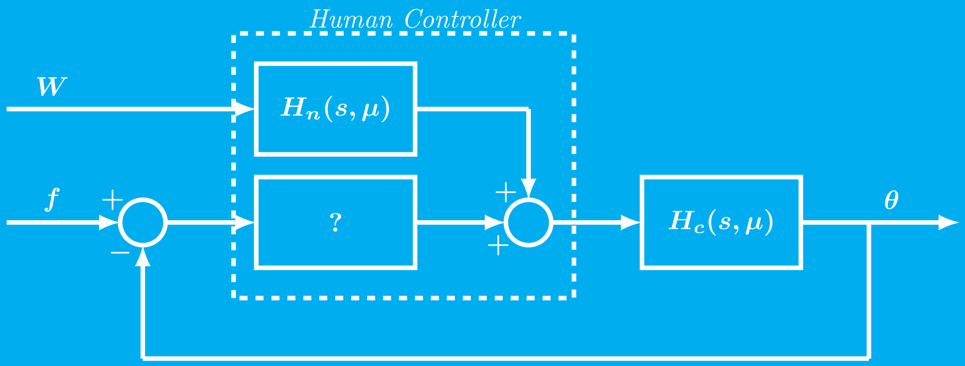
Thesis

Human Operator Identification in the LPV System Framework

R.F.M. Duarte

Technische Universiteit Delft



$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= \sum_{i=1}^{m} \mu_k^{(i)} \left(\boldsymbol{A}^{(i)} \boldsymbol{x}_k + \boldsymbol{B}^{(i)} \boldsymbol{u}_k + \boldsymbol{K}^{(i)} \boldsymbol{e}_k \right) \\
\boldsymbol{y}_k &= \boldsymbol{C} \boldsymbol{x}_k + \boldsymbol{D} \boldsymbol{u}_k + \boldsