Time-Varying Human Operator Identification With Box-Jenkins Models

Ortiz Moya, Álvaro; Pool, D.M.; van Paassen, M.M.; Mulder, Max

DOI
10.2514/6.2025-2476

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the AIAA SCITECH 2025 Forum

AIAA SciTech Forum
6-10 January 2025, Orlando, FL
AIAA SCITECH 2025 Forum

10.2514/6.2025-2476

Time-Varying Human Operator Identification with Box-Jenkins Models

Alvaro Ortiz Moya*, Daan M. Pool†, Marinus M. van Paassen‡ and Max Mulder§
Delft University of Technology, Delft, Zuid-Holland, The Netherlands

The identification of time-varying, adaptive behavior of a human operator in basic manual control tasks is currently still a focus area, since most methodologies only account for time-invariant system dynamics. Previous authors have proven that estimation techniques based on ARX model structures can be used to identify time-varying HO model parameters. However, ARX methods do present several problems, such as a persistent bias in the obtained estimates of the HO model poles (neuromuscular parameters) that increases due to coupled noise and system models. Therefore, in this paper a novel identification technique based on Box-Jenkins (BJ) models is proposed, to achieve a better match between the BJ estimator's inherently uncoupled system and noise models and measured HO control dynamics. The identification process was tested offline (batch-fitting) using Ordinary Least Squares and the Prediction Error Method for both ARX and BJ models, respectively, or online when Recursive Least Squares and Recursive PEM are employed. The BJ estimator has excellent potential as an identification tool due to its bias reduction capabilities, as clearly shown in batch-fitting, although the non-linear optimization processes decrease its convergence speed by 500%. An RPEM algorithm with a forgetting factor of $\lambda = 0.99609$ and a first-order remnant model incorporated in the BJ structure was tested on Monte Carlo simulation and experimental data. While the recursive BJ estimator showed the same bias-diminishing advantages also seen in batch-fitting, the non-linear RPEM estimator's results showed much slower convergence after HO behavior adaptations and frequent instabilities of the obtained parameter estimates. Hence, further research is needed for implementing a practical bias-free HO model estimator based on the BJ model structure.

Nomenclature

Latin letters

$A(z^{-1})$	BJ model's HO output polynomial
A_k	Sinusoid amplitude, deg
a_i	Coefficient i of $A(z^{-1})$
$B(z^{-1})$	BJ model's HO input polynomial
$B_r(\vartheta)$	Relative bias for parameter ϑ, %
b_i	Coefficient i of $B(z^{-1})$
$C(z^{-1})$	BJ model's remnant input polynomial
c_i	Coefficient i of $C(z^{-1})$
$D(z^{-1})$	BJ model's remnant output polynomial
d_i	Coefficient i of $D(z^{-1})$
$e(t)$	Error signal, deg
$f_t(t)$	Forcing function, deg
G	Maximum rate of change in sigmoid, s^{-1}
$g(t_k)$	Gain vector
$H_{CE}(s,t)$	Time-varying CE dynamics
$H_{HO}(s,t)$	Time-varying HO dynamics
$H_{HO_e}(s,t)$	Time-varying HO's linear response to $e(t)$
$H_{nm}(s,t)$	Neuromuscular dynamics
$H_n^m(s,t)$	Time-varying remnant filter of order m
$K_e(t)$	Error gain
$K_n(t)$	Remnant gain
$K_p(t)$	Control gain
M	Monte Carlo replications
m	Order of remnant filter
N	Number of samples
N_e	Memory horizon in samples for λ
N_t	Number of sinusoids

*M.Sc. student (graduated), Control and Simulation section, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, The Netherlands; alvrtmoya@gmail.com.

†Assistant Professor, Control and Simulation section, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, The Netherlands; d.m.pool@tudelft.nl. Associate Fellow AIAA.

‡Associate Professor, Control and Simulation section, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, The Netherlands; m.m.vanpaassen@tudelft.nl. Member AIAA.

§Professor, Control and Simulation section, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, The Netherlands; m.mulder@tudelft.nl. Associate Fellow AIAA.

$n(t)$	Remnant signal, deg
n_a	Number of coefficients in $A(z^{-1})$
n_b	Number of coefficients in $B(z^{-1})$
n_c	Number of coefficients in $C(z^{-1})$
n_d	Number of coefficients in $D(z^{-1})$
n_k	HO model's integer time delay
n_t	Integer in sinusoid k
$P(t_k)$	Scaled covariance matrix
P_n	Remnant intensity level
$p(t)$	Parameter function
s	Laplace variable
T_e	Memory horizon for λ, s
$T_L(t)$	Lead-time constant, s
T_m	Total simulation time, s
T_n	Remnant-time constant, s
T_s	Sampling time, s
t	Continuous-time variable, s
t_k	Discrete-time variable, s
t_M	Time of maximum rate of change, s
$u(t)$	Control-output signal, deg
V^N	Cost function, deg^2
VAF	Variance Accounted For at BJ model
$x(t)$	System output signal, deg
z	Z-transform variable
Z^N	Data set with pairs $\{u(t_k), e(t_k)\}$, deg

Greek letters

$\varepsilon(t)$	Remnant Gaussian White noise, deg
$\epsilon(t)$	Prediction error, deg
$\zeta_{nm}(t)$	Neuromuscular damping ratio
θ	Parameter vector
ϑ	Global variable for BJ or HO parameter
λ	Forgetting factor
$\pi(t_k)$	Negative prediction error gradient
σ^2	Variance
τ_e	HO time delay, s
ϕ_k	Sinusoid phase shift, rad
φ_m	Phase margin, deg
ω	Frequency, rad/s
$\omega_b(t)$	Break frequency, rad/s
$\omega_c(t)$	Crossover frequency, rad/s
ω_k	Sinusoid frequency, rad/s
$\omega_{k,0}$	Sinusoid base frequency, rad/s
ω_{nm}	Neuromuscular frequency, rad/s

Superscripts

0	True
$*$	Simulated in BJ model
d	Discrete time
c	Continuous time

Subscripts

i	Initial
f	Final

I. Introduction

Human manual-control behaviour naturally changes over time, in various contexts, and across operators. In order to explain the dynamic features of human operators (HOs) in skill-based manual control tasks, identification methods have been developed [1–8]. However, they are often only applicable in situations when the control behaviour is sufficiently time-invariant [6]. The availability of control-theoretic models that can capture both the adaptive and learning aspects of manual-control behaviour has long been a goal [9–14]. Modern cybernetics is unable to fully explain how HOs modify their behaviour to deal with control-task changes. The continued development of time-varying identification techniques is necessary to make rapid progress in our knowledge of how people really interact with dynamic control systems [15].

The majority of research on identifying time-varying manual-control behaviour focuses on task variable changes, particularly those caused by variations in the dynamics of the controlled element (CE) [3, 9–13, 16]. Many research projects begin by examining single-axis compensatory control tasks and are based on the well-known crossover theory [3], assuming that those developed models are scaleable to scenarios with additional types of inputs to the HO or even multiple axes of control [13].

The purpose of this article is to study and develop a tool capable of identifying real-time (i.e. online) human control behaviour adaptation in a compensatory manual-control task. Real-time execution of these processes offers new possibilities in addition to advancing the earlier study objectives. For instance, pilot adaptation to anomalies in aircraft or controlled elements could be studied and predicted [11–14]. On the other hand, reduced attention or distraction in real-world control activities could be identified by continuously monitoring the operator through an update of a human operator model [17]. Additionally, this might allow for adaptive haptic feedback that matches the operator's present behaviour [18]. Online identification can assist in modifying experimental circumstances in real time to directly analyze adaptation behaviour or obtain desired haptic feedback characteristics in research with humans-in-the-loop.

Multiple authors have attempted to apply different techniques to achieve a successful online identification [7, 10, 18–23]. Nevertheless, most of the methodologies used are not able to provide ideal results due to the high difficulty of the problem. For instance, strategies based on recursive estimation of Auto-Regressive-eXogeneous (ARX) model structures [6, 10, 12, 24, 25] may fail in reducing the relative bias [10] of the predicted model, in spite of their outstanding

qualities in terms of low computational effort and straightforward estimation. Consequently, it is necessary to find an identification method that is truly capable of assuming this task, so that a novel estimation technique based on Box-Jenkins (BJ) structures is proposed. This paper lays the foundation for the development of a recursive BJ algorithm, which aims to solve the previous issues in the human behaviour identification process. In parallel, the performance of such a novel identification technique is compared to the ARX estimation outcome. Both methods are evaluated under a time-varying scenario through Monte Carlo simulation and with experimental data obtained by Van Grootheest et al. [10].

This article is structured as follows. The compensatory manual-control task and previous identification methods are described in Section II. The Box-Jenkins identification approach is detailed in Section III. Afterwards, Section IV discusses the required simulation conditions and their setup, while Section V presents an overview of the Monte Carlo simulation results obtained. Identification results for experimental data are presented in Section VI. Section VII discusses the applicability, limitations and impact of the research project. Final conclusions are drawn in Section VIII.

II. Compensatory Manual-Control Task

In a control task, the human operator is typically a multichannel, adaptive, learning, non-linear controller [6–8, 20, 24–26]. Although a general control-theoretic model of the human controller has not yet been discovered, validated models do exist for certain control tasks, as shown in [15, 27]. In particular, McRuer and Jex [3] proposed quasi-linear human operator models that still represent the state-of-the-art in HO modeling. These models are based on the Crossover Model theory [1, 3], which is only applicable to simple cases, i.e., single-channel tracking tasks with only a feedback path from visual perception (pure compensatory display). The quasi-linear models separate the additional unexplained behaviour by adding noise called 'remnant' n and capture the linear behaviour of the human controller in a descriptive transfer function $H_{HO_e}(s,t)$.

Figure 1 depicts the quasi-linear operator model embedded in a compensating tracking task [11]. The HO, represented by the model $H_{HO}(s,t)$, monitors and responds to the error $e(t)$ between a goal $f_t(t)$ and the output $x(t)$ of the CE dynamics $H_{CE}(s,t)$. The HO dynamics are composed of the remnant $n(t)$ and deterministic responses from $H_{HO_e}(s,t)$. Regarding the remnant $n(t)$, it is generated by feeding a white-noise signal $\varepsilon(t)$ with a determined statistical distribution through a remnant filter $H_n^m(s,t)$. Additionally, the HO must alter its control strategy as the CE dynamics change over time (CE adaptation) [9].

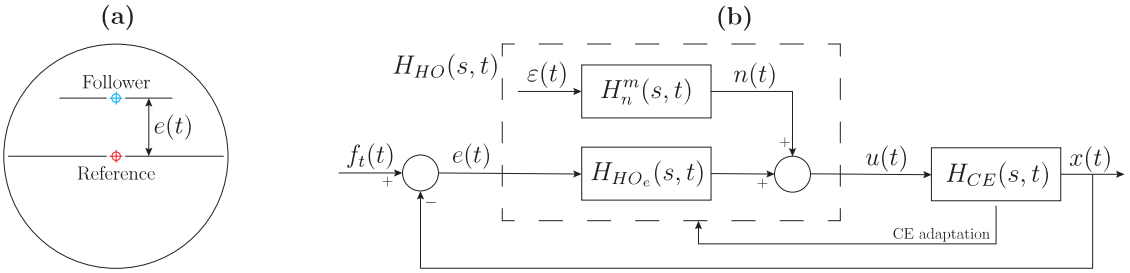

Fig. 1 Single-axis compensatory manual-control task with time-varying dynamics: (a) Compensatory display, where $e(t)$ acts as a stimulus, and (b) Block diagram.

A. Human-Operator Dynamics

In the Crossover model, McRuer and Jex [3] state that people modify their control behaviour to satisfy

$$H_{OL}(j\omega) = H_{HO_e}(s = j\omega)H_{CE}(j\omega) = \frac{\omega_c}{j\omega}e^{-j\omega\tau_e}, \omega \approx \omega_c \tag{1}$$

in the crossover zone when transitory behaviour is eliminated. Then adjustment rules define how the describing function $H_{HO_e}(s,t)$ behaves in relation to the controlled element $H_{CE}(s,t)$ and what impact it has on crossover frequency ω_c and time delay τ_e in the frequency domain [3, 9]. By modeling the neuromuscular system (NMS) as a second-order transfer function $H_{nm}(j\omega)$ and the operator equalization as a gain and a lead(L) [3–5, 11], while using the approach

stated in [10–12], the general formulation of the describing function can be defined as follows:

$$H_{HO_e}(s,t) = K_e(t)\,[T_L(t)s+1]\,e^{-s\tau_e}H_{nm}(j\omega) = \frac{K_e(t)\,[T_L(t)s+1]\,e^{-s\tau_e}\omega_{nm}^2}{s^2+2\zeta_{nm}\omega_{nm}s+\omega_{nm}^2}. \tag{2}$$

The equalization parameters, i.e. $K_e(t)$ and $T_L(t)$, determine the action of the feedback controller in the pilot-vehicle loop. On the other hand, ω_{nm} and ζ_{nm} model the neuromuscular dynamics.

Regarding possible remnant signal models, the theoretical background is limited and there is no consensus on how to model and take into account this remnant in Monte Carlo simulations [11, 20, 28, 29]. In most cases, a remnant signal is obtained by passing zero-mean Gaussian white noise (GWN) through a filter.

The literature contains a variety of filter options, however, the most accepted one is the m^{th}-order remnant-filter proposed by Zaal [11] and later used in [10, 12]:

$$H_n^m(s,t) = \frac{K_n(t)}{(T_n s+1)^m}. \tag{3}$$

Additionally, the noise level P_n has to be set during Monte Carlo simulations to give a certain value for the remnant gain. The most adequate definition is provided by Van der El et al. [30], $P_n = \sigma_{u_n}^2/\sigma_u^2$, which compares the variance of $u(t)$ due to the remnant to the $u(t)$ signal's total variance σ_u^2.

B. Controlled Element Dynamics

The following second-order CE dynamics were taken into consideration by several authors [10–12], which serve as a general low-order approximation of typical vehicle dynamics [3]:

$$H_{CE}(s,t) = \frac{K_c(t)}{s(s+\omega_b(t))}. \tag{4}$$

The break frequency $\omega_b(t)$ and the control gain $K_c(t)$ can both change over time. The dynamics variation of the controlled element from single- to double-integrator dynamics (i.e., $1/s \leftrightarrow 1/s^2$) occurs at approximately $\omega_b(t)$. Furthermore, a sigmoid function is used in [10–12] to define the time variation of the operator equalization parameters in Eq. (2) and CE coefficients in Eq. (4):

$$p(t) = p_i + \frac{p_f - p_i}{1+e^{-G(t-t_M)}}, \tag{5}$$

where $p(t)$ is the time-varying parameter, p_i and p_f are the initial and final parameter values, G is the transition rate, and t_M is the time when states transition occurs.

C. Identification of Time-Varying Operator Behaviour

In HO system identification, there are two possible directions to estimate the H_{HO_e} dynamics. The non-parametric method [7, 18, 19, 31, 32], only provides direct estimations of frequency response, and the parametric approach assumes a HO model structure and requires estimation of its parameters.

The parametric methodology provides a more tangible understanding of the human controller than non-parametric strategies. Five research lines are found: batch-fitting methods (maximum likelihood estimation [11, 20], fitting Linear Parameter Variable (LPV) state space systems [21]) and recursive fitting methods (Kalman filter estimation [16, 22, 33], fitting recursive ARX models [10, 12, 25], identification of Artificial Neural Networks (ANNs) [23]).

In batch fitting strategies, the fitting is applied on the whole dataset at once, and typically, the operation limitation parameters (neuromuscular dynamics and operator time delay) are deemed constant while only the operator equalization parameters may vary. On the other hand, in recursive fitting methods, the operation equalization parameters can vary, but also, the NMS and HO time delay can be set constant or assumed time-varying depending on the type of estimator used. The ARX model structure [34] is employed in [10, 12, 25] to estimate time-varying HO behaviour, extending the work done in [6, 24]. ARX parameters are computed by the Recursive Least Squares (RLS) [35], which minimizes a weighted linear least squares cost function relating to the input signals. Additionally, the RLS algorithm can be tuned by evaluating the ARX structure for constant HO models [10] through an Ordinary Least Squares (OLS) algorithm [34].

ARX identification methods represent a simple, efficient option since they only require a linear optimization process. In addition, convergence is fast in recursive estimation, which guarantees an acceptable adaptation to HO

changes. However, permanent biases are found in estimated model parameters due to the mismatch in the remnant filter poles (mainly when the remnant order m equals 1 [10]), seeing that both denominators of the HO linear component and remnant filter are considered as equivalent in ARX structures. Hence, an analysis of other model structures is recommended in order to reduce the persistent bias found in ARX.

III. Box-Jenkins Model Identification

A. Box-Jenkins Model Family

Ljung [34] presents a series of transfer-function models from a general family of model structures, which can be employed to identify the discrete-time transfer function (TF) corresponding to an HO model. Figure 2 shows the ARX and BJ structures:

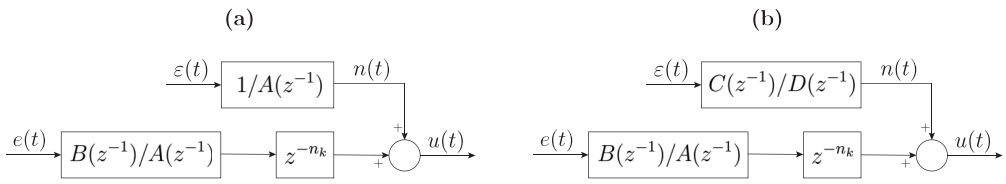

Fig. 2 Discrete-time model structures: (a) ARX, and (b) Box-Jenkins.

For the present case study, the Box-Jenkins structure represents the most suitable option since it enables more freedom for zero and pole placement in both the remnant filter and the linear HO model, while ARX assumes a coupling between these two components. This BJ family can be represented by the following expression:

$$u(t_k) = \frac{B(z^{-1})}{A(z^{-1})} z^{-n_k} e(t_k) + \frac{C(z^{-1})}{D(z^{-1})} \varepsilon(t_k), \quad \varepsilon(t_k) \sim N(0, \sigma_\varepsilon), \tag{6}$$

where $u(t)$ is the output signal associated with the control-output, $e(t)$ is the input signal corresponding to the tracking error, and $\varepsilon(t)$ is the Gaussian White Noise (GWN) with a standard deviation σ_ε. The variable z^{-1} acts as a discrete-time shift operator, while t_k represents the discretized time. The general model structure depends on a total of 4 polynomials $\{A, B, C, D\}$ with orders defined by the integers $\{n_a, n_b, n_c, n_d\}$:

$$A(z^{-1}) = 1 + a_1^d z^{-1} + \ldots + a_{n_a}^d z^{-n_a}, \qquad B(z^{-1}) = b_0^d + b_1^d z^{-1} + \ldots + b_{n_b}^d z^{-n_b}, \tag{7a}$$

$$C(z^{-1}) = 1 + c_1^d z^{-1} + \ldots + c_{n_c}^d z^{-n_c}, \qquad D(z^{-1}) = 1 + d_1^d z^{-1} + \ldots + d_{n_d}^d z^{-n_d}. \tag{7b}$$

In addition, the predictor of the control-output signal, $\hat{u}(t)$, would present the following general expression [34]:

$$\hat{u}(t_k \mid \theta) = \frac{B(z^{-1})D(z^{-1})}{C(z^{-1})A(z^{-1})} z^{-n_k} e(t_k) + \left[1 - \frac{D(z^{-1})}{C(z^{-1})}\right] u(t_k), \tag{8}$$

where θ is the adjustable parameters vector, i.e.,

$$\theta = \left[a_1^d, a_2^d, \ldots, a_{n_a}^d, b_0^d, b_1^d, \ldots, b_{n_b}^d, c_1^d, c_2^d, \ldots, c_{n_c}^d, d_1^d, d_2^d, \ldots, d_{n_d}^d\right]^T. \tag{9}$$

By adjusting this previous expression, the prediction error is found:

$$\epsilon(t_k, \theta) = u(t_k) - \hat{u}(t_k \mid \theta) = \frac{D(z^{-1})}{C(z^{-1})} \left[u(t_k) - \frac{B(z^{-1})}{A(z^{-1})} z^{-n_k} e(t_k)\right]. \tag{10}$$

B. HO Model Discretization and BJ Structure

Before setting batch-fitting or recursive estimations of BJ models, it is necessary to make sure those structures match the discretized HO and remnant filter models. Franklin et al. [36] present a variety of discretization techniques that can be applied to the continuous-time TF, and can follow a numerical integration, a Z-transform mapping strategy or try

to model the sampled system. The discrete equivalents via numerical integration are based on the numerical method [37] used to find a solution of the differential equation associated to the continuous-time TF to be discretized. The principal methods are: Forward-Euler, Backward-Euler, Tustin (with/without pre-warp). Then, the zero-pole matching equivalents are obtained by mapping the continuous-time TF's poles and zeros from the relationship between s- and z-planes. On the other hand, Tangirala [38] proposes model-sampling techniques, based on the Zero-Order Hold (ZOH) or First-Order Hold (FOH), that reconstruct the measured signal and enable a two-step continuous-to-discrete time TF conversion. In the HO model, a combination between ZOH and Backward Euler is applied.

1. HO Linear Component

In the case of $H_{HO_e}(s,t)$, Hess and Mnich [39] proved that the only feasible discretization method option is ZOH (there must be a single solution for HO coefficients), achieving a discrete model with $n_a = 2$ and $n_b = 1$:

$$H_{HO_{e,dis}}(z) = \frac{b_0^d + b_1^d z^{-1}}{1 + a_1^d z^{-1} + a_2^d z^{-2}} \cdot z^{-n_k}, \tag{11}$$

where an additional unit-sample delay z^{-1} is obtained due to the discretization method. The integer n_k represents the time delay in the discretized model. Thus, the estimated continuous-time TF of the HO model, $\hat{H}_{HO_e}(s,t_k)$, and the identified human operator coefficients are:

$$\hat{H}_{HO_e}(s,t_k) = \frac{b_0^c s + b_1^c}{s^2 + a_1^c s + a_2^c} \;\Rightarrow\; \left\{ \hat{K}_e = \frac{b_1^c}{a_2^c},\ \hat{T}_L = \frac{b_0^c}{b_1^c},\ \hat{K}_{\dot{e}} = \frac{b_0^c}{a_2^c},\ \hat{\omega}_{nm} = \sqrt{a_2^c},\ \hat{\zeta}_{nm} = \frac{a_1^c}{2\sqrt{a_2^c}}, \hat{\tau}_e = T_s(n_k - 1) \right\}. \tag{12}$$

2. Remnant Filter

By the Backward Euler discretization, $s = (1 - z^{-1})/T_s$, a proper discrete-time transfer function for the remnant filter is obtained:

$$H_n^m(s) = \frac{K_n}{(T_n s + 1)^m} \quad \rightarrow \quad H_{n,dis}^m(z) = \frac{K_n}{(T_n \frac{1-z^{-1}}{T_s} + 1)^m} = \frac{K_n \left(\frac{T_s}{T_n + T_s}\right)^m}{\left(1 - \frac{T_n}{T_n + T_s} z^{-1}\right)^m}. \tag{13}$$

Since the numerator of the estimated discrete-time remnant filter must be equal to 1 based on the BJ model structure, the resulting coefficient in the numerator of $H_{n,dis}^m(z)$ will be incorporated into the variance σ_ε^2, giving as a result a modified ε' with standard deviation $\sigma_{\varepsilon'}$:

$$\sigma_{\varepsilon'} = \frac{K_n T_s^m}{(T_n + T_s)^m} \sigma_\varepsilon. \tag{14}$$

Hence, the discrete-time model of the remnant noise signal would be the following one:

$$n(t_k) = \frac{1}{\left(1 - \frac{T_n}{T_n + T_s} z^{-1}\right)^m} \varepsilon'(t_k), \qquad \varepsilon'(t_k) \sim N(0, \sigma_{\varepsilon'}). \tag{15}$$

Depending on the m^{th}-order of the remnant filter, the number of discrete-time parameters, d_i^d, to be estimated changes, while the time constant T_n would need to be averaged from such parameters computed:

$$d_i^d = f_i(T_s, T_{n,d_i^d}) \quad \rightarrow \quad T_{n,d_i^d} \quad \rightarrow \quad \hat{T}_n = \frac{1}{m} \sum_{i=1}^{m} T_{n,d_i^d}. \tag{16}$$

3. Applied BJ Model Expression

Based on the considerations outlined in the previous sections, a BJ model structure $\{n_a = 2, n_b = 1, n_c = 0, n_d = m^*\}$ is chosen to address the identification problem:

$$u(t_k) = \frac{b_0^d + b_1^d z^{-1}}{1 + a_1^d z^{-1} + a_2^d z^{-2}} z^{-n_k^*} e(t_k) + \frac{1}{1 + d_1^d z^{-1} + \ldots + d_{m^*}^d z^{-m^*}} \varepsilon'(t_k), \quad \varepsilon'(t_k) \sim N(0, \frac{K_n T_s^{m^*}}{(T_n + T_s)^{m^*}} \sigma_\varepsilon), \tag{17}$$

where m^* and n_k^* are the remnant order and time delay considered in the estimation model, respectively. Thus, the $BJ(n_k^*, m^*)$ prediction error and adjustable parameter vector are given by the expressions:

$$\epsilon(t_k, \theta) = D(z^{-1}) \left[u(t_k) - \frac{B(z^{-1})}{A(z^{-1})} z^{-n_k^*} e(t_k) \right], \quad \theta = \left[a_1^d, a_2^d, b_0^d, b_1^d, d_1^d, \ldots, d_{m^*}^d \right]^T. \tag{18}$$

C. Prediction Error Method

The parameter estimation of each BJ discrete-time polynomial requires the optimization of a non-linear problem, hence, a Prediction Error Method (PEM) algorithm has to be evaluated as explained in [34, 38, 40, 41]. The PEM procedure consists of initial state estimation, definition of the optimization model, and variance estimation.

Other methods can also be applied to the discrete-time transfer function estimation problem, such as the Maximum Likelihood (ML) or the Refined Instrumental Variable (RIV) method [34, 40, 41]. The ML technique finds the full optimization model from a log-likelihood function, which is based on a Gaussian distribution of the noise ε. The RIV method is a pseudo-linear regression approach to ML estimation. Both methods could be in ARX and BJ structures, but they are only applicable to systems with Gaussian noise, thus, the PEM algorithm presents a more generalist alternative. The ML model is only used to find an expression to estimate the noise variance (see Eq. (30)).

1. The Algorithm

Prediction error methods are based on the idea of minimizing a cost function, V^N, that measures the level of prediction error, $\epsilon(t_k, \theta)$, to find a solution for $\hat{\theta}$ [34, 40]. Thus, from a batch of data Z^N,

$$Z^N = \left[u(t_1), e(t_1), u(t_2), e(t_2), \ldots, u(t_N), e(t_N) \right], \tag{19}$$

and the prediction error formula (see Eq. (18)), the cost function can be defined as follows:

$$V^N(\theta, Z^N) = \frac{1}{N} \sum_{i=1}^{i=N} l(\epsilon(t_i, \theta)). \tag{20}$$

In $V^N(\theta, Z^N)$, $l(\cdot)$ is a scalar-valued (typically positive) function. The quadratic norm is the most common in optimization problems:

$$l(\epsilon) = \frac{1}{2} \epsilon^2. \tag{21}$$

Therefore, the goal of a PEM algorithm is to find the vector of parameters $\hat{\theta}$ that minimizes the cost function:

$$\hat{\theta} = \arg\min_{\theta} V^N(\theta, Z^N) = \arg\min_{\theta} \left\{ \frac{1}{2N} \sum_{i=1}^{i=N} \left[D(z^{-1}) u(t_i) - \frac{B(z^{-1}) D(z^{-1})}{A(z^{-1})} z^{-n_k} e(t_i) \right]^2 \right\}. \tag{22}$$

To optimize such cost function, the partial differentiation of $V^N(\theta, Z^N)$ with respect to all the parameters is made:

$$\nabla_\theta \left[V^N(\theta, Z^N) \right] = \frac{1}{N} \sum_{i=1}^{i=N} \epsilon(t_i, \theta) \nabla_\theta \left[\epsilon(t_i, \theta) \right]. \tag{23}$$

Consequently, the optimization model is defined by the following set of equations, when set to zero:

$$\frac{\partial V^N}{\partial a_{j=1,2}^d} = \frac{1}{N} \sum_{i=i_0}^{i=N} \left[D(z^{-1}) u(t_i) - \frac{B(z^{-1}) D(z^{-1})}{A(z^{-1})} z^{-n_k} e(t_i) \right] \times \frac{B(z^{-1}) D(z^{-1})}{A^2(z^{-1})} z^{-j-n_k} e(t_i) = 0, \tag{24a}$$

$$\frac{\partial V^N}{\partial b_{j=0,1}^d} = \frac{1}{N} \sum_{i=i_0}^{i=N} \left[D(z^{-1}) u(t_i) - \frac{B(z^{-1}) D(z^{-1})}{A(z^{-1})} z^{-n_k} e(t_i) \right] \times \frac{D(z^{-1})}{A(z^{-1})} z^{-j-n_k} e(t_i) = 0, \tag{24b}$$

$$\frac{\partial V^N}{\partial d_{j=1,2,\ldots,m^*}^d} = \frac{1}{N} \sum_{i=i_0}^{i=N} \left[D(z^{-1}) u(t_i) - \frac{B(z^{-1}) D(z^{-1})}{A(z^{-1})} z^{-n_k} e(t_i) \right] \times \left[z^{-j} u(t_i) - \frac{B(z^{-1}) z^{-j-n_k}}{A(z^{-1})} e(t_i) \right] = 0. \tag{24c}$$

At this point, the PEM and ML optimization problems are identical [41], since Eq. (24) match the ones obtained in an ML scenario. In order to find the optimal solution to this minimization problem, BJ models require non-linear optimization methods based on gradient-based schemes [42], or more innovative techniques, such as the combination of different line search algorithms at each iteration [43], which is used in this case.

When calculating each component of Eq. (24), there may be some difficulties in evaluating fractions of polynomials. To solve this issue, Maclaurin series [43, 44] can be used to convert a fraction into a pure polynomial:

$$\frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots \tag{25}$$

2. Initial States Estimation

In BJ estimation, the index i_0 in Eq. (24) should be always equal to 1 due to the higher relevance of the initial states in the optimization process. Otherwise, if a first-order series is assumed, the minimum index should be:

$$i_0 = 3n_a + n_b + n_d + n_k + 1 = m^* + n_k^* + 8. \tag{26}$$

Therefore, an initial model needs to be estimated beforehand to find the pairs $\{u(t_{k<1}), e(t_{k<1})\}$, i.e.,

$$Z_i^N = [\dots, u(t_{-1}), e(t_{-1}), u(t_0), e(t_0)]\,, \tag{27}$$

based on the initial conditions θ_i (i.e., $A_i(z^{-1})$, $B_i(z^{-1})$, $C_i(z^{-1})$, $D_i(z^{-1})$) [43]:

$$\hat{Z}_i^N = \arg\min_{Z_i^N} V_i^N(\theta_i, Z_i^N) = \arg\min_{Z_i^N} \left\{ \frac{1}{2N} \sum_{i<1} \left[D_i(z^{-1})u(t_i) - \frac{B_i(z^{-1})D_i(z^{-1})}{A_i(z^{-1})} z^{-n_k^*} e(t_i) \right]^2 \right\}. \tag{28}$$

3. Variance Estimation

Once, the optimization process is converged and a parameter vector solution is found, i.e., a partial differentiation of the log-likelihood function for N observations shown by Young [41],

$$\mathscr{L}(\theta, \sigma^2_{\varepsilon'}, u(t), e(t)) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln(\sigma^2_{\varepsilon'}) - \frac{1}{2\sigma^2_{\varepsilon'}} \sum_{i=1}^{i=N} \left[D(z^{-1})u(t_i) - \frac{B(z^{-1})D(z^{-1})}{A(z^{-1})} z^{-n_k} e(t_i) \right]^2 \tag{29}$$

is computed to find an estimation of the noise variance $\hat{\sigma}^2_{\varepsilon'}$ based on the predicted parameter vector:

$$\frac{\partial \mathscr{L}}{\partial \sigma^2_{\varepsilon'}} = 0 \quad \rightarrow \quad \hat{\sigma}^2_{\varepsilon'} = \frac{1}{N} \sum_{i=1}^{i=N} \left[\hat{D}(z^{-1})u(t_i) - \frac{\hat{B}(z^{-1})\hat{D}(z^{-1})}{\hat{A}(z^{-1})} z^{-n_k^*} e(t_i) \right]^2. \tag{30}$$

D. Recursive Prediction Error Minimization

The Recursive Prediction Error Minimization (RPEM) is one of the most extended online estimation techniques, which enables a recursive implementation of the PEM algorithm [34, 40]. Other methods can be applied to the online BJ structure estimation problem, such as the Real-Time Recursive Refined Instrumental Variable (RRIV) or the Extended Kalman Filter (EKF) [40]. The RRIV method is probably the most similar to the RPEM algorithm, although it presents some differences in terms of the covariance matrix definition (two sub-matrices with null off-diagonal blocks are used in RRIV), the robustness of the algorithm (the Instrumental Variable modifications ensure a stable estimation process), or the steps required (while RPEM is fully recursive, RRIV needs to iterate at each k^{th} period).

A Recursive PEM algorithm [40] is built, by analogy with the RLS method and fulfilling the theoretical requirements to achieve convergence. The RPEM algorithm is an online estimation method that consists of 3 steps, in which the parameter vector $\theta(t_k)$ is adjusted online by means of the gain vector $g(t_k)$ and the prediction error of $u(t_k)$. The gain vector is also tuned based on a forgetting factor λ, the negative gradient of the prediction error $\pi(t_k)$, and the scaled covariance matrix $P(t_k)$, which accounts for the certainty in the estimation conducted of each model parameter. Eq. (31) [34, 40] present the formulas used in this iterative process:

$$\hat{\theta}(t_k) = \hat{\theta}(t_{k-1}) + P(t_k)\pi(t_k)\epsilon(t_k), \tag{31a}$$

$$g(t_k) = \frac{P(t_{k-1})\pi(t_k)}{\lambda + \pi^T(t_k)P(t_{k-1})\pi(t_k)}, \tag{31b}$$

$$P(t_k) = \frac{1}{\lambda}\left[P(t_{k-1}) - g(t_k)\pi^T(t_k)P(t_{k-1})\right]. \tag{31c}$$

Equation Eq. (31a) can be evaluated in two ways due to the relationship $g(t_k) = P(t_k)\pi(t_k)$, although the first option is usually preferred since it is more computationally efficient [40]. Each element of the forgetting matrix must belong to the interval $I = [0, 1]$, where a λ near a null value provides a negligible memory horizon, and values close to 1 increase this horizon significantly. Hence, for a forgetting factor λ, a total number of $N_e = 1/(1-\lambda)$ samples are considered in the RLS algorithm for a time horizon of $T_e = T_s/(1-\lambda)$.

The definition of vector $\pi(t_k)$ is based on the minimization of the instantaneous part of the cost function in Eq. (20), in which the one-step-ahead prediction error can be approximated by the employment of the model parameters estimation at the previous $(k-1)^{th}$ instant:

$$V^k(\theta(t_k)) = \frac{1}{2}\left[\epsilon^2(t_k, \theta(t_k))\right] \approx V^k(\hat{\theta}(t_{k-1})) = \frac{1}{2}\left[\epsilon^2(t_k, \hat{\theta}(t_{k-1}))\right]. \tag{32}$$

As a result, the gain $\pi(t_k)$ is computed as the negative gradient of the prediction error [40]:

$$\pi(t_k) = -\frac{\partial\epsilon(t_k, \hat{\theta}(t_{k-1}))}{\partial\hat{\theta}(t_{k-1}))}. \tag{33}$$

From Eq. (18), the prediction error derivatives can be defined as follows:

$$\frac{\partial\epsilon(t_k)}{\partial a^d_{i=1,2}} = \frac{D(z^{-1})}{A(z^{-1})}\frac{B(z^{-1})}{A(z^{-1})}e(t_{k-i-n_k}) = \frac{D(z^{-1})}{A(z^{-1})}y(t_{k-i-n_k}) = y_{f_1}(t_{k-i-n_k}), \tag{34a}$$

$$\frac{\partial\epsilon(t_k)}{\partial b^d_{i=0,1}} = -\frac{D(z^{-1})}{A(z^{-1})}e(t_{k-i-n_k}) = -e_{f_1}(t_{k-i-n_k}), \tag{34b}$$

$$\frac{\partial\epsilon(t_k)}{\partial d^d_{i=1,2,\ldots,m^*}} = -D(z^{-1})u(t_{k-i}) - \frac{D(z^{-1})B(z^{-1})}{A(z^{-1})}e(t_{k-i-n_k}) = -\epsilon(t_{k-i}). \tag{34c}$$

Where the subscript f_1 denotes that the variable is filtered by the transfer functions:

$$f_1 = \frac{D(z^{-1})}{A(z^{-1})}. \tag{35}$$

Therefore, the negative gradient $\pi(t_k)$ presents the following expression:

$$\pi(t_k) = -\frac{\partial\epsilon}{\partial\hat{\theta}} = \left[-\hat{y}_{\hat{f}_1}(t_{k-1-n_k}), -\hat{y}_{\hat{f}_1}(t_{k-2-n_k}), e_{\hat{f}_1}(t_{k-n_k}), e_{\hat{f}_1}(t_{k-1-n_k}), \hat{\epsilon}(t_{k-1}), \ldots, \hat{\epsilon}(t_{k-n_d})\right]^T. \tag{36}$$

The prefilter $\hat{f_1}$ and variables $\hat{y}(t_k)$ are calculated from the latest estimated polynomials, respectively:

$$\hat{f_1} = \frac{\hat{D}(z^{-1})}{\hat{F}(z^{-1})}, \quad \hat{y}(t_k) = \frac{\hat{B}(z^{-1})}{\hat{F}(z^{-1})}e(t_{k-n_k}). \tag{37}$$

Thus, the classical RLS algorithm is adapted to non-linear cases by means of converting $\pi(t_k)$ into a vector composed of linear variables, as shown in Eq. (36). A forgetting factor $\lambda = 0.99609$ is selected for the implementation of the recursive BJ method, based on previous results from Van Grootheest et al. [10]. In addition, the initial covariance matrix can be defined as:

$$P_i = diag\left(0.1, 0.1, 0.1, 0.1 \times m^*\right). \tag{38}$$

E. Quality-of-Fit Metrics

In order to verify the quality of the estimation performed, several metrics need to be used. These metrics have to be relevant in terms of addressing how well the predicted model represents the dynamics of the real one. To achieve such a goal, the prediction error in estimated model parameters has to be analyzed, but the prediction capabilities of such a model must be also studied.

Hence, two different quality-of-fit metrics are employed: the Variance Accounted For (VAF) [6], and the relative bias, $B_r(\vartheta)$. Both have been employed in previous works to verify the ARX results [10, 12], thus, these ones should be also applicable to the BJ results. The VAF metric [45],

$$VAF^{n_k^*,m^*} = max\left\{0, \left(1 - \frac{\sum_{k=1}^{N} \mid u(t_k) - \hat{u}^{n_k^*,m^*}(t_k) \mid^2}{\sum_{k=1}^{N} \mid u(t_k) \mid^2}\right) \cdot 100\%\right\}, \tag{39}$$

evaluates the correctness of a model, by comparing the real output, $u(t_k)$, with the estimated output of the model, $\hat{u}^{n_k^*,m^*}(t_k)$. The VAF of two signals that are the same is 100%, while it will be lower if they differ. For a certain combination of time delay and remnant filter order in the structure, the VAF difference between ARX and BJ models can be used to compare the accuracy of both methods and analyze the effect of the BJ model remnant order on estimation results:

$$\Delta VAF^{n_k^*,m^*} = VAF_{BJ}^{n_k^*,m^*} - VAF_{ARX}^{n_k^*}. \tag{40}$$

On the other hand, the relative bias can be used to verify the accuracy of parameter estimation:

$$B_r(\vartheta) = \left(\frac{\hat{\vartheta} - \vartheta^0}{\vartheta^0}\right) \cdot 100\%, \tag{41}$$

where ϑ can be a discrete-time parameter from ARX or BJ models, or an HO coefficient.

IV. Method

A. Forcing Function

To accomplish adequate simulations, the forcing function needs to be defined properly. As proposed by Zaal [11], a summation of N_t sinusoids with different amplitudes A_k, frequencies ω_k and phases ϕ_k,

$$f_t(t) = \sum_{k=1}^{N_t} A_k \cdot \sin\left(\omega_k \cdot (t - t_0) + \phi_k\right), \tag{42}$$

is an effective option to excite the closed-loop system shown in Figure 1. To avoid spectral leakage, the frequencies ω_k are chosen as multiples of the base frequency $\omega_{k,0} = 2\pi/T_m$, where T_m is the total simulation time. Thus, the integer n_t is defined as $n_t = \omega_k/\omega_{k,0}$. In addition, a transient interval time is introduced in the simulation, t_0, in order that control loop variables are stabilized, consequently, the sinusoids are initialized at $t = t_0$. Table 1 shows the coefficients of each sinusoid employed in simulations, while Figure 3 depicts its auto-Power Spectral Density (PSD) function, $S_{f_t f_t}(j\omega)$, and its peaks located at ω_k.

Table 1 Target function parameters.

$k, -$	$n_t, -$	$\omega_k, rad/s$	A_k, deg	ϕ_k, rad
1	3	0.230	1.186	-0.753
2	5	0.384	1.121	1.564
3	8	0.614	0.991	0.588
4	13	0.997	0.756	-0.546
5	22	1.687	0.447	0.674
6	34	2.608	0.245	-1.724
7	53	4.065	0.123	-1.963
8	86	6.596	0.061	-2.189
9	139	10.661	0.036	0.875
10	229	17.564	0.025	0.604

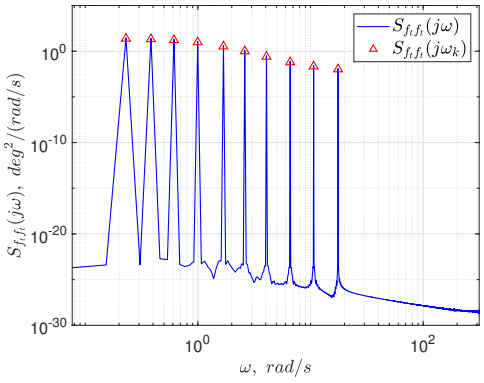

Fig. 3 Auto-PSD of forcing function $f_t(t)$.

B. Remnant Noise

Then, a remnant realization is generated for every single simulation run by feeding the zero-mean GWN with unit variance through a filter, i.e., $\sigma_\varepsilon = 1$. Different m^{th}-order filters, $m \in \{1, 2, 3, 4\}$, and a remnant time constant, $T_n = 0.06$, can be simulated to address their influence on results (see [10, 11, 26]). Additionally, it is possible to introduce multiple noise levels $P_n \in [0.0, 1.0)$ in the system by modifying the remnant gain K_n. $P_n \in \{0.0, 0.1, 0.2, 0.3\}$ are considered in batch-fitting estimation, while $P_n \in \{0.01, 0.10, 0.20\}$ are used in recursive analysis. Consequently, a certain value for the K_n parameter has to be selected to obtain a requested noise level at each simulation, for which the stochastic theory presented by Ljung [34] can be employed, so that a suitable formula is developed. From the noise level definition, $P_n = \sigma^2_{u_n}/\sigma^2_u$, based on the forcing function expression and HO and CE models:

$$K_n = \sqrt{\frac{P_n}{(1-P_n)\cdot T_s} \frac{\frac{\pi}{2}\sum_{k=1}^{N_t} A_k^2 \left| \frac{H_{HO_e}(j\omega_k)}{1+H_{HO_e}(j\omega_k)H_{CE}(j\omega_k)} \right|^2}{\int_0^{\pi/T_s} \frac{\mathrm{d}\omega}{\left|(T_n(j\omega)+1)^m (1+H_{HO_e}(j\omega)H_{CE}(j\omega))\right|^2}}}. \tag{43}$$

C. Simulation Conditions

To create a time-varying simulation framework, two different states, s_1 and s_2, are defined on the basis of the CE dynamics variation and its effect on the human operator. Zaal [11] employs two sets of parameters for the CE dynamics and the assumed HO dynamics resulting from adaptation. In the state s_1, the HO×CE dynamics are given by a crossover frequency $\omega_c = 1.5\ rad/s$ and a phase margin $\phi_m = 77.0°$, while these two parameters are converted into $\omega_c = 2.8\ rad/s$ and $\phi_m = 22.7°$ in state s_2. When a ZOH discretization is applied, different discrete-time coefficients are obtained for each set. Furthermore, the corresponding remnant gain K_n is defined for each state based on Eq. (43). These parameters are recorded in Table 2:

Table 2 CE, HO and ZOH-discretization parameters for states s_1 and s_2.

State	CE		HO							ZOH			
	K_c	ω_b	K_e	T_L	$K_{\dot{e}}$	τ_e	ω_{nm}	ζ_{nm}	T_n	$a_1^{d,0}$	$a_2^{d,0}$	$b_0^{d,0}$	$b_1^{d,0}$
	–	rad/s	–	s	s	s	rad/s	–	s	–	–	–	–
s_1	90	6.0	0.09	0.40	0.036	0.28	11.25	0.35	0.06	-1.9121	0.9243	0.0443	-0.0432
s_2	30	0.2	0.07	1.20	0.084	0.28	11.25	0.35	0.06	-1.9121	0.9243	0.1024	-0.1016

Two types of simulation conditions are considered in this article based on the scenarios proposed by Zaal [11]: a constant set of parameters (C1-C2), or a time-varying scenario (C3-C6). In the first case, the state s_1 (C1) or s_2 (C2) is implemented during the entire simulation trial. The second case is defined by a change between states (performed by a sigmoid function in Eq. (5)) $s_1 \rightarrow s_2$, in C3-C4 case, or $s_2 \rightarrow s_1$, in C5-C6. Two types of transitions are considered based on the sigmoid parameter G (see Eq. (5)), i.e., a slow states change $G = 0.5\ s^{-1}$ in C3 and C5, and a fast transition $G = 100\ s^{-1}$ in C4 and C6. A transition from single- to double-integrator dynamics, together with an aggressive CE transition, i.e. $G = 100\ s^{-1}$, is applied in order to analyze the most challenging scenario, employing the simulation condition 'C4' in consequence. The HO time-delay and NMS parameters are assumed to remain constant during the simulation [10, 11]. Figure 4 shows the effect of each simulation condition on K_e as an example:

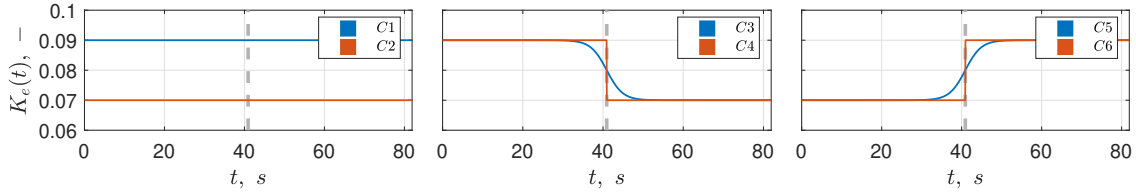

Fig. 4 HO gain for simulation conditions C1-C6 under simulation time $T_m = 81.92s$ and time of maximum rate of change $t_M = 40.96s$.

V. Simulation Results

The obtained Monte Carlo simulation results are addressed in this section. The ARX and BJ estimators' performance is evaluated in a batch-fitting scenario under simulation condition C1 using OLS and PEM algorithms, respectively, so that the estimation accuracy and influence of model parameters, such as the remnant filter or time delay, and the noise level can be analyzed. Two cases of model remnant order m^* are studied: an order that matches the simulation remnant order, i.e. $m^* = m^0 \in \{1, 2, 3, 4\}$, or a fixed first-order, i.e. $m^* = 1$. An optimal PEM algorithm with a BJ($m^* = 1$) structure is selected due to its efficiency and accuracy. Once the BJ estimator is studied, the RLS and RPEM algorithms are evaluated in a time-varying scenario (i.e. simulation condition C4) for ARX and BJ models, respectively. A total of $M = 100$ replications are evaluated in each case, which are shown in Box and Whiskers plots in batch-fitting scenario, or averaged in recursive estimation. A measurement time $T_m = 81.92\ s$ and transient time $t_0 = 10.00\ s$ are employed. Estimation processes are only conducted during the measurement time, after tracking error and control output signals are stabilized. The sample time is $T_s = 0.01\ s$.

A. Batch-Fitting Estimation

1. ARX and BJ Model with Remnant Order $m^* = m^0$

Figure 5 shows that a large relative bias is obtained for $m^0 = 1$ due to poles mismatch in the ARX remnant filter structure, since it is modeled with the same discrete-time denominator used in the linear HO transfer function. Thus, the ARX method sacrifices estimation accuracy to explain the remnant dynamics more precisely, by means of a filter structure unable to model a first-order system. However, when remnant filters with order $m^0 \geq 2$ are simulated, the ARX bias is diminished to acceptable values because the number of model remnant filter poles is never higher than the number of poles in $H_n^m(s,t)$, although the estimation error in b_0^d and b_1^d is still around 50% in most cases.

Changes in model time delay (x-axis) produce a linear trend in computed relative bias since the ARX estimator tries to create or reduce lag. Hence, first-order integrator dynamics are achieved sooner for time delays $n_k^* < n_k^0 = 29$ by placing the NMS poles at a lower frequency, while additional lead is generated in cases when $n_k^* > n_k^0 = 29$.

Increments in noise level give as a result a greater bias in general, which affects the stability of the ARX estimator severely. Since the model filter structure is wrong and imposes unrealistic poles, higher remnant levels will lead to an increase in estimation error. On the other hand, a null bias can be obtained in a remnant-free scenario if the ideal time delay is employed (this fact proves a discrete-time estimator is implemented properly).

As depicted in Figure 6, the BJ estimator is capable of improving on the ARX results considerably. The BJ model is initialized with coefficients close to the true simulated HO parameters to ensure convergence, although the PEM algorithm's flexibility with inaccurate initial conditions has been proven. The important bias found in first-order remnant order simulations is now highly reduced, while estimations with similar accuracy are achieved for different remnant orders.

Since BJ structures allow different pole allocations in $H_{HO_e}(s,t)$ and $H_n^m(s,t)$, but also the same number of poles in simulated and model remnant filters, BJ estimators are quite adaptive to create discrete-time structures that fit each HO model adequately. For remnant orders $m^0 \geq 2$, ARX bias results present comparable values to BJ ones, nevertheless, BJ estimators still provide a response with around 50% less estimation error.

Similar to the ARX case, the BJ estimator creates lag or lead depending on the assumed model time delay, hence, a linear trend is also found. Nonetheless, relative bias values are close to zero for increasing values in noise level, which ensures an ideal HO identification. Only small changes in b_0^d and b_1^d are found for $m^0 = 4$. In contrast to ARX models, relative bias results oscillate around the value obtained for the remnant-free scenario in BJ estimators. Therefore, BJ structures represent a more consistent, robust alternative in batch-fitting cases, since the simulation remnant order and noise level do not affect the accuracy of estimation.

2. ARX and BJ Model with Remnant Order $m^* = 1$

Although the PEM algorithm for BJ model structures has been proven to be an adequate, precise estimation method in the discrete-time domain, more information about the effect of the model remnant order is required to develop a suitable, efficient estimator. Consequently, the VAF metric is evaluated in Figure 7 for multiple BJ model time delays and remnant orders in a $m^0 = 3$ scenario, while it is also compared to the VAF results for ARX models in Figure 8. The BJ estimator presents a homogeneous average VAF value for all model remnant orders, and differences are only found for different time delays and noise levels. However, in scenarios with $m^0 = 1$, it has been observed a poorer performance when high model remnant orders are utilized. No significant differences are found between BJ and ARX

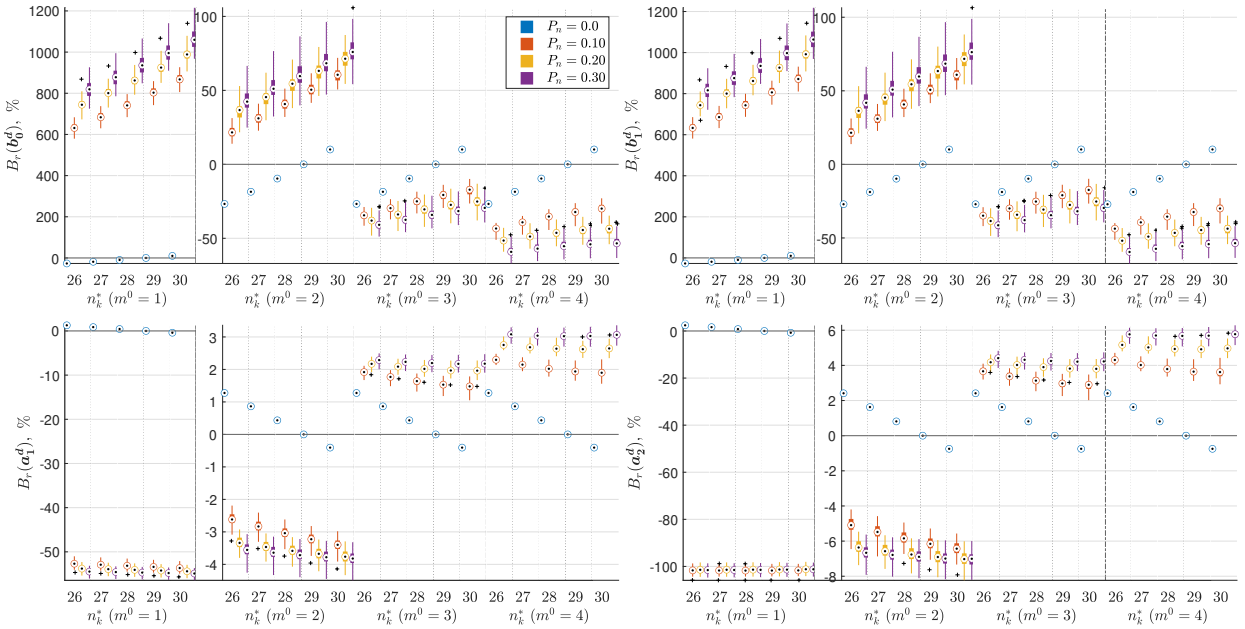

Fig. 5 Box and Whisker plots for relative bias of discrete-time parameters in ARX model.

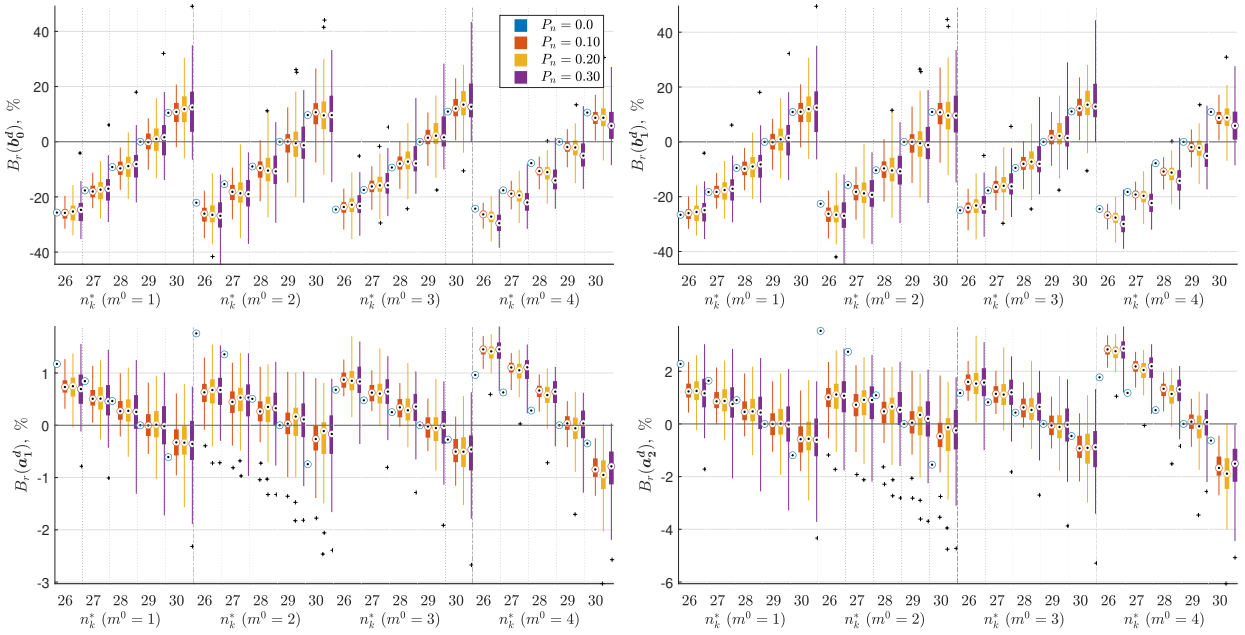

Fig. 6 Box and Whisker plots for relative bias of discrete-time parameters in BJ model with $m^* = m^0$.

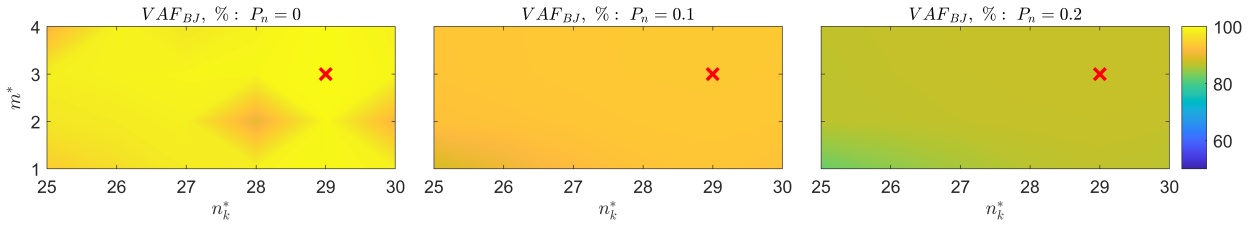

Fig. 7 VAF **results for BJ model with multiple model time delay** n_k^* **and remnant order** m^***. Simulation time delay** $n_k^0 = 29$ **and remnant order** $m^0 = 3$ **(red cross).**

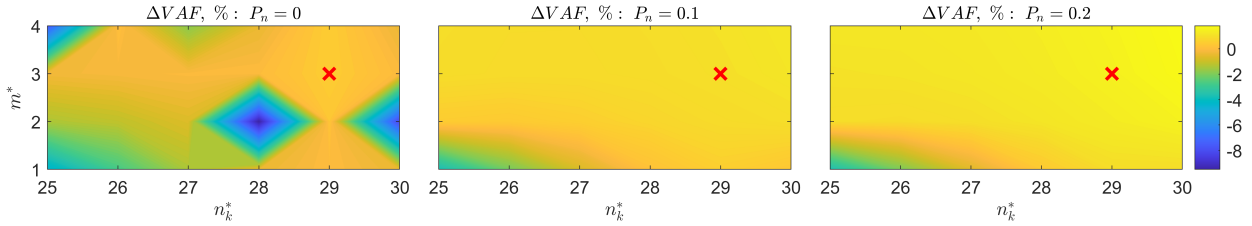

Fig. 8 Difference in VAF **results between BJ and ARX models with multiple model time delay** n_k^* **and remnant order** m^***. Simulation time delay** $n_k^0 = 29$ **and remnant order** $m^0 = 3$ **(red cross).**

for cases with a time delay close to $n_k^* = 29$. Hence, the use of BJ models with $m^* = 1$ is recommended to reduce the degrees of freedom in these structures without a cost in estimation accuracy.

Such a hypothesis about the use of a constant model remnant order $m^* = 1$ can be also validated by the relative bias analysis, shown in Figure 9. Equivalent results are found when compared to Figure 6, and the bias in $m^0 = 4$ is even slightly diminished. BJ structures with $m^* = 1$ present a discretized remnant filter directly obtained from the Backward-Euler method, while the number of parameters in $H_n^m(s,t)$ to be estimated differs from the number of d^d coefficients, which makes the estimation more robust when $m^* = 1$.

Bode plots of ARX and BJ models for multiple noise levels and simulation remnant orders, when the ideal time delay is selected, are shown in Figure 10. The real H_{HO_e} dynamics are represented by the ZOH discretization. There is a clear difference between both model structures, which can be explained by the relative bias results presented previously. The zero is located at lower frequencies for higher noise levels in ARX results, obtaining a higher magnitude peak. The ARX structure fails to capture the NMS poles in a first-order remnant filter, while these ones are moved to lower frequencies when being estimated for higher remnant orders. ARX models attempt to explain remnant dynamics by generating lag when the remnant order is increased, giving as a result a wrong estimation. This issue is not present in the BJ case owing to the flexibility of its model structure, accomplishing almost 100% accurate estimations in all scenarios.

B. Recursive Estimation

To estimate ARX and BJ models recursively, RLS and RPEM algorithms with a forgetting factor $\lambda = 0.996609$ are employed, respectively. The covariance matrix is initialized as shown in Eq. (38) in both methods. Two batch-fitting estimations are performed for time intervals $[0, 30.72]$ s and $[51.20, 81.92]$ s to capture s_1 and s_2 dynamics, based on [10]. The first OLS batch-fitting outcome is utilized to initialize the RLS algorithm (i.e. OLS→RLS). Also, OLS estimations are used as initial conditions for the PEM algorithm, afterwards, these converged batch-fitting coefficients are employed in the RPEM initialization (i.e. OLS→PEM→RPEM). Through this strategy, the recursive BJ algorithm is independent of initial conditions set by the user.

Figure 11 depicts the estimation of the discrete-time parameters from RLS and RPEM for multiple remnant orders and noise levels. In the $m^0 = 1$ case, ARX models lead to significant relative bias while BJ provides a much more precise estimation with almost null bias after full convergence, similar to what is shown previously in Figures 5 and 9. For remnant orders $m^0 \geq 2$, ARX improves its results, but it still shows a considerable bias with respect to BJ outcome. Therefore, batch-fitting results observed in previous figures also represent the real functioning of RLS and RPEM algorithms in recursive estimation.

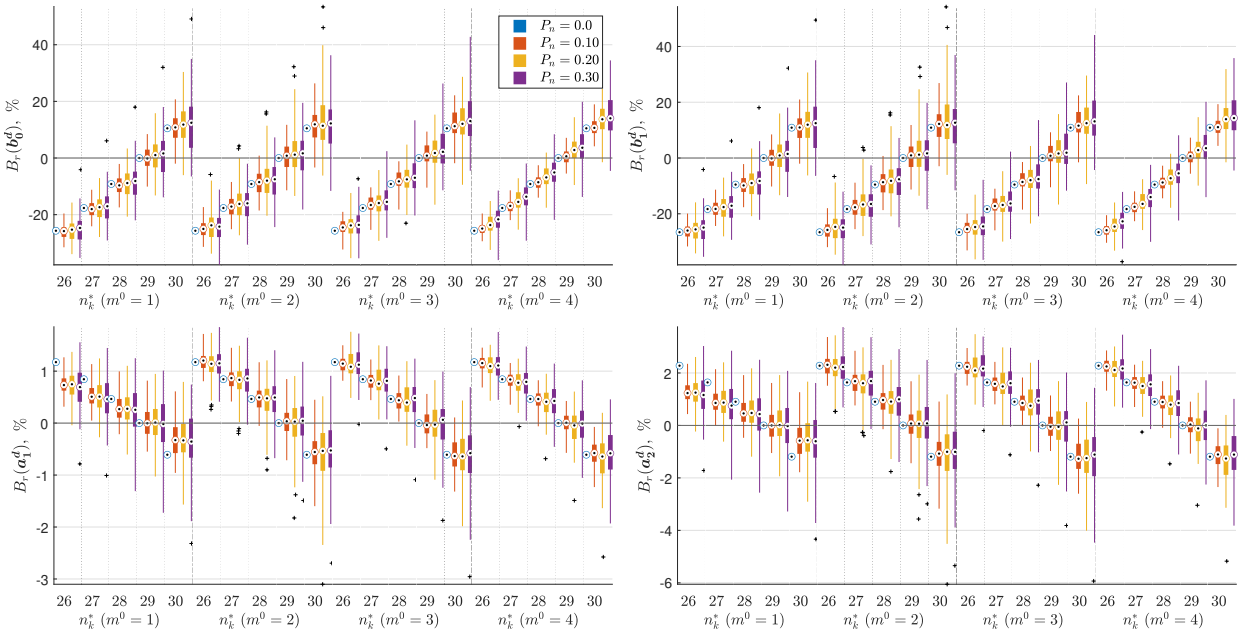

Fig. 9 Box and Whisker plots for relative bias of discrete-time parameters in BJ model with $m^* = 1$.

