linear structure [46]

$$u = K(x_d - x) - B^\dagger(Ax_d - \dot{x}_d), \tag{2}$$

where $K \in \mathbb{R}^{m\times n}$ is an unknown stabilizing gain, $B^\dagger$ is the Moore-Penrose pseudoinverse of B, $x_d, \dot{x}_d \in \mathbb{R}^n$ are the desired reference and its derivative. Substituting (2) in (1) leads to

$$\dot{x} = (A - BK)(x - x_d) + \dot{x}_d, \tag{3}$$

The matrix $A - BK$ is Hurwitz, such that the closed-loop error dynamics

$$\dot{x} - \dot{x}_d = (A - BK)(x - x_d) \tag{4}$$

exhibits exponential stability [47], that is, $\lim_{t\to\infty} x(t) = x_d(t)$. This paper aims to estimate the hidden physics and infer the trajectory of (3) from the partial data measurements y and knowledge of the desired reference. However, this approach can also be expanded for unknown desired references.

III. State parameterization

The following parameterization is derived from the structure of a Luenberger observer [48], [49]. The LTI system (1) can be expressed as

$$\begin{aligned} \dot{x} &= Ax + Bu + L(y - y) \\ u &= K(x_d - x) - B^{\dagger}(Ax_d - \dot{x}_d) \\ y &= Cx, \end{aligned} \tag{5}$$

where $L \in \mathbb{R}^{n\times p}$ is a stabilizing gain. The above system can be reduced to the following differential equation

$$\dot{x} = (A - BK + LC)x - (A - BK)x_d + \dot{x}_d - Ly. \tag{6}$$

The additional term does not affect the real system trajectories, however is helpful for this approach since we are able to incorporate the output measurements in the main differential equation. Notice that the matrix $A-BK+LC$ is also Hurwitz. The Laplace transform of (6) is

$$\begin{aligned} X(s) =&(sI - \mathcal{A})^{-1}x_0 + (sI - \mathcal{A})^{-1}(sI - \mathcal{A} + F)X_d(s) \\ &- (sI - \mathcal{A})^{-1}LY(s), \end{aligned} \tag{7}$$

where $\mathcal{A} = A - BK + F \in \mathbb{R}^{n\times n}$ and $F = LC \in \mathbb{R}^{n\times n}$. Then, the solution of (6) is

$$x(t) = e^{\mathcal{A}t}x_0 + x_d(t) + \int_0^t e^{\mathcal{A}(t-\tau)}(Fx_d(\tau) - Ly(\tau))d\tau. \tag{8}$$

The solution (8) gives one way to parametrize the states in terms of the desired reference x_d and the output y. The solution (7) can be expressed as

$$\begin{aligned} X(s) =&(sI - \mathcal{A})^{-1}x_0 - (sI - \mathcal{A})^{-1}\sum_{i=1}^{p} L^i Y^i(s) \\ &+ X_d(s) + (sI - \mathcal{A})^{-1}\sum_{i=1}^{n} F^i X_d^i(s), \end{aligned} \tag{9}$$

where L^i and F^i denote the column i of matrices L and F, respectively; analogously, the terms X_d^i and Y^i denote the element i of vectors X_d and Y. By construction, the characteristic polynomial of $sI - \mathcal{A}$ satisfies

$$D(s) = \det(sI - \mathcal{A}) = s^n + \alpha_{n-1}s^{n-1} + \cdots + \alpha_1 s + \alpha_0,$$

where $\alpha_i > 0$. Then,

$$\begin{aligned} \sum_{i=1}^{n}(sI - \mathcal{A})^{-1}F^i X_d^i(s) :=& \sum_{i=1}^{n}\frac{\text{adj}(sI - \mathcal{A})}{D(s)}F^i X_d^i(s), \\ \sum_{i=1}^{p}(sI - \mathcal{A})^{-1}L^i Y^i(s) :=& \sum_{i=1}^{p}\frac{\text{adj}(sI - \mathcal{A})}{D(s)}L^i Y^i(s), \end{aligned} \tag{10}$$

where $\text{adj}(A)$ is the adjoint matrix of matrix A. Consider the first summation of (10), then it is possible to split the numerator as

$$\begin{aligned} \text{adj}(sI - \mathcal{A})F^i &= \begin{bmatrix} \beta_{n-1}^{i1}s^{n-1} + \cdots + \beta_1^{i1}s + \beta_0^{i1} \\ \beta_{n-1}^{i2}s^{n-1} + \cdots + \beta_1^{i2}s + \beta_0^{i2} \\ \vdots \\ \beta_{n-1}^{in}s^{n-1} + \cdots + \beta_1^{in}s + \beta_0^{in} \end{bmatrix} \\ &= \begin{bmatrix} \beta_0^{il} & \beta_1^{il} & \cdots & \beta_{n-1}^{il} \\ \beta_0^{i2} & \beta_1^{i2} & \cdots & \beta_{n-1}^{i2} \\ \vdots & \vdots & \vdots & \vdots \\ \beta_0^{in} & \beta_1^{in} & \cdots & \beta_{n-1}^{in} \end{bmatrix} \begin{bmatrix} 1 \\ s \\ \vdots \\ s^{n-1} \end{bmatrix} \\ &:= W_x^i N(s) \end{aligned}$$

where $W_x^i \in \mathbb{R}^{n\times n}$ contains the coefficients of the numerator and $N(s) \in \mathbb{R}^n$ is a vector of powers of s. So,

$$\sum_{i=1}^{n}\frac{\text{adj}(sI - \mathcal{A})}{D(s)}F^i X_d^i(s) = \sum_{i=1}^{n} W_x^i \frac{N(s)}{D(s)}X_d^i(s). \tag{11}$$

The term $N(s)/D(s)$ defines n filters applied to the desired reference $x_d(t)$. This vector of n filters of $x_d^i(t)$ can be obtained by the following linear system

$$\begin{aligned} \dot{\xi}_x^i &= A_L\xi_x^i + B_L x_i^d, \quad \xi_x^i(0) = 0, \\ \zeta_x^i &= \xi_x^i, \end{aligned} \tag{12}$$

where ξ_x^i defines the state of the linear system i, ζ_x^i is the output which is equivalent to the state ξ_x, and the matrices $A_L \in \mathbb{R}^{n\times n}$ and $B_L \in \mathbb{R}^n$ are given by

$$A_L = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \cdots & -\alpha_n \end{bmatrix}, \; B_L = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Note that $(sI - A_L) = D(s)$. So, the first summation in (10) can be written in the time domain as

$$\mathcal{L}^{-1}\left\{\sum_{i=1}^{n}\frac{\text{adj}(sI - \mathcal{A})}{D(s)}F^i X_d^i(s)\right\} = W_x\xi_x, \tag{13}$$

where $W_x = [W_x^1, W_x^2, \cdots, W_{x_2}^n] \in \mathbb{R}^{n\times n^2}$ and $\xi_x = [(\xi_x^1)^\top, (\xi_x^2)^\top, \cdots, (\xi_x^n)^\top]^\top \in \mathbb{R}^{n^2}$. A similar procedure is applied to the second summation. So, its time domain representation is

$$\mathcal{L}^{-1}\left\{\sum_{i=1}^{p}\frac{\text{adj}(sI - \mathcal{A})}{D(s)}L^i Y^i(s)\right\} = W_y\xi_y, \tag{14}$$

where $W_y = [W_y^1, W_y^2, \cdots, W_y^p] \in \mathbb{R}^{n\times np}$ and $\xi_y = [(\xi_y^1)^\top, (\xi_y^2)^\top, \cdots, (\xi_y^p)^\top]^\top \in \mathbb{R}^{np}$. The elements $W_y^i \in \mathbb{R}^{n\times n}$ are the coefficients of $\text{adj}(sI - \mathcal{A})L^i$ and $\xi_y^i \in \mathbb{R}^n$ is the state vector obtained from the following linear system

$$\begin{aligned} \dot{\xi}_y^i &= A_L\xi_y^i + B_L y^i, \quad \xi_y^i(0) = 0, \\ \zeta_y^i &= \xi_y^i. \end{aligned} \tag{15}$$

So (8) can be equivalently written as

$$x = e^{\mathcal{A}t}x_0 + x_d + W_x\xi_x - W_y\xi_y. \tag{16}$$

We can easily compute the term $e^{\mathcal{A}t}$ by considering the eigendecomposition [50] of matrix $\mathcal{A}$, i.e.,

$$\mathcal{A} = P\mathcal{D}P^{-1}, \tag{17}$$

where $P \in \mathbb{R}^{n\times n}$ is a matrix whose columns are the eigenvectors of matrix $\mathcal{A}$, and $\mathcal{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose elements on its diagonal are the eigenvalues of matrix $\mathcal{A}$. Then the exponential of matrix $\mathcal{A}$ is given by

$$e^{\mathcal{A}t} = Pe^{\mathcal{D}t}P^{-1}. \tag{18}$$

Remark 1: The existence of the eigendecomposition requires that the algebraic multiplicity of repeated eigenvalues have the same number of associated eigenvectors. This issue can be solved by choosing different eigenvalues with algebraic multiplicity one.

However, the term $e^{\mathcal{A}t}x_0$ exponentially converges to zero as $t \to \infty$. So, for instance we can assume that this term is zero or that $x_0 = 0$. Taking the time derivative of (16) along the linear systems (12) and (15) is

$$\begin{aligned}\dot{x} =& \dot{x}_d + W_x\dot{\xi}_x - W_y\dot{\xi}_y \\ =& \dot{x}_d + W_x\begin{bmatrix}A_L & \cdots & A_L\end{bmatrix}\xi_x + W_x\begin{bmatrix}B_L & \cdots & B_L\end{bmatrix}x_d \\ & - W_y\begin{bmatrix}A_L & \cdots & A_L\end{bmatrix}\xi_y - W_y\begin{bmatrix}B_L & \cdots & B_L\end{bmatrix}y \\ =& \dot{x}_d + W_x(A_x\xi_x + B_x x_d) - W_y(A_y\xi_y + B_y y) \\ =& \dot{x}_d + \Phi_x^\top\Theta_x - \Phi_y^\top\Theta_y,\end{aligned} \tag{19}$$

where the matrices associated to the desired reference are given by $A_x = \text{diag}\{A_L, \cdots, A_L\} \in \mathbb{R}^{n^2\times n^2}$, $B_x = \text{diag}\{B_L, \cdots, B_L\} \in \mathbb{R}^{n^2\times n}$, and equivalently, the matrices associated to the output are $A_y = \text{diag}\{A_L, \cdots, A_L\} \in \mathbb{R}^{np\times np}$, and $B_y = \text{diag}\{B_L, \cdots, B_L\} \in \mathbb{R}^{np\times n}$. The terms $\Phi_x = \Phi(x_d, \xi_x) \in \mathbb{R}^{r_1\times n}$ and $\Phi_y = \Phi_y(y, \xi_y) \in \mathbb{R}^{r_2\times n}$ define the regressor matrices and $\Theta_x \in \mathbb{R}^{r_1}$, $\Theta \in \mathbb{R}^{r_2}$ are the parameters vectors which can be expressed as

$$\begin{aligned}\Phi_x &= I_n \otimes (A_x\xi_x + B_x x_d), \quad \Theta_x = \text{vec}(W_x) \\ \Phi_y &= I_n \otimes (A_y\xi_y + B_y y), \quad \Theta_y = \text{vec}(W_y).\end{aligned} \tag{20}$$

Notice that $W_xB_x = F$ and $W_yB_y = L$. In the next section the trajectory inference algorithm is developed using only the output measurements y and the desired destination x_d.

IV. CLOE Trajectory Inference

The proposed algorithm aims to estimate both the states and parameters of the parameterization (19). To achieve this goal, the output measurements need to be rich enough [51], [52] to excite the identification algorithm. So, an excitation signal is added to the output measurements as

$$y_\tau = y + C\tau, \tag{21}$$

where $\tau \in \mathbb{R}^n$ is an excitation vector. This new output satisfies $x_\tau = x + \tau$ and belongs to the next LTI system

$$\dot{x}_\tau = \mathcal{A}x_\tau - (\mathcal{A} - F)(x_d + \tau) + \dot{x}_d + \dot{\tau} - Ly_\tau \tag{22}$$

which can be equivalently written as

$$\begin{aligned}\dot{x}_\tau =& \quad \dot{x}_d + \dot{\tau} + W_x(A_x\xi_x + B_x(x_d + \tau)) \\ & - W_y(A_y\xi_y + B_y y_\tau) \\ =& \quad \dot{x}_d + \dot{\tau} + \Phi_x^\top\Theta_x - \Phi_y^\top\Theta_y,\end{aligned} \tag{23}$$

where Φ_x and Φ_y are rewritten as

$$\begin{aligned}\Phi_x &= I_n \otimes (A_x\xi_x + B_x(x_d + \tau)) \\ \Phi_y &= I_n \otimes (A_y\xi_y + B_y y_\tau).\end{aligned}$$

Fig. 1 depicts the diagram of the proposed trajectory inference algorithm. Two subsystems composed of n linear systems $\dot{\xi}_x^i$ and p linear systems $\dot{\xi}_w^i$; with the output error e, between the output measurements y_τ and the output w of an estimated model (constructed by the proposed state-parameterization), feed an identification algorithm that subsequently updates the estimated model.

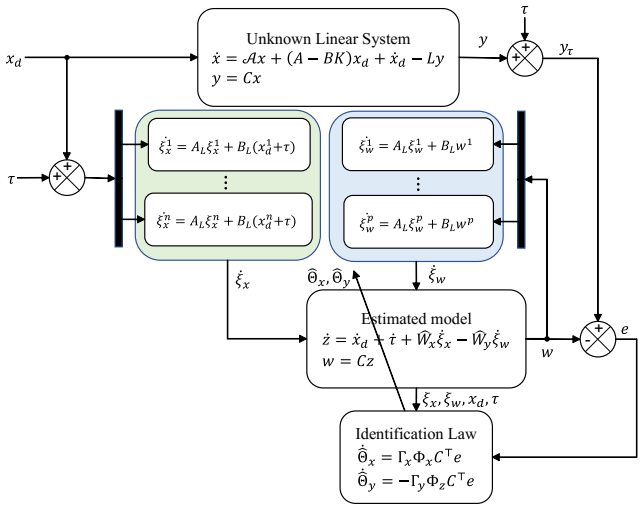


Fig. 1. Block diagram of the proposed CLOE trajectory inference

Consider an estimated model which has the following structure

$$\begin{aligned}\dot{z} &= \widehat{\mathcal{A}}z - (\widehat{\mathcal{A}} - \widehat{F})(x_d + \tau) + \dot{x}_d + \dot{\tau} - \widehat{L}w \\ w &= Cz, \quad z(0) = z_0 = x_0,\end{aligned} \tag{24}$$

where $z \in \mathbb{R}^n$ denotes the states of the estimated model, $w \in \mathbb{R}^p$ is the output, and $\widehat{\mathcal{A}}$, $\widehat{F}$, and $\widehat{L}$ are estimates of $\mathcal{A}$, F, and L, respectively. The estimated model (24) written in the form of (23) is

$$\begin{aligned}\dot{z} =& \quad \dot{x}_d + \dot{\tau} + \widehat{W}_x(A_x\xi_x + B_x(x_d + \tau)) \\ & - \widehat{W}_y(A_y\xi_w + B_y w) \\ =& \quad \dot{x}_d + \dot{\tau} + \Phi_x^\top\widehat{\Theta}_x - \Phi_z^\top\widehat{\Theta}_y,\end{aligned} \tag{25}$$

where $\widehat{W}_x \in \mathbb{R}^{n\times n^2}$ and $\widehat{W}_y \in \mathbb{R}^{n\times np}$ are the estimates of W_x and W_y, respectively. The term $\Phi_z = \Phi_z(w, \xi_w) \in \mathbb{R}^{r_2\times n}$ defines the regressor matrix constructed with the output measurements of the estimated model and $\widehat{\Theta}_x \in \mathbb{R}^{r_1}$, $\widehat{\Theta}_y \in \mathbb{R}^{r_2}$ are the parameter estimates vectors. The regressor matrix Φ_z and the parameter estimates vector $\widehat{\Theta}$ are given by

$$\begin{aligned}\Phi_z &= I_n \otimes (A_y\xi_w + B_y w), \\ \widehat{\Theta}_x &= \text{vec}(\widehat{W}_x), \quad \widehat{\Theta}_y = \text{vec}(\widehat{W}_y).\end{aligned} \tag{26}$$

The state $\xi_w = [(\xi_w^1)^\top, (\xi_w^2)^\top, \cdots, (\xi_w^n)^\top]^\top \in \mathbb{R}^{np}$ is obtained from the following linear systems

$$\begin{aligned}\dot{\xi}_w^i &= A_L\xi_w^i + B_Lw^i, \\ \zeta_w^i &= \xi_w^i.\end{aligned} \tag{27}$$

Define the output error between the output measurements y_τ and the output of the estimated model w as

$$e = y_\tau - w. \tag{28}$$

The closed-loop output error (CLOE) dynamics between (23) and (25) is

$$\begin{aligned}
\dot{e} =& C\Big(W_x(A_x\xi_x + B_x(x_d+\tau)) - W_y(A_y\xi_y + B_y y_\tau) \\
&- \widehat{W}_x(A_x\xi_x + B_x(x_d+\tau)) + \widehat{W}_y(A_y\xi_w + B_y w)\Big) \\
=& - C\Big(W_y A_y(\xi_y - \xi_w) + W_y B_y e \\
&+ \widetilde{W}_x(A_x\xi_x + B_x(x_d+\tau)) - \widetilde{W}_y(A_y\xi_w + B_y w)\Big) \\
=& - C\Big(W_y A_y(\xi_y - \xi_w) + W_y B_y e + \Phi_x^\top\widetilde{\Theta}_x - \Phi_z^\top\widetilde{\Theta}_y\Big) \\
=& - C(W_y B_y e + \Phi_x^\top\widetilde{\Theta}_x - \Phi_z^\top\widetilde{\Theta}_y + \varepsilon)
\end{aligned} \tag{29}$$

where $\widetilde{W}_x = \widehat{W}_x - W_x \in \mathbb{R}^{n\times n^2}$, $\widetilde{W}_y = \widehat{W}_y - W_y \in \mathbb{R}^{n\times np}$ denote the parametric error of matrices W_x and W_y; the terms $\widetilde{\Theta}_x = \text{vec}(\widetilde{W}_x) \in \mathbb{R}^{r_1}$ and $\widetilde{\Theta}_y = \text{vec}(\widetilde{W}_y) \in \mathbb{R}^{r_2}$ denote the parametric error of the linear parameterization; and $\varepsilon := W_y A_y(\xi_y - \xi_w)$ is a bounded residual error, i.e., $\|\varepsilon\| \le \bar{\varepsilon} \ge 0$.

Parameter and states convergence is achieved under the fulfilment of the next persistent of excitation condition.

Lemma 1: [53] The regressors $\Phi_x : \mathbb{R}^n \times \mathbb{R}^{n^2} \to \mathbb{R}^{r_1\times n}$ and $\Phi_z : \mathbb{R}^p \times \mathbb{R}^{np} \to \mathbb{R}^{r_2\times n}$ are persistently exciting (PE) if there exists α_1, α_2, α_3, $\alpha_4 > 0$, and $T > 0$ such that for all $t \ge 0$, the next relationships are fulfilled

$$\begin{aligned}
\alpha_1 I_{r_1} \le S_1 &= \int_t^{t+T} \Phi_x(\tau)\Phi_x^\top(\tau)d\tau \le \alpha_2 I_{r_1}, \\
\alpha_3 I_{r_2} \le S_2 &= \int_t^{t+T} \Phi_z(\tau)\Phi_z^\top(\tau)d\tau \le \alpha_4 I_{r_2}.
\end{aligned} \tag{30}$$

The following theorem establishes the uniformly ultimate boundedness (UUB) [54] of the CLOE trajectories under the fulfilment of the PE condition (30).

Theorem 1: Consider the CLOE dynamics (29). Assume that the regressor matrix Φ_z fulfils the PE condition (30). Define $k := \lambda_{\min}(CL)$ which verifies

$$k > \lambda_{\max}(C)\bar{\varepsilon} + \rho \tag{31}$$

where $\rho \in \mathbb{R}^+$. If the parameter estimates $\widehat{\Theta}_x$ and $\widehat{\Theta}_y$ are updated by the following update rules

$$\dot{\widehat{\Theta}}_x = \dot{\widetilde{\Theta}}_x = \Gamma_x\Phi_x C^\top e, \tag{32}$$

$$\dot{\widehat{\Theta}}_y = \dot{\widetilde{\Theta}}_y = -\Gamma_y\Phi_z C^\top e, \tag{33}$$

where $\Gamma_x \in \mathbb{R}^{r_1\times r_1}$ and $\Gamma_y \in \mathbb{R}^{r_2\times r_2}$ are positive diagonal matrices. Then the trajectories of (29) are UUB with a practical bound $\mu = \frac{\lambda_{\max}(C)}{k}\bar{\varepsilon}$ and the parameter estimates $\widehat{\Theta}$ remain bounded.

Proof: Consider the following Lyapunov function

$$V = \frac{1}{2}e^\top e + \frac{1}{2}\widetilde{\Theta}_x^\top\Gamma_x^{-1}\widetilde{\Theta}_x + \frac{1}{2}\widetilde{\Theta}_y^\top\Gamma_y^{-1}\widetilde{\Theta}_y \tag{34}$$

The time-derivative of the Lyapunov function (34) along the CLOE dynamics (29) and the update rules (32) and (33) is

$$\begin{aligned}
\dot{V} =& - e^\top C(W_y B_y e + \Phi_x^\top\widetilde{\Theta}_x - \Phi_z^\top\widetilde{\Theta}_y + \varepsilon) + \widetilde{\Theta}_x^\top\Gamma_x^{-1}\dot{\widetilde{\Theta}}_x \\
&+ \widetilde{\Theta}_y^\top\Gamma_y^{-1}\dot{\widetilde{\Theta}}_y \\
=& - e^\top C W_y B_y e - \widetilde{\Theta}_x^\top(\Phi_x C^\top e - \Gamma_x^{-1}\dot{\widetilde{\Theta}}_x) - e^\top C\varepsilon \\
&+ \widetilde{\Theta}_y^\top(\Phi_z C^\top e + \Gamma_y^{-1}\dot{\widetilde{\Theta}}_y) \\
=& - e^\top C(W_y B_y e + \varepsilon) \\
\le& - \lambda_{\min}(CL)\|e\|^2 + \lambda_{\max}(C)\|\varepsilon\|\|e\| \\
=& - k\|e\|\left(\|e\| - \frac{\lambda_{\max}(C)\bar{\varepsilon}}{k}\right)
\end{aligned}$$

Therefore, $\dot{V}$ is negative definite if

$$\|e\| > \frac{\lambda_{\max}(C)}{k}\bar{\varepsilon} \equiv \mu \tag{35}$$

If we select a large enough gain L such that (31) is satisfied ensures that the CLOE trajectories (29) converge to a compact set S_e of radius μ, i.e., $\|e\| \le \mu$ and hence, the trajectories of (29) are UUB.

The uniform complete observability (UCO) result [51] is used to show that $\widehat{\Theta}_x$ and $\widehat{\Theta}_y$ are bounded. First, the update rules (32) and (33) can be written as the following linear-time variant (LTV) system [55]

$$\begin{aligned}
\begin{bmatrix}\dot{\widehat{\Theta}}_x \\ \dot{\widehat{\Theta}}_y\end{bmatrix} &= \begin{bmatrix}\Gamma_x\Phi_x C^\top \\ \Gamma_y\Phi_z C^\top\end{bmatrix} u_y \\
\sigma &= \Phi_x^\top\widehat{\Theta}_x + \Phi_z^\top\widehat{\Theta}_y
\end{aligned} \tag{36}$$

where $\sigma \in \mathbb{R}^n$ is the outputs of each LTV system, $u_y = e$ is the control input, and $B(t) = \begin{bmatrix}\Gamma_x\Phi_x C^\top \\ -\Gamma_y\Phi_z C^\top\end{bmatrix} \in \mathbb{R}^{(r_1+r_2)\times n}$ is a time-varying matrix. Since e is bounded then the output w is also bounded. This implies that the regressor Φ_z is bounded. By construction Φ_x is always bounded. These facts guarantee boundedness of the term

$$\sigma \equiv -CW_y B_y e - \dot{e} - \varepsilon. \tag{37}$$

Notice that (37) and (29) are equivalent. So, since the input $u_y = e$ and the output σ are bounded, and the regressors Φ_x and Φ_z are PE ensure the boundedness of the parametric errors $\widetilde{\Theta}_x$ and $\widetilde{\Theta}_y$, and hence $\widehat{\Theta}_x$ and $\widehat{\Theta}_y$ are also bounded. This completes the proof. ∎

The real estimated trajectory can be easily computed by

$$\begin{aligned}
\widehat{y} &= w - \tau \\
\widehat{x} &= z - [\tau, \dot{\tau}]^\top.
\end{aligned} \tag{38}$$

V. CLOE Trajectory Inference: Unknown Reference

The approach can also be expanded for unknown desired reference or destination. So, (22) can be rewritten as

$$\dot{x}_\tau = \mathcal{A}x_\tau - (\mathcal{A} - F)\tau + \dot{\tau} - Ly_\tau + \eta d, \tag{39}$$

where $\eta = \text{diag}\{\dot{x}_d - (\mathcal{A}-F)x_d\} \in \mathbb{R}^{n\times n}$ is a diagonal matrix that contains the unknown desired reference/destination and

$d = [1, \cdots, 1]^\top \in \mathbb{R}^n$. The proposed state-parameterization is used in (39) to obtain the next LTI system

$$\begin{aligned} \dot{x}_\tau = \;& \dot{\tau} + W_x(A_x\xi_x + B_x\tau) - W_y(A_y\xi_y + B_y y_\tau) \\ & + W_\eta(A_x\xi_\eta + B_x d) \\ =\;& \dot{\tau} - \Phi_y^\top \Theta_y + \Phi_\eta^\top \Theta_\eta. \end{aligned} \tag{40}$$

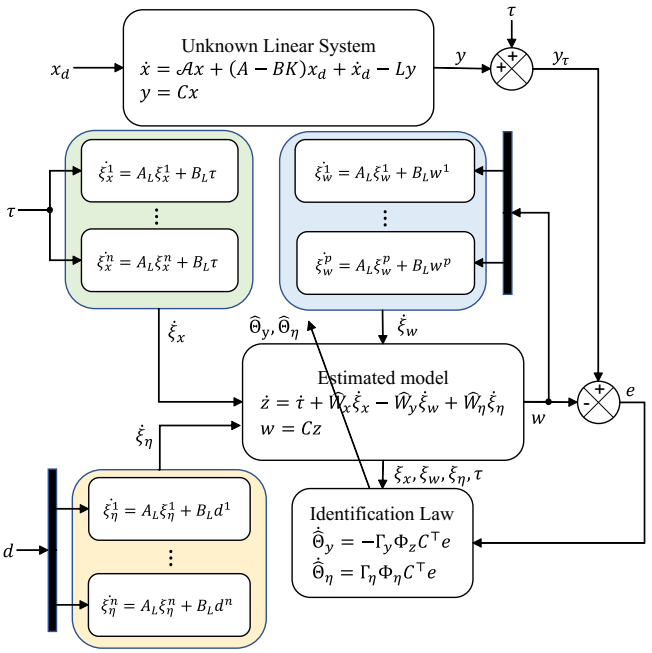


Fig. 2. CLOE diagram for unknown desired reference/destination

The matrix $W_\eta = [W_\eta^1, \cdots, W_\eta^n] \in \mathbb{R}^{n\times n^2}$ is a parameters matrix and their components are computed as $W_\eta^i = \text{adj}(sI - \mathcal{A})\eta^i$; $\xi_\eta = [(\xi_\eta^1)^\top, \cdots, (\xi_\eta^n)^\top]^\top \in \mathbb{R}^{n^2}$ where $\xi_\eta^i \in \mathbb{R}^n$ denotes the state of the following LTI system

$$\begin{aligned} \dot{\xi}_\eta^i &= A_L\xi_\eta^i + B_L d^i \\ \zeta_\eta^i &= \xi_\eta^i, \end{aligned} \tag{41}$$

and the regressor $\Phi_\eta = \Phi_\eta(\tau, \xi_x, \xi_\eta) \in \mathbb{R}^{r_3\times n}$ and the parameters $\Theta_\eta \in \mathbb{R}^{r_3}$ are defined as

$$\Phi_\eta = \begin{bmatrix} I_n \otimes (A_x\xi_x + B_x\tau) \\ I_n \otimes (A_x\xi_\eta + B_x d) \end{bmatrix}, \quad \Theta_\eta = \begin{bmatrix} \text{vec}(W_x) \\ \text{vec}(W_\eta) \end{bmatrix}. \tag{42}$$

Then, the estimated model has the following structure

$$\begin{aligned} \dot{z} =\;& \dot{\tau} + \widehat{W}_x(A_x\xi_x + B_x\tau) - \widehat{W}_y(A_y\xi_w + B_y w) \\ & + \widehat{W}_\eta(A_x\xi_\eta + B_x d) \\ =\;& \dot{\tau} - \Phi_y^\top \widehat{\Theta}_y + \Phi_\eta^\top \widehat{\Theta}_\eta, \\ w =\;& Cz, \end{aligned} \tag{43}$$

where $\widehat{W}_\eta \in \mathbb{R}^{n\times n^2}$ are estimates of W_η and the parameter estimates vector $\widehat{\Theta}_\eta \in \mathbb{R}^{r_3}$ is defined as

$$\widehat{\Theta}_\eta = \begin{bmatrix} \text{vec}(\widehat{W}_x) \\ \text{vec}(\widehat{W}_\eta) \end{bmatrix} \in \mathbb{R}^{r_3}.$$

So, the CLOE dynamics between (40) and (43) is

$$\begin{aligned} \dot{e} =& C\Big(W_x(A_x\xi_x + B_x\tau) - W_y(A_y\xi_y + B_y y_\tau) \\ & + W_\eta(A_x\xi_\eta + B_x d) - \widehat{W}_x(A_x\xi_x + B_x\tau) \\ & + \widehat{W}_y(A_y\xi_w + B_y w) - \widehat{W}_\eta(A_x\xi_\eta + B_x d)\Big) \\ =& -C\Big(W_y B_y e - \Phi_z^\top \widetilde{\Theta}_y + \Phi_\eta^\top \widetilde{\Theta}_\eta + \varepsilon\Big) \end{aligned} \tag{44}$$

Fig. 2 depicts the diagram of the proposed CLOE algorithm for unknown desired reference/destination. The diagram is almost the same as the diagram of Fig. 1 where another subsystem composed of n linear systems $\dot{\xi}_\eta$ is added to feed the identification algorithm that subsequently updates the estimated model estimates.

Theorem 1 holds for the unknown reference case where the update rule (32) is changed to

$$\dot{\widetilde{\Theta}}_\eta = \Gamma_\eta\Phi_\eta C^\top e, \tag{45}$$

where $\Gamma_\eta \in \mathbb{R}^{r_3\times r_3}$ is a diagonal gain matrix and Φ_η fulfils the PE condition (30).

Remark 2: The matrix η can include unknown and bounded exogenous disturbances without modifying the state parameterization (40).

Notice that if the output measurements y are noise-free and the regressors Φ_y, Φ_x or Φ_η fulfil the PE condition (30), then $\varepsilon = 0$. So, the output error e converges to zero.

Remark 3: For the proposed method, the involved computation is dominated by the CLOE identification rule to estimate either $\widehat{\Theta}_x$, $\widehat{\Theta}_y$, or $\widehat{\Theta}_\eta$. If one does the calculations of the right hand side of (32) and (33), for $\widehat{\Theta}_x$ has growth with $(r_1 p)^2 n$ due to the Kronecker products, and for $\widehat{\Theta}_y$ has growth with $(r_2 p)^2 n$. Thus, the complexity is given by $\mathcal{O}\left(n((r_1 p)^2 + (r_2 p)^2)\right)$. The same procedure is applied to update rule (45), the CLOE with unknown desired destination one has growth with $(r_3 p)^2 n$, so the complexity is given by $\mathcal{O}(n((r_2 p)^2 + (r_3 p)^2))$.

VI. Simulation Studies

The performance of the proposed trajectory inference algorithm are assessed using a F-16 aircraft dynamics, a 2-DOF robot manipulator, and a quadcopter drone model. A sampling period of 0.1ms and the ODE5 solver are used to obtain more accurate approximations of the continuous-time update laws.

A. F-16 aircraft dynamics

Consider a F-16 short period dynamics [54]

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \\ y &= Cx, \quad C = [I_2, 0_{2\times 1}]. \end{aligned}$$

Here, $x = [\alpha, q, \delta_e]^\top$, where α is the angle of attack, q is the pitch rate and δ_e is the elevator deflection angle, and $u = \delta_{ec}$ is the elevator command. Consider a stabilization task at the origin. Gaussian random noise is added at the output measurements to model sensor noise. In addition, an excitation

signal τ is added to guarantee parameter estimate convergence. This signal is composed by an exponential weighted sum of sinusoidal functions of different frequencies. It is clear that the pair (A, B) is controllable and the pair (A, C) is observable.

Assume that the F-16 dynamics is controlled by any unknown feedback controller of the form $u = -Kx$ where $K \in \mathbb{R}^{1\times 3}$ is a stabilizing gain. So, since K is unknown we are able to choose any stable characteristic polynomial. It is proposed to locate the system's poles at $-5, -6, -4$, that is, $D(s) = (s+6)(s+5)(s+4)$.

Remark 4: On the one hand, if the poles are chosen to be far from the origin in the left-side of the complex plane, it implies that the control gain K is large and hence the output gain L will increase too. Conversely, if the poles are located near the origin in the left-side of the complex plane, then the control gain K and L will be small.

The estimated model (25) of the F-16 aircraft dynamics is

$$\dot{z} = \exp^{\mathcal{A}t} z_0 + \dot{\tau} + \widehat{W}_x(A_x\xi_x + B_x\tau) - \widehat{W}_y(A_y\xi_w + B_y w),$$

where $\widehat{W}_x \in \mathbb{R}^{3\times 9}$ and $\widehat{W}_y \in \mathbb{R}^{3\times 6}$. So we need to estimate 27 parameters for $\widehat{\Theta}_x$ and 18 parameters for $\widehat{\Theta}_y$. The gains of the update law are manually tuned until the best performance is achieved. The final gains are $\Gamma_x = 700 I_{27}$ and $\Gamma_y = 1000 I_{18}$. The initial condition term is computed as

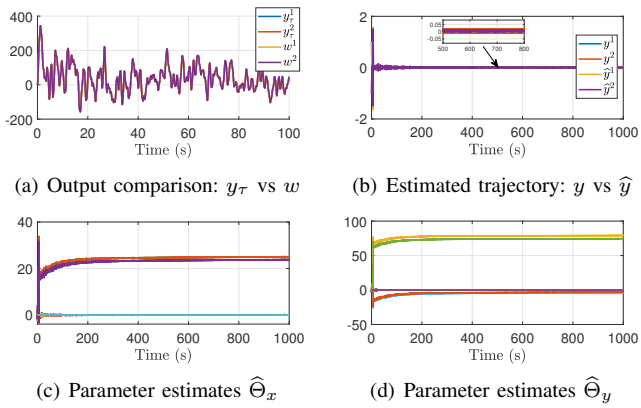
$$e^{\mathcal{A}t}x_0 = Pe^{\mathcal{D}t}P^{-1}x_0$$

with $x_0 = z_0 = [1, -1, 1]^\top$, $\mathcal{D} = -\text{diag}\{4, 5, 6\}$, and

$$P = \begin{bmatrix} 0.0605 & -0.0392 & -0.0274 \\ -0.2421 & 0.1960 & 0.1643 \\ 0.9684 & 0.9798 & -0.9860 \end{bmatrix}.$$

Fig. 3 shows the trajectory inference results. Fig. 3(a) shows an accurate tracking of the real system's trajectory with excitation signal. The real trajectory (without excitation signal) was recovered by computing $\widehat{y} = w - \tau$, the results are shown in Fig. 3(b). Notice that the estimated trajectories exhibit a smooth response in comparison to the noisy signal y_τ. Parameter estimates convergence of both $\widehat{\Theta}_x$ and $\widehat{\Theta}_y$ are shown in Fig. 3(c) and Fig. 3(d). The design of the excitation signal τ is one of the main challenges of any identification algorithm because parametric drift may exist. Notice that the proposed excitation signal guarantees parameter estimates convergence but is slow.

Fig. 4 shows the output error results. The output error is equivalent to the innovation of a Kalman filter algorithm where the outcome is only the noise of the noisy-measurements which implies that the estimated trajectory can infer the real trajectory and reduce the level of noise.



(a) Output comparison: y_τ vs w (b) Estimated trajectory: y vs $\widehat{y}$

(c) Parameter estimates $\widehat{\Theta}_x$ (d) Parameter estimates $\widehat{\Theta}_y$

Fig. 3. Trajectory Inference results: F-16 aircraft case

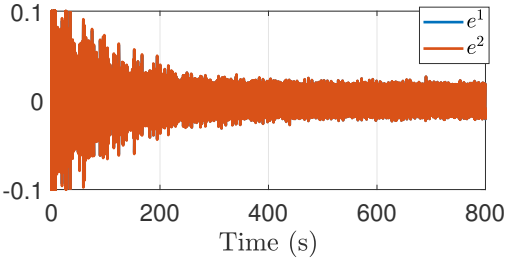


Fig. 4. Output error trajectories

B. 2-DOF robot

Consider a 2-DOF robot [56] endowed with gearbox such that its dynamics can be expressed as a perturbed linear system of the form

$$\ddot{q} = -A_q\dot{q} + B_q u + D_q,$$

where $q = [q^1, q^2]^\top \in \mathbb{R}^2$ denotes the joint positions, $u \in \mathbb{R}^2$ is the control input associated to the voltage applied to the actuators, $A_q = A_q^\top > 0 \in \mathbb{R}^{2\times 2}$ stands for a dissipative matrix, $B_q = B_q^\top > 0 \in \mathbb{R}^{2\times 2}$ is a control matrix gain, and $D_q \in \mathbb{R}^2$ is a disturbance vector. Consider that the robot is controlled by a PD controller of the form

$$u = K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) + B_q^{-1}(\ddot{q}_d + A_q\dot{q}_d),$$

where $K_p, K_d > 0 \in \mathbb{R}^{2\times 2}$ denote the proportional and derivative control gains, and $q_d, \dot{q}_d, \ddot{q}_d \in \mathbb{R}^2$ are the desired reference and their respective time-derivatives. So, the closed-loop dynamics between the robot model and the PD controller is given by

$$\ddot{q} = \ddot{q}_d + B_q K_p(q_d - q) + (A_q + B_q K_d)(\dot{q}_d - \dot{q}) + D_q.$$

In most cases, the terms $B_q^{-1}(\cdot)$ in u are not used in real control laws because the control gains are chosen to be large enough such that the disturbance vector D_q and the terms associated to the time-derivatives of the desired reference are attenuated. This means, that the closed-loop trajectories have practical stability or equivalently they are UUB.

The above dynamics written in state space is

$$\dot{x} = -Ax + Ax_d + \dot{x}_d + D,$$

with $x = [q^\top, \dot{q}^\top]^\top \in \mathbb{R}^4$, $x_d = [q_d^\top, \dot{q}_d^\top]^\top \in \mathbb{R}^4$, and $A > 0 \in \mathbb{R}^{4\times 4}$, and $D \in \mathbb{R}^4$ are given by

$$A = \begin{bmatrix} 0_2 & I_2 \\ B_q K_p & A_q + B_q K_d \end{bmatrix}, \quad D = \begin{bmatrix} 0_{2\times 1} \\ D_q \end{bmatrix}.$$

Assume that we only have measurements of the joint angles, so the output matrix is $C = [I_2, 0_2]$.