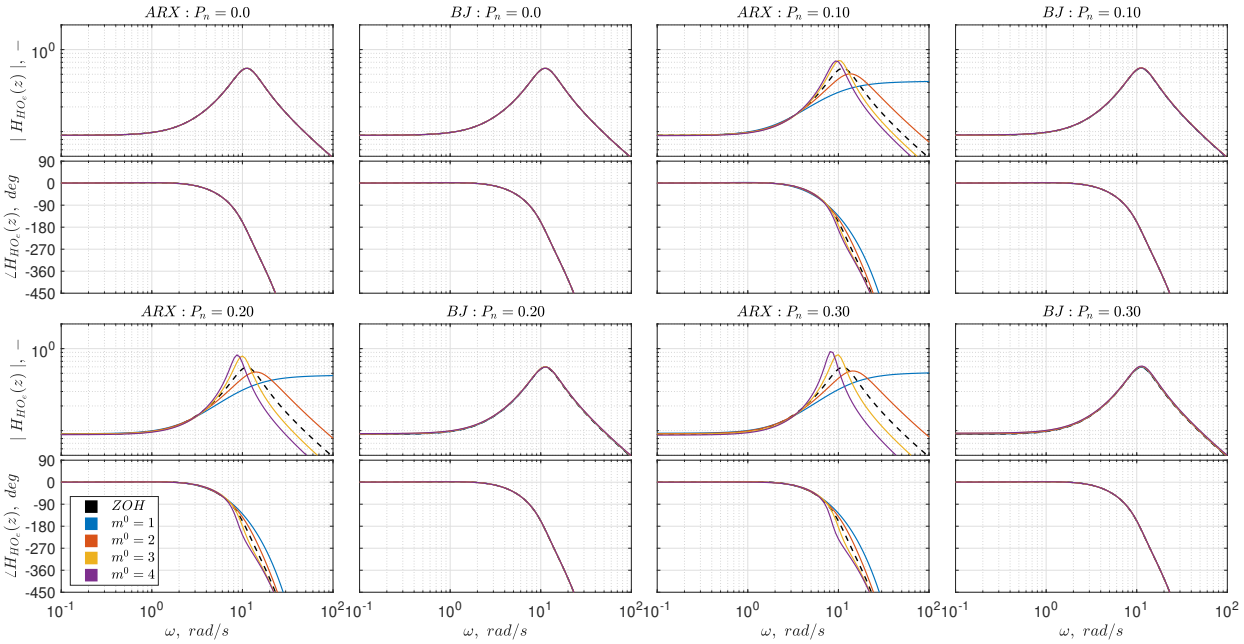

Fig. 10 Bode plots for ARX and BJ models with model time-delay $n_k^* = n_k^0 = 29$ and remnant filter order $m^* = 1$. Reference ZOH discretization (black, dashed line).

Nonetheless, the recursive BJ estimator presents more difficulties in achieving convergence for high noise levels. While the RLS algorithm adapts quite fast to changes in simulation parameters, and provides a stable estimation once those parameters are fixed, the BJ structure needs 500% more time to process, which may turn into divergence problems due to excessive oscillations in some cases. The RPEM algorithm denotes a slower adaptation capacity, making it more difficult for the recursive estimation to converge to each sub-batch-fitting result.

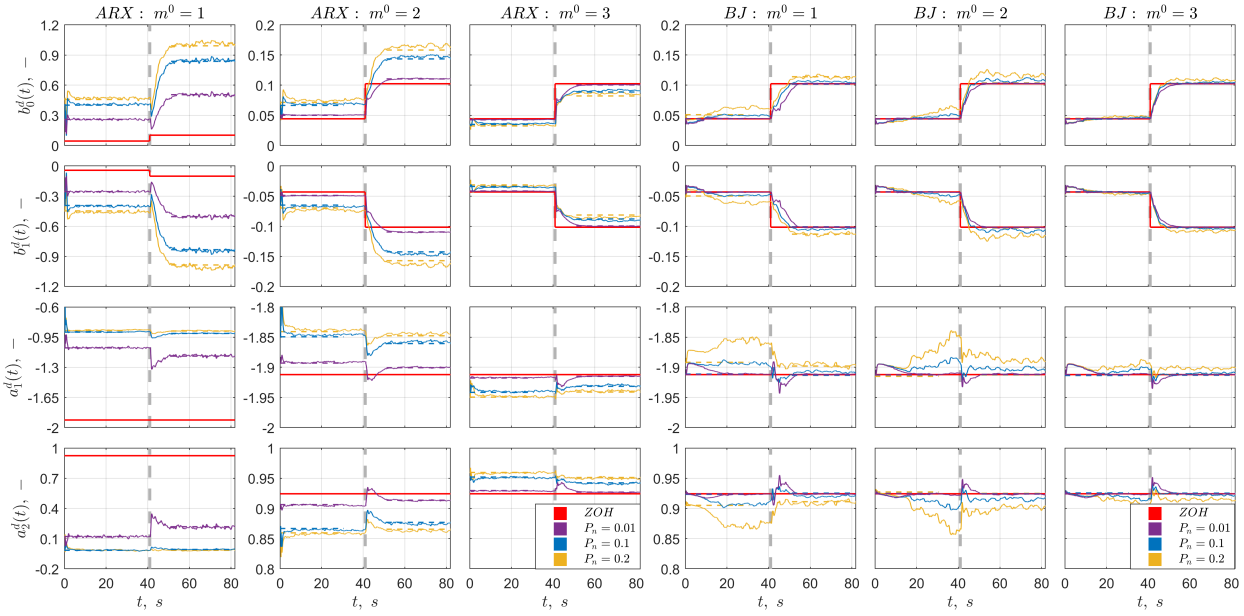

Fig. 11 Simulation results of recursive ARX($n_k^* = 29$) and BJ($n_k^* = 29, m^* = 1$) algorithms in discrete-time parameters. ARX estimations: OLS (dashed line), RLS (continuous line). BJ estimations: PEM (dashed line), RPEM (continuous line).

Figure 12 shows the identified HO coefficients by ARX and BJ model structures. The RLS algorithm presents considerable errors in the NMS coefficients for remnant order $m^0 = 1$ simulations, which are mainly produced by highly biased discrete-time estimations. When the discrete-time coefficients are obtained, the discrete-time state-space system form is calculated for the transfer function $B(z^{-1})/A(z^{-1})$. Then, Gajic's procedure [46] is used reversely to get the continuous-time state-space system by means of the logarithm of its extended matrix. To obtain a successful conversion of the state-space system, its associated extended matrix must be invertible and with no negative real eigenvalues [46]. Hence, the following discrete-time parameter constraints must be respected to ensure a proper state-space conversion: $\{a_1^d < 0, a_2^d > 0\}$.

When simulations with $m^0 \geq 2$ are conducted, the bias in ARX estimations is reduced by almost 10 times, although it is still persistent in NMS coefficients. However, ARX offers more stable, precise results than BJ in K_e, T_L and $K_{\dot{e}}$, while the BJ model provides a more adequate outcome in NMS parameters. Particularly, clear oscillations are visible in the T_L estimates obtained from BJ structures. Thus, ARX is able to adjust its discrete-time estimations to focus on gain and zero estimation mainly, at the cost of a permanent NMS bias. On the other hand, BJ gives the same priority to all coefficients, based on its non-linear optimization process. In addition, an additional error is produced when $m^0 = 2$ in BJ estimations since coupling can occur in poles of $H_{HO_e}(s,t)$ and $H_n^m(s,t)$.

VI. Experimental Results

Experimental results obtained in an experiment in the SIMONA Research Simulator at Delft University of Technology by Van Grootheest et al. [10] are analyzed in this section, following a similar estimation strategy explained in Section V.B. Three subjects conducted the single-axis compensatory tracking task detailed in Figure 1. The run-in time is $t_0 = 8.08\ s$, while the measurement time $T_m = 81.92\ s$. A total of 7 runs were performed, from which the last 5 ones are evaluated. The sample time is $T_s = 0.01\ s$. Only the results for (simulation) condition C4 are studied below, in which a fast transition from state s_1 to s_2 is performed, see Section IV.C.

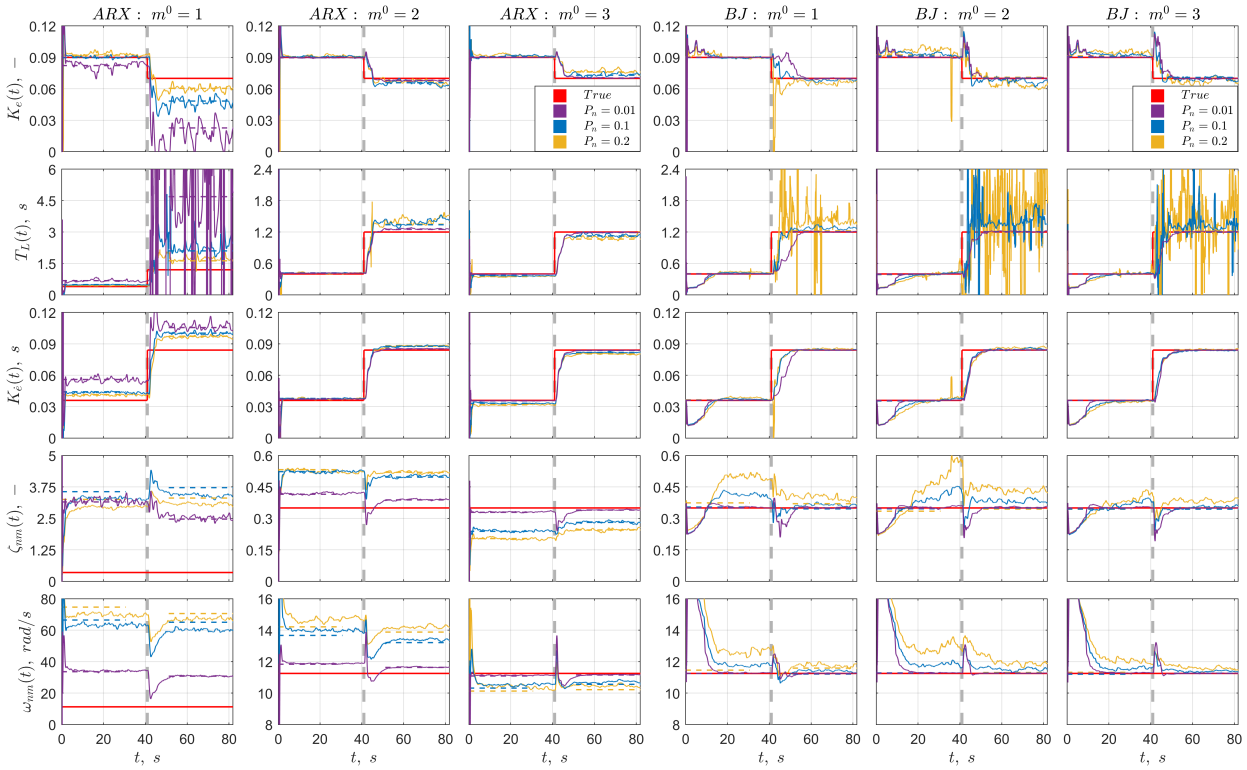

Fig. 12 Simulation results of recursive ARX($n_k^* = 29$) and BJ($n_k^* = 29, m^* = 1$) algorithms in HO coefficients. ARX estimations: OLS (dashed line), RLS (continuous line). BJ estimations: PEM (dashed line), RPEM (continuous line).

A remnant model with order $m^* = 1$ is employed, while the model time delay n_k^* and remnant time constant T_n are computed a priori by means of the average between the best two sub-batch-fitting estimations made. Thus, specific n_k^* and T_n^* are used in each run and model structure, which are shown in Table 3.

Table 3 Model parameters n_k^* and T_n^* used in each run and model structure.