For this case, the state parameterization (40) is used because the dynamics possesses a disturbance vector. Assume that

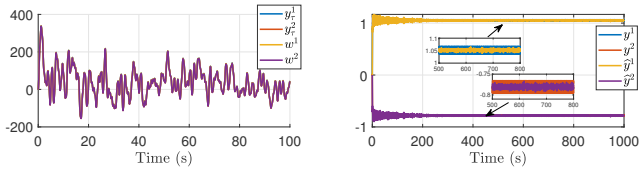
the desired reference x_d is known in advance. The estimated model has the following structure

$$\begin{aligned}\dot{z} =& e^{\mathcal{A}t}z_0 + \dot{x}_d + \dot{\tau} + \widehat{W}_x(A_x\xi_x + B_x(x_d+\tau)) \\ & - \widehat{W}_y(A_y\xi_y + B_y w) + \widehat{W}_\eta(A_x\xi_\eta + B_x d) \\ =& \dot{x}_d + \dot{\tau} - \Phi_y^\top \widehat{\Theta}_y + \Phi_\eta^\top \widehat{\Theta}_\eta,\end{aligned}$$

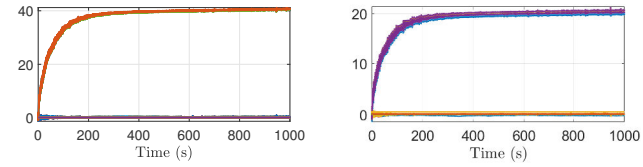
where $\widehat{W}_x \in \mathbb{R}^{4\times 16}$, $\widehat{W}_y \in \mathbb{R}^{4\times 8}$, and $\widehat{W}_\eta \in \mathbb{R}^{4\times 16}$. So, we have to estimate 128 parameters for $\widehat{\Theta}_\eta$, and 32 parameters for $\widehat{\Theta}_y$. The same excitation signal is used for the trajectory inference algorithm. The system's poles are located at $-4,-5,-6,-7$, that is, $D(s) = (s+4)(s+5)(s+6)(s+7)$. The initial condition is $x_0 = z_0 = [0.1, 0.1, 0, 0]^\top \in \mathbb{R}^4$, $\mathcal{D} = -\text{diag}\{7,6,5,4\}$, and

$$e^{\mathcal{A}t}z_0 = Pe^{\mathcal{D}t}P^{-1}z_0$$

$$P = \begin{bmatrix} -0.0029 & 0.0046 & -0.0078 & 0.0151 \\ 0.0202 & -0.0274 & 0.0392 & -0.0605 \\ -0.1414 & 0.1643 & -0.1960 & 0.2421 \\ 0.9897 & -0.9860 & 0.9798 & -0.9683 \end{bmatrix}.$$



(a) Output comparison: y_τ vs w (b) Estimated trajectory: y vs $\widehat{y}$



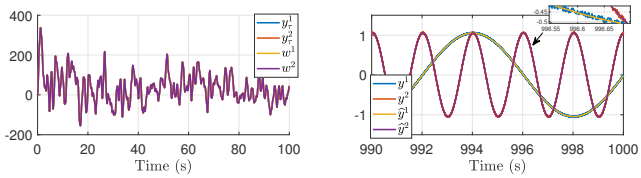
(c) Parameter estimates $\widehat{\Theta}_y$ (d) Parameter estimates $\widehat{\Theta}_\eta$

Fig. 5. Trajectory Inference results: 2-DOF robot case with known constant reference

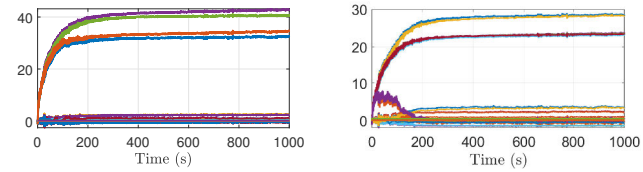
The update rule gains are manually tuned until the best identification performance is achieved. The final matrices are $\Gamma_y = 2000 I_{32}$ and $\Gamma_\eta = 1500 I_{128}$. First, the following constant desired references are used $q_d^1 = \frac{\pi}{3}$, $q_d^2 = -\frac{\pi}{4}$.

The trajectory inference results are shown in Fig. 5. Similarly to the F-16 aircraft results (see Fig. 3), the inference algorithm is able to extract the real trajectory and reduce the noise. Parameter estimates convergence of both $\widehat{\Theta}_y$ and $\widehat{\Theta}_\eta$ are also achieved. Here, the excitation signal τ is not applied to the real system dynamics, instead is an external signal added to the output measurements for identification purposes. Thus, we are able to use any excitation signal τ with different frequencies and amplitudes to guarantee parameters convergence.

The constant desired reference is modified to the following time-varying reference, $q_d^1(t) = \frac{\pi}{3}\sin\left(\frac{\pi}{4}t\right)$, $q_d^2(t) = \frac{\pi}{3}\cos(\pi t)$. Fig. 6 shows the trajectory inference results for a time-varying reference. Similar results to the constant reference case (see Fig. 5) are obtained. Noise attenuation and parameter estimates convergence are achieved by using the prior knowledge of the desired reference and the proposed excitation signal. In addition, a time-varying reference adds a new way to excite the identification algorithm



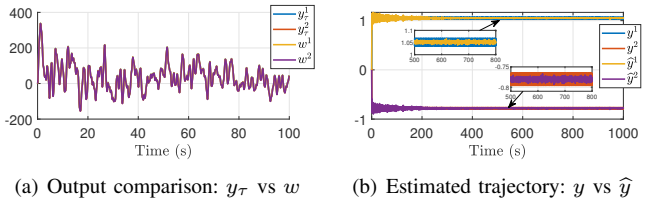
(a) Output comparison: y_τ vs w (b) Estimated trajectory: y vs $\widehat{y}$



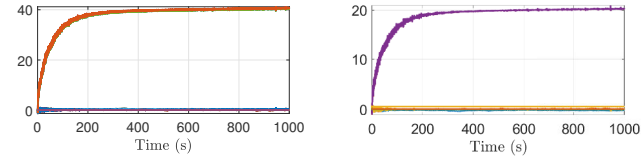
(c) Parameter estimates $\widehat{\Theta}_y$ (d) Parameter estimates $\widehat{\Theta}_\eta$

Fig. 6. Trajectory Inference results: 2-DOF robot case with known time-varying reference

Finally, assume that the desired reference is unknown, that is, we only have available the output measurements y and the proposed excitation signal τ. Fig. 7 shows the obtained inference results for the constant desired reference case. Notice that it is obtained similar results to the ones obtained in Fig. 3. Here the desired reference x_d is estimated in $\widehat{\Theta}_\eta$ such that almost the same performance is achieved.



(a) Output comparison: y_τ vs w (b) Estimated trajectory: y vs $\widehat{y}$

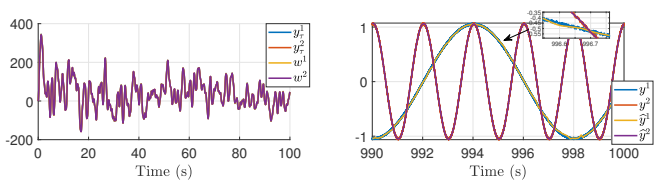


(c) Parameter estimates $\widehat{\Theta}_y$ (d) Parameter estimates $\widehat{\Theta}_\eta$

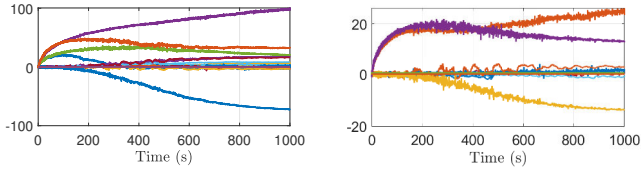
Fig. 7. Trajectory Inference results: 2-DOF robot case with unknown constant reference

Fig. 8 shows the trajectory inference results for an unknown time-varying reference. In contrast to the previous cases, an unknown time-varying reference causes that the estimates of $\widehat{\Theta}_\eta$ to be time-variant. This modifies the output error e, and consequently affects the estimates of $\widehat{\Theta}_y$. Despite the estimated trajectory was noise-free and achieve good inference results, the parameter estimates cannot converge. Further work will extend the proposed approach for LTV and nonlinear systems.

The norm of the output error $\bar{e}_i = \|ke_i\|$, $i = 1, 2$ of the last 200 seconds of simulation time is used as a performance metric of the proposed approach, where k is a scaling factor. Table I summarizes the numerical results of the output error norm using a scaling factor of $k = 100$.



(a) Output comparison: y_τ vs w (b) Estimated trajectory: y vs $\widehat{y}$



(c) Parameter estimates $\widehat{\Theta}_y$ (d) Parameter estimates $\widehat{\Theta}_\eta$

Fig. 8. Trajectory Inference results: 2-DOF robot case with unknown time-varying reference

TABLE I
OUTPUT ERROR NORM RESULTS

Reference x_d	**Constant reference**		**Time-varying reference**	
	$\bar{e}_1$	$\bar{e}_2$	$\bar{e}_1$	$\bar{e}_2$
Known	404.2606	404.2652	399.9450	400.0409
Unknown	404.2611	404.2660	659.573	497.8619

Table I clearly shows that almost the same output error results is achieved for a constant desired reference (either known and unknown). In contrast, the output error increases for unknown time-varying references since the parameter estimates become time-variant.