Model	Subject 1					Subject 2					Subject 3				
Parameters	N1	N2	N3	N4	N5	N1	N2	N3	N4	N5	N1	N2	N3	N4	N5
ARX: n_k^*, –	21	24	22	28	20	24	22	20	25	24	25	24	24	24	26
BJ: n_k^*, –	30	30	31	32	31	26	28	26	29	27	28	30	26	27	27
BJ: T_n^*, s	0.16	0.13	0.16	0.13	0.11	0.17	0.11	0.13	0.06	0.14	0.10	0.05	0.08	0.15	0.05

Figures 13 and 14 show the discrete-time and HO model coefficients estimation results for ARX and BJ, respectively. In all subjects, both recursive algorithms are capable of detecting HO adaptation, although subject 1 does not seem to generate as much lead in the state s_2 as subjects 2 and 3. As explained in Section V, the RPEM algorithm requires more time to achieve convergence when facing changes in HO parameters, hence, this fact has a strong impact on the BJ performance in real life, seeing that the human operator is not a steady, perfect controller.

Magnitudes of discrete-time parameter estimations are different for ARX and BJ, as can be observed in Figure 13. This fact can be due to the effort of ARX structures in focusing more on HO gain and zero estimation, while RPEM follows an optimization for all model coefficients equally. Modifications in a_1^d and a_2^d coefficients magnitudes directly imply an adjustment in b_0^d and b_1^d, and vice versa. Additionally, incorrect estimations in n_k^* in ARX and BJ models also affect the discrete-time parameter magnitudes.

Oscillations in estimated parameters are more noticeable in BJ than ARX. Since the RPEM algorithm follows a non-linear optimization process, more noise is found in BJ estimations than whether the RLS was used, particularly in

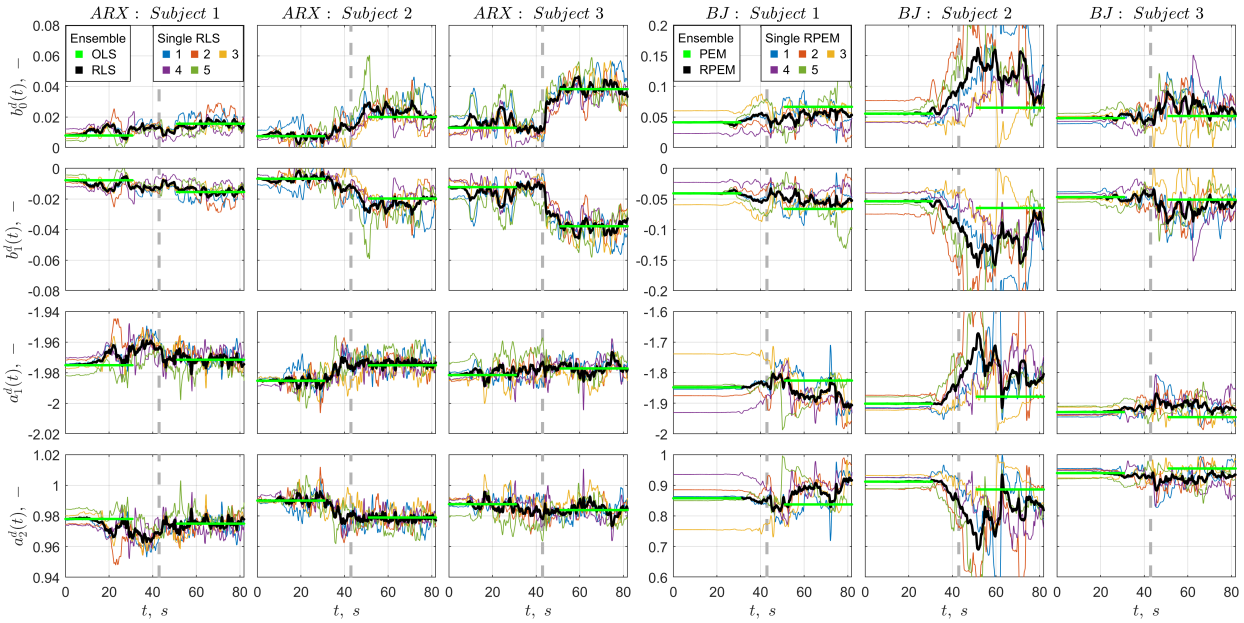

Fig. 13 Experimental results of recursive ARX and BJ algorithms in discrete-time parameters.

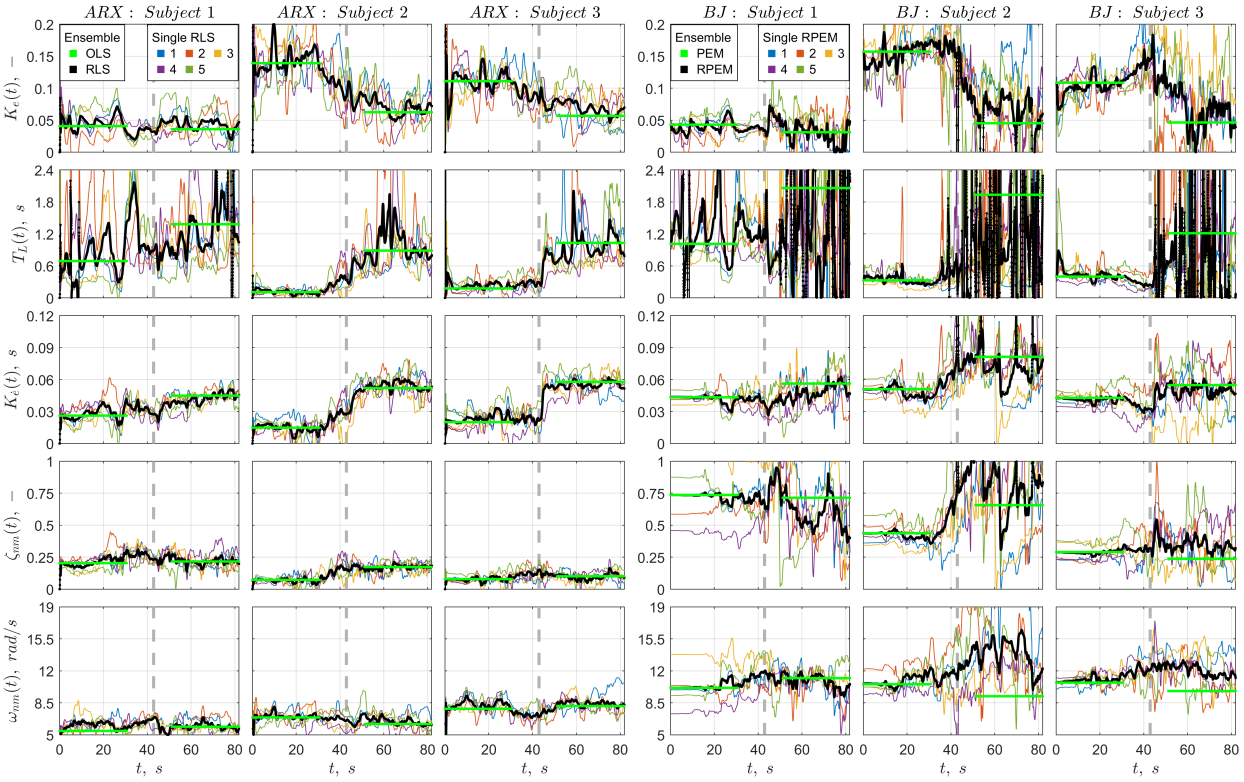

Fig. 14 Experimental results of recursive ARX and BJ algorithms in HO coefficients.

the T_L coefficient. BJ model covariance matrix starts to increase from the start of estimation, since more uncertainty is given about the accuracy of predicted parameters, which turns into convergence issues when the state s_2 is being evaluated.

In addition, estimated NMS coefficients are not expected to show strong oscillations during the entire simulation (relatively small changes are given in real life), but the RPEM algorithm provides an outcome with greater fluctuations. Nevertheless, the average NMS estimations in BJ structures show higher values than in ARX models, which shows that a considerable bias may be generated by the RLS algorithm.

VII. Discussion

A novel time-varying HO identification method based on recursive BJ model structures is developed, which tries to make a precise fit of the HO model coefficients in single-axis manual-control tasks. An optimal version of the algorithm is found for a structure with a model remnant order $m^* = 1$, based on batch-fitting results. An RPEM algorithm, implemented with a constant forgetting factor λ, shows acceptable results when tested on simulation data, since it is able to correctly track HO adaptations to CE dynamics transitions proposed by Zaal [11]. Through testing on experimental data, the recursive BJ is capable of detecting adaptation in subjects, while providing HO coefficient estimations whose magnitudes appear feasible. BJ results in batch and recursive fitting are compared to ARX results to analyze the advantages and drawbacks of estimating HO coefficients by PEM or RPEM (BJ) algorithms instead of OLS and RLS (ARX).

While BJ model structures have great potential as identification method, they do present several problems that can affect their applicability. Firstly, the PEM algorithm is strongly dependent on initial conditions, and although it offers adequate flexibility, initial conditions with high deviations from real values may lead to convergence failure. This fact is quite relevant regarding PEM initialization by a prior OLS estimation, since it is proven that ARX batch-fitting will provide a highly biased output. Furthermore, despite BJ model structure's capability of detecting HO adaptation, the convergence time required can be even 5 times higher due to the non-linear optimization procedure it follows, which can cause identification problems in real life due to the fast changes in HO dynamics. Noticeable fluctuations can be found in recursive BJ estimations, mainly in the parameter T_L, and in the NMS coefficients that normally are expected to be steady during an entire trial.

When comparing ARX and BJ model structures, BJ offers a 5000% more accurate estimation when the remnant order is $m^* = 1$, owing to the mismatch of the ARX model in the remnant filter poles. The BJ model can also prevent the persistent bias found in ARX estimations, because of the versatility of BJ structures when facing different remnant dynamics. Thus, RPEM can provide accurate results in NMS coefficients once the algorithm is converged, while RLS reaches its final estimation fast, but it will always have a bias. Nevertheless, ARX is able to modify the discrete-time parameters in order to still reliably estimate K_e, T_L and $K_{\dot{e}}$, achieving a more adequate, steady fit than BJ for considerably high noise levels.

Still, there is room for improvement in the RPEM algorithm implementation. For instance, the covariance matrix could be updated based on the VAF from a validation data set, in order to reduce the strong estimate oscillations that are observed in state s_2. A forgetting matrix RPEM algorithm could also be implemented, although no MATLAB functions are available for this setup, so that the algorithm should be built based on the theory explained in Section III. Such a forgetting matrix could weigh all samples in discrete-time poles estimation, while applying a certain time horizon for b_0^d and b_1^d, as conducted in [10]. Another strategy would consist of applying the PEM algorithm to sub-batches of data recursively, since this method has shown really high accuracy in all HO coefficient estimations. However, divergence problems may occur due to wrong initialization, or whether the sub-batches are too small. Additionally, other recursive BJ algorithms could be explored, such as the RRIV, which can be implemented through the Captain Toolbox [41]. In particular, this could solve the issues of dependency on initial conditions and convergence rate.

Once the recursive BJ algorithm is refined, conducting a decimation analysis is important to verify the BJ model structure's applicability when different sample times are used, as shown by Van Grootheest et al. [10]. When n_k can not be a multiple of the sample time T_s, or a time-varying delay is given, it is necessary to implement a Padé approximation [47] that represents the effect of time delay by a transfer function. This approximation increases the number of discrete-time parameters to be estimated, thus, it may affect the accuracy and convergence of recursive BJ algorithms. Moreover, other models for remnant dynamics should be evaluated to test the performance of the RPEM algorithm in a real-life scenario.

Experimental results are also affected by the time delay estimation, which may not be optimal in the calculations performed in this article. It is more convenient to have an independent time delay estimation algorithm [12, 22] that

provides an acceptable recursive identification for each subject. This way, the predicted time delay can be introduced into the BJ structure in the RPEM algorithm. In addition, the model remnant time constant T_n could be predicted by means of other methods, although it does not have much effect on the estimation accuracy.

Based on the comparison of RLS and RPEM performance as shown in this paper, an 'identification method fusion' could be an interesting alternative. ARX has been proven to give adequate results for the HO gain and lead contribution, while BJ can remove the bias in NMS coefficients. An ideal identification technique should have the precision of BJ, but also the adaptation velocity of ARX. Therefore, RPEM could be implemented with a forgetting matrix, so NMS estimations performed by RLS can be adjusted based on recursive BJ calculations, diminishing the persistent bias in these coefficients.

VIII. Conclusion

This article lays the foundation for the development of a Human Operator (HO) identification method based on a Box-Jenkins (BJ) model structure that is able to provide an adequate estimation of the time-varying adaptation of a human operator in compensatory tracking tasks. A Monte Carlo simulation analysis, based on the experiment conditions tested by Zaal [11], is conducted to evaluate the performance of the Prediction Error Method (PEM) algorithm (for offline batch-fitting) and the Recursive Prediction Error Minimization (RPEM) (online recursive fitting). The recursive BJ estimator employs a constant forgetting factor $\lambda = 0.99609$, based on earlier research [10], and a model remnant order $m^* = 1$. Furthermore, both Recursive Least Squares (RLS) and RPEM are tested on the experimental data from three subjects. In batch-fitting, BJ is found to fix all bias problems that are typically present in ARX identification results. The remnant filter order no longer meaningfully affects the bias and accuracy in BJ estimation, while it clearly does for ARX. Furthermore, the use of a fixed remnant model order $m^* = 1$ does not change the bias obtained by BJ, which makes it a suitable model structure configuration. Additionally, BJ presents more robustness than ARX when facing high noise levels. For (online) recursive estimation, BJ still offers a reduction in persistent bias found in identified coefficients, while ARX converges to its final estimation 5 times faster, reaching a quicker adaptation to changes in HO parameters. Except for a first-order remnant case, the RLS algorithm usually captures the HO gain and zero with 0-5% less bias, while BJ estimates the HO model's neuromuscular poles with 25% less error. When an $m^0 = 1$ simulation is conducted, the BJ model clearly outperforms ARX. Through experimental testing, it is proven that the slow adaptation of the recursive BJ may lead to excessive fluctuations in the identified HO coefficients. In addition, BJ shows the expected adaptation of K_e, T_L and $K_{\dot{e}}$ less clearly than the ARX outcomes. Overall, this paper contributes to development of enhanced time-varying HO identification methods, by providing a novel non-linear BJ estimation procedure that can improve on current widely-used estimation methodologies, especially for offline batch-fitting applications.

References

[1] McRuer, D. T., "Human Pilot Dynamics in Compensatory Systems," *Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, OH*, 1965. https://doi.org/10.21236/ad0470337.

[2] McRuer, D. T., and Weir, D. H., "Theory of Manual Vehicular Control," *IEEE Transactions on Man-Machine Systems*, Vol. 10, No. 4, 1969, pp. 257–291. https://doi.org/10.1109/TMMS.1969.299930.

[3] McRuer, D. T., and Jex, H., "A Review of Quasi-Linear Pilot Models," *IEEE Transactions on Human Factors in Electronics*, Vol. HFE-8, No. 3, 1967, pp. 231–249. https://doi.org/10.1109/THFE.1967.234304.

[4] McRuer, D. T., Graham, D., and Krendel, E. S., "Manual control of single-loop systems: Part I," *Journal of the Franklin Institute*, Vol. 283, No. 1, 1967, pp. 1–29. https://doi.org/10.1016/0016-0032(67)90112-3.

[5] McRuer, D. T., Graham, D., and Krendel, E. S., "Manual control of single-loop systems: Part II," *Journal of the Franklin Institute*, Vol. 283, No. 2, 1967, pp. 145–168. https://doi.org/10.1016/0016-0032(67)90231-1.

[6] Nieuwenhuizen, F. M., Zaal Pmt, M., Mulder, Zaal, P. M. T., Mulder, M., van Paassen, M., and Mulder, J., "Modeling Human Multichannel Perception and Control Using Linear Time-Invariant Models," *Journal of Guidance Control and Dynamics*, 2008. https://doi.org/10.2514/1.32307.

[7] van Paassen, M., and Mulder, M., "Identification of Human Operator Control Behaviour in Multiple-Loop Tracking Tasks," *IFAC Proceedings Volumes*, Vol. 31, No. 26, 1998, pp. 455–460. https://doi.org/10.1016/S1474-6670(17)40135-2, 7th IFAC Symposium on Analysis, Design and Evaluation of Man-Machine Systems (MMS'98), Kyoto, Japan, 16-18 September 1998.

[8] Stapleford, R., McRuer, D., and Magdaleno, R., "Pilot Describing Function Measurements in a Multiloop Task," *IEEE Transactions on Human Factors in Electronics*, Vol. HFE-8, No. 2, 1967, pp. 113–125. https://doi.org/10.1109/THFE.1967.233628.

[9] Young, L. R., "On Adaptive Manual Control," *Ergonomics*, Vol. 12, No. 4, 1969, pp. 635–674. https://doi.org/10.1080/00140136908931083.

[10] van Grootheest, A., Pool, D. M., van Paassen, M., and Mulder, M., "Identification of Time-Varying Manual-Control Adaptations with Recursive ARX Models," *AIAA*, 2018. https://doi.org/10.2514/6.2018-0118.

[11] Zaal, P. M., "Manual Control Adaptation to Changing Vehicle Dynamics in Roll-Pitch Control Tasks," *Journal of Guidance Control and Dynamics*, 2016. https://doi.org/10.2514/1.g001592.

[12] Plaetinck, W., Pool, D. M., van Paassen, M., and Mulder, M., "Online Identification of Pilot Adaptation to Sudden Degradations in Vehicle Stability," *IFAC-PapersOnLine*, Vol. 51, No. 34, 2019, pp. 347–352. https://doi.org/10.1016/j.ifacol.2019.01.020, 2nd IFAC Conference on Cyber-Physical and Human Systems CPHS 2018.

[13] Hess, R. A., "Modeling Human Pilot Adaptation to Flight Control Anomalies and Changing Task Demands," *Journal of Guidance Control and Dynamics*, 2016. https://doi.org/10.2514/1.g001303.

[14] Hess, R. A., "Modeling Pilot Control Behavior with Sudden Changes in Vehicle Dynamics," *Journal of Aircraft*, 2009. https://doi.org/10.2514/1.41215.

[15] Mulder, M., Pool, D. M., Abbink, D. A., Boer, E. R., Zaal, P. M. T., Drop, F. M., van der El, K., and van Paassen, M. M., "Manual Control Cybernetics: State-of-the-Art and Current Trends," *IEEE Transactions on Human-Machine Systems*, Vol. 48, No. 5, 2018, pp. 468–485. https://doi.org/10.1109/THMS.2017.2761342.

[16] Popovici, A., Zaal, P. M. T., and Pool, D. M., "Dual Extended Kalman Filter for the Identification of Time-Varying Human Manual Control Behavior," *AIAA*, 2017. https://doi.org/10.2514/6.2017-3666.

[17] Ameyoe, A., Chevrel, P., Le-Carpentier, E., Mars, F., and Illy, H., "Identification of a Linear Parameter Varying Driver Model for the Detection of Distraction," *IFAC-PapersOnLine*, Vol. 48, No. 26, 2015, pp. 37–42. https://doi.org/10.1016/j.ifacol.2015.11.110, 1st IFAC Workshop on Linear Parameter Varying Systems LPVS 2015.

[18] Olivari, M., Nieuwenhuizen, F. M., Bülthoff, H. H., and Pollini, L., "Identifying time-varying neuromuscular system with a recursive least-squares algorithm: a Monte-Carlo simulation study," *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 3573–3578. https://doi.org/10.1109/SMC.2014.6974484.

[19] Zaal, P. M. T., and Sweet, B. T., "Estimation of Time-Varying Pilot Model Parameters," *AIAA*, 2011. https://doi.org/10.2514/6.2011-6474.

[20] Zaal, P. M. T., Pool, D. M., Chu, Q., van Paassen, M., Mulder, M., and Mulder, J., "Modeling Human Multimodal Perception and Control Using Genetic Maximum Likelihood Estimation," *Journal of Guidance Control and Dynamics*, 2009. https://doi.org/10.2514/1.42843.

[21] Duarte, R., Pool, D. M., van Paassen, M., and Mulder, M., "Experimental Scheduling Functions for Global LPV Human Controller Modeling," *IFAC-PapersOnLine*, Vol. 50, No. 1, 2017, pp. 15853–15858. https://doi.org/10.1016/j.ifacol.2017.08.2329.

[22] Boer, E., and Kenyon, R., "Estimation of time-varying delay time in nonstationary linear systems: an approach to monitor human operator adaptation in manual tracking tasks," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 28, No. 1, 1998, pp. 89–99. https://doi.org/10.1109/3468.650325.

[23] Jiao, J., Sun, L., Tan, W., Xu, S., and Liu, X., "Identifying Pilot Control Adaptations to Sudden Changes in Aircraft Dynamics," *Journal of Guidance Control and Dynamics*, 2023. https://doi.org/10.2514/1.g007358.

[24] Drop, F. M., Pool, D. M., Mulder, M., and Bulthoff, H. H., "Constraints in Identification of Multi-Loop Feedforward Human Control Models," *IFAC-PapersOnLine*, Vol. 49, No. 19, 2016, pp. 7–12. https://doi.org/10.1016/j.ifacol.2016.10.444, 13th IFAC Symposium on Analysis, Design, and Evaluation of Human-Machine Systems HMS 2016.

[25] Linssen, M., *Identifying Time-Varying Multimodal Manual Control Using Recursive ARX Model Techniques*, TU Delft Library, 2020. URL http://resolver.tudelft.nl/uuid:442f4308-0ea2-41a5-b38c-ee6b1a289f78.

[26] Pool, D. M., Pais, A. V., Vroome, A. D., van Paassen, M., and Mulder, M., "Identification of Nonlinear Motion Perception Dynamics Using Time-Domain Pilot Modeling," *Journal of Guidance Control and Dynamics*, 2012. https://doi.org/10.2514/1.56236.

[27] Lone, M., and Cooke, A., "Review of pilot models used in aircraft flight dynamics," *Aerospace Science and Technology*, Vol. 34, 2014, pp. 55–74. https://doi.org/10.1016/j.ast.2014.02.003.

[28] Metz, L. D., "A Time-Varying Approach to the Modeling of Human Control Remnant," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 12, No. 1, 1982, pp. 24–35. https://doi.org/10.1109/TSMC.1982.4308772.

[29] Levison, W. H., Baron, S., and Kleinman, D. L., "A Model for Human Controller Remnant," *IEEE Transactions on Man-Machine Systems*, Vol. 10, No. 4, 1969, pp. 101–108. https://doi.org/10.1109/TMMS.1969.299906.

[30] van der El, K., Pool, D. M., and Mulder, M., "Analysis of Human Remnant in Pursuit and Preview Tracking Tasks," *IFAC-PapersOnLine*, Vol. 52, No. 19, 2019, pp. 145–150. https://doi.org/10.1016/j.ifacol.2019.12.165, 14th IFAC Symposium on Analysis, Design, and Evaluation of Human Machine Systems HMS 2019.

[31] Olivari, M., Venrooij, J., Nieuwenhuizen, F. M., Pollini, L., and Bulthoff, H. H., "Identifying Time-Varying Pilot Responses: A Regularized Recursive Least-Squares Algorithm," *AIAA*, 2016. https://doi.org/10.2514/6.2016-1182.

[32] Klyde, D. H., Brenner, M. J., and Thompson, P., "Wavelet-based time-varying human operator models." *AIAA*, 2001. https://doi.org/10.2514/6.2001-4009.

[33] Mandal, T., and Gu, Y., "Online Pilot Model Parameter Estimation Using Sub-Scale Aircraft Flight Data," *AIAA*, 2016. https://doi.org/10.2514/6.2016-0636.

[34] Ljung, L., *System Identification: Theory for the User, 2nd Edition*, Prentice Hall Information and System Sciences Series, 2012. https://doi.org/10.1109/MRA.2012.2192817.

[35] Islam, S. A. U., and Bernstein, D. S., "Recursive Least Squares for Real-Time Implementation [Lecture Notes]," *IEEE Control Systems Magazine*, Vol. 39, No. 3, 2019, pp. 82–85. https://doi.org/10.1109/MCS.2019.2900788.

[36] Franklin, G. F., Powell, J., and Workman, M. L., *Digital Control Of Dynamic Systems 3rd Edition*, Ellis-Kagle Press, 1998.

[37] Butcher, J. C., *Numerical methods for ordinary differential equations*, John Wiley & Sons, Ltd, 2003. https://doi.org/10.1002/9781119121534.

[38] Tangirala, A. K., *Principles of System Identification: Theory and Practice*, Taylor & Francis Group, 2017. https://doi.org/10.1201/9781315222509.

[39] Hess, R. A., and Mnich, M. A., "Identification of pilot-vehicle dynamics from in-flight tracking data," *Journal of Guidance Control and Dynamics*, 1986. https://doi.org/10.2514/3.20129.

[40] Young, P. C., *Recursive Estimation and Time-Series Analysis, 2nd Edition*, Springer, 2011. https://doi.org/10.1007/978-3-642-21981-8.

[41] Young, P. C., "Refined instrumental variable estimation: Maximum likelihood optimization of a unified Box–Jenkins model," *Automatica*, Vol. 52, 2015, pp. 35–46. https://doi.org/10.1016/j.automatica.2014.10.126.

[42] Box, G. E. P., and Jenkins, G. M., *Time series analysis forecasting and control*, 5th ed., John Wiley & Sons, Inc., 1970.

[43] Ljung, L., *System Identification Toolbox: User's Guide*, Mathworks, 2023.

[44] Brabenec, R. L., *Resources for the Study of Real Analysis*, Mathematical Association of America, 2004.

[45] Verhaegen, M., and Verdult, V., *Filtering and System Identification: A Least Squares Approach*, Cambridge University Press, 2007.

[46] Gajic, Z., *Linear Dynamic Systems and Signals*, Prentice Hall, 2003.

[47] Vajta, M., *Some remarks on Padé-approximations*, 2000, pp. 53–58. 3rd TEMPUS-INTCOM Symposium on Intelligent Systems in Control and Measurements 2000 ; Conference date: 09-09-2000 Through 14-09-2000.