C. Drone model

A drone model is used to further test the proposed approach. Consider the following drone linear model

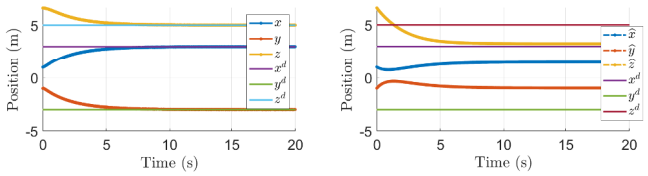
$$\dot{X} = \begin{bmatrix} 0_3 & I_3 \\ 0_3 & 0_3 \end{bmatrix} X + \begin{bmatrix} 0_{3\times1} & 0_{3\times1} & 0_{3\times1} \\ 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & \frac{1}{m} \end{bmatrix} u,$$
$$Y = \begin{bmatrix} I_3 & 0_3 \end{bmatrix} X,$$

where $X = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^\top \in \mathbb{R}^6$ denotes the vector of linear positions and velocities, Y is the output vector, $u = [\phi, \theta, \mu]^\top \in \mathbb{R}^3$ defines the control input given by the roll and pitch Euler angles and the total thrust, respectively; $m = 0.467$kg is the mass of the drone and $g = 9.81$m/s^2 is the gravity acceleration. This linear model is obtained around the hover flight condition [57]. Each state is corrupted by a low amplitude Gaussian noise with zero mean and standard deviation of 0.1. The drone is controlled by a liner quadratic regulator (LQR) controller that ensures the small angle condition. The initial condition is $X_0 = [1, -1, mg + 2, 0, 0, 0]^\top \in \mathbb{R}^6$. The desired reference is $X^d = [3, -3, 5, 0, 0, 0]^\top$.

First, we aim to show the main problem of model-based approaches for trajectory inference such as Kalman filter. Dynamic mode decomposition (DMD) [58] is used to obtain a linear matrix from data collected of a random and rich mission profile. The final estimated matrix is

$$\widehat{A} = \begin{bmatrix} -0.5172 & 0.0618 & -0.0212 & 0.4891 & -0.2033 & 0.0374 \\ 0.0155 & -0.6329 & 0.1055 & 0.0150 & 0.4398 & 0.0119 \\ -0.0287 & 0.1604 & -0.2172 & -0.1267 & -0.0423 & 0.5105 \\ 0.0922 & 0.2043 & 0.1058 & -7.6491 & 0.7124 & 0.0349 \\ -0.0773 & -0.0554 & -0.0212 & 0.8802 & -4.6677 & 0.2985 \\ 0.0293 & 0.1427 & -0.1908 & -0.1169 & 0.2121 & -0.4928 \end{bmatrix}.$$

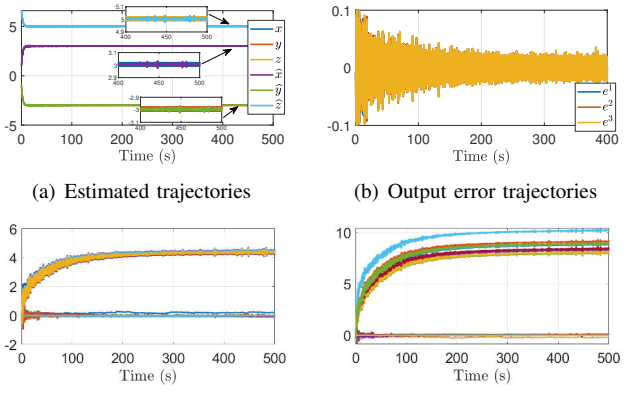
Matrix $\widehat{A}$ is used to compute a Kalman filter algorithm with covariance matrices $Q = 1I_6$ and $R = 0.5I_3$. The results of the trajectory inference using the standard Kalman filter are shown in Fig. 9. It is observed that the estimated linear matrix is not accurate due to the presence of noise in the state measurements and the lack of information of the control input. In consequence, the estimated trajectories are not precise and require fine tuning of either the state space matrices or the covariance matrices. Another alternative is to use data-driven approaches, but they require a large amount of data and access to the control input signals in order to be applied online which violates Assumption 3.



(a) Real trajectories (b) Estimated trajectories

Fig. 9. Kalman filter trajectory inference results

In view of the above, we aim to show the benefits of our CLOE trajectory inference algorithm. The drone's poles are located at $-4, -5, -6, -7, -8, -9$, that is, $D(s) = (s+4)(s+5)(s+6)(s+7)(s+8)(s+9)$. The initial condition is computed similarly to the F-16 and 2-DOF robot models. The update rule gains are manually tuned until the best identification performances is achieved. The final matrices are $\Gamma_y = 500I_{216}$ and $\Gamma_\eta = 500I_{108}$. The inference results are given in Fig. 10.



(a) Estimated trajectories (b) Output error trajectories

(c) Parameter estimates $\widehat{\Theta}_y$ (d) Parameter estimates $\widehat{\Theta}_\eta$

Fig. 10. Trajectory inference results: Drone case with unknown constant reference

Similar results are obtained for the drone model, that is, accurate trajectory inference with noise attenuation and parameter estimates convergence. One important point to observe is that the number of parameters to estimate increases as the

number of states of the system increases. Here, the number of parameters of the real model can be arbitrarily large or small since we are not considering a fixed parametric model structure as in standard identification algorithms, e.g., least squares methods. On the other hand, the proposed technique outperforms Kalman filter when an inaccurate prior model is available.

VII. Conclusions

This paper reports a trajectory inference algorithm based on a CLOE approach using partial states measurements. The algorithm combines the main advantages of a state estimation and identification algorithms to estimate both the states and parameter estimates simultaneously. A physics-informed state-parameterization model is used to express controllable and observable linear system in terms of the desired reference and the output measurements. Known and unknown desired reference cases are analysed. The stability and convergence of the proposed approach are rigorously assessed using Lyapunov stability theory. Simulations verify the proposed approach with good trajectory inference results and noise attenuation.

The number of parameter estimates increases as the number of states and output measurements increases even for a system with low number of states. Further work will investigate the dimensionality reduction of the number of parameter estimates to be computed. In addition, the algorithm requires a good excitation signal design for parameter estimates convergence. Further work will also investigate the adequate design of excitation signals to improve convergence time and guarantee parameter estimates convergence. Furthermore, the extension of the proposed approach for LTV and nonlinear systems are the main concern for our future work.

References

[1] A. Janot, P.-O. Vandanjon, and M. Gautier, "Identification of physical parameters and instrumental variables validation with two-stage least squares estimator," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1386–1393, 2012.

[2] O. Nelles, "Nonlinear dynamic system identification," in *Nonlinear System Identification*. Springer, 2020, pp. 831–891.

[3] 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.

[4] A. Perrusquía and W. Guo, "A closed-loop output error approach for physics-informed trajectory inference using online data," *IEEE Transactions on Cybernetics*, vol. 53, no. 3, pp. 1379–1391, 2023.

[5] W. Gilpin, "Chaos as an interpretable benchmark for forecasting and data-driven modelling," *35th Conference on Neural Information Processing Systems (NeurIPS'21)*, 2021.

[6] E. Archer, I. M. Park, L. Buesing, J. Cunningham, and L. Paninski, "Black box variational inference for state space models," *International Conference on Learning Representations*, 2016.

[7] G. Revach, N. Shlezinger, R. J. Van Sloun, and Y. C. Eldar, "Kalmannet: Data-driven kalman filtering," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3905–3909.

[8] C.-T. Chen, *Linear System Theory and Design*. Oxford University Press, 1999.

[9] Z. Peng, Y. Jiang, L. Liu, and Y. Shi, "Path-guided model-free flocking control of unmanned surface vehicles based on concurrent learning extended state observers," *IEEE Transactions on Systems, Man, an Cybernetics: Systems*, 2023.

[10] H. K. Khalil and L. Praly, "High-gain observers in nonlinear feedback control," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 6, pp. 993–1015, 2014.

[11] S. K. Spurgeon, "Sliding mode observers: a survey," *International Journal of Systems Science*, vol. 39, no. 8, pp. 751–764, 2008.

[12] R. Corteso, J. Park, and O. Khatib, "Real-time adaptive control for haptic telemanipulation with kalman active observers," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 987–999, 2006.

[13] L. B. White and F. Carravetta, "State-space realizations and optimal smoothing for gaussian generalized reciprocal processes," *IEEE Transactions on Automatic Control*, vol. 65, no. 1, pp. 389–396, 2019.

[14] R. Rezaie and X. R. Li, "Gaussian conditionally markov sequences: Modeling and characterization," *Automatica*, vol. 131, p. 109780, 2021.

[15] L. Martino, J. Read, V. Elvira, and F. Louzada, "Cooperative parallel particle filters for online model selection and applications to urban mobility," *Digital Signal Processing*, vol. 60, pp. 172–185, 2017.

[16] J. Elfring, E. Torta, and R. van de Molengraft, "Particle filters: A hands-on tutorial," *Sensors*, vol. 21, no. 2, p. 438, 2021.

[17] 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.

[18] L. Xu and R. Niu, "Ekfnet: Learning system noise statistics from measurement data," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4560–4564.

[19] C. Legaard, T. Schranz, G. Schweiger, J. Drgoňa, B. Falay, C. Gomes, A. Iosifidis, M. Abkar, and P. Larsen, "Constructing neural network based models for simulating dynamical systems," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–34, 2023.

[20] O. Ogunmolu, X. Gu, S. Jiang, and N. Gans, "Nonlinear systems identification using deep dynamic neural networks," *American Control Conference*, 2017.

[21] M. Lutter, C. Ritter, and J. Peters, "Deep lagrangian networks: Using physics as model prior for deep learning," *International Conference on Learning Representations*, 2019.

[22] B. Chang, L. Meng, E. Haber, F. Tung, and D. Begert, "Multi-level residual networks from dynamical systems view," *International Conference on Learning Representations*, 2018.

[23] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Advances in neural information processing systems*, vol. 31, 2018.

[24] A. Perrusquía and W. Yu, "Discrete-time $\mathcal{H}_2$ neural control using reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 11, pp. 4879 – 4889, 2021.

[25] X. Zheng, M. Zaheer, A. Ahmed, Y. Wang, E. P. Xing, and A. J. Smola, "State space lstm models with particle mcmc inference," *arXiv preprint arXiv:1711.11179*, 2017.

[26] P. Becker, H. Pandya, G. Gebhardt, C. Zhao, C. J. Taylor, and G. Neumann, "Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces," in *International Conference on Machine Learning*. PMLR, 2019, pp. 544–552.

[27] A. Perrusquía, "A complementary learning approach for expertise transference of human-optimized controllers," *Neural Networks*, vol. 145, pp. 33–41, 2022.

[28] J. Zhao, C. Yang, W. Gao, and L. Zhou, "Reinforcement learning and optimal setpoint tracking control of linear systems with external disturbances," *IEEE Transactions on Industrial Informatics*, 2022.

[29] S. Cao, L. Sun, J. Jiang, and Z. Zuo, "Reinforcement learning-based fixed-time trajectory tracking control for uncertain robotic manipulators with input saturation," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[30] M.-B. Radac and T. Lala, "Hierarchical cognitive control for unknown dynamic systems tracking," *Mathematics*, vol. 9, no. 21, p. 2752, 2021.

[31] A. Perrusquía and W. Yu, "Neural $\mathcal{H}_2$ control using continuous-time reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 4485–4494, 2022.

[32] S. A. A. Rizvi and Z. Lin, "Output feedback q-learning control for the discrete-time linear quadratic regulator problem," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 5, pp. 1523–1536, 2018.

[33] T. Yamammoto, M. Bernhardt, A. Peer, M. Buss, and A. M. Okammura, "Techniques for Environment parameter Estimation during telemanipulation," *Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2008.

[34] W. Yu and A. Perrusquía, *Human-Robot Interaction Control Using Reinforcement Learning*. Wiley-IEEE Press, 2022.

[35] I. D. Landau and A. Karimi, "An output error recursive algorithm for unbiased identification in closed loop," *Automatica*, vol. 33, no. 5, 1997.

[36] A. Janot, P.-O. Vandanjon, and M. Gautier, "A generic instrumental variable approach for industrial robot identification," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 1, pp. 132–145, 2013.

[37] A. Perrusquía and W. Guo, "Closed-loop output error approaches for drone's physics informed trajectory inference," *IEEE Transactions on Automatic Control*, 2023.

[38] S. Zeng and E. Fernandez, "Adaptive controller design and disturbance attenuation for sequantially interconnected SISO linear systems under noisy output measurements with partly measured disturbances," *2008 American Control Conference*, 2008.

[39] D. Li, B. Zhang, P. Li, E. Q. Wu, R. Law, X. Xu, A. Song, and L.-M. Zhu, "Parameter estimation and anti-sideslip line-of-sight method-based adaptive path-following controller for a multijoint snake robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.

[40] C. Urrea and J. Pascal, "Design and validation of a dynamic parameter identification model for industrial manipulator robots," *Archive of Applied Mechanics*, pp. 1–27, 2021.

[41] M. Brunot, A. Janot, P. C. Young, and F. Carrillo, "An improved instrumental variable method for industrial robot model identification," *Control Engineering Practice*, vol. 74, pp. 107–117, 2018.

[42] M. Brunot, A. Janot, F. Carrillo, J. Cheong, and J.-P. Noël, "Output error methods for robot identification," *Journal of Dynamic Systems, Measurement, and Control*, vol. 142, no. 3, 2020.

[43] I. D. Landau and A. Karimi, "An output error recursive algorithm for unbiased identification in closed loop," *Automatica*, vol. 33, no. 5, pp. 933–938, 1997.

[44] A. Janot, M. Gautier, A. Jubien, and P. O. Vandanjon, "Comparison between the cloe method and the didim method for robots identification," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1935–1941, 2014.

[45] A. Perrusquía, R. Garrido, and W. Yu, "Stable robot manipulator parameter identification: A closed-loop input error approach," *Automatica*, vol. 141, p. 110294, 2022.

[46] A. Perrusquia, J. A. Flores-Campos, C. R. Torres-Sanmiguel, and N. Gonzalez, "Task space position control of slider-crank mechanisms using simple tuning techniques without linearization methods," *IEEE Access*, vol. 8, pp. 58 435–58 442, 2020.

[47] H. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[48] A. Perrusquía, "Solution of the linear quadratic regulator problem of black box linear systems using reinforcement learning," *Information Sciences*, vol. 595, pp. 364–377, 2022.

[49] S. A. A. Rizvi and Z. Lin, "Reinforcement learning-based linear quadratic regulation of continuous-time systems using dynamic output feedback," *IEEE transactions on cybernetics*, vol. 50, no. 11, pp. 4670–4679, 2019.

[50] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for machine learning*. Cambridge University Press, 2020.

[51] F. Lewis, S. Jagannathan, and A. Yeşildirek, *Neural Network control of robot manipulators and nonlinear systems*. Taylor & Francis, 1999.

[52] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Systems & Control Letters*, pp. 14–20, 2017.

[53] A. Perrusquía, W. Yu, and X. Li, "Nonlinear control using human behavior learning," *Information Sciences*, vol. 569, pp. 358–375, 2021.

[54] K. Vamvoudakis and F. L. Lewis, "On-line actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, pp. 878–888, 2010.

[55] A. Perrusquía, "Robust state/output feedback linearization of direct drive robot manipulators: A controllability and observability analysis," *European Journal of Control*, 2022.

[56] A. Perrusquía and W. Yu, "Continuous-time reinforcement learning for robust control under worst-case uncertainty," *International Journal of Systems Science*, vol. 52, no. 4, pp. 770–784, 2021.

[57] 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.

[58] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.



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.



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.



https://dspace.org/

Trajectory inference of unknown linear systems based on partial states measurements

Perrusquía, Adolfo; Guo, Weisi

2024-04-01
Institute of Electrical and Electronics Engineers (IEEE)
https://doi.org/10.1109/TSMC.2023.3344017
Perrusquía A, Guo W. (2024) Trajectory inference of unknown linear systems based on partial states measurements. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Volume 54, Issue 4, April 2024, pp. 2276-2286

Downloaded from a DSpace repository. https://dspace.org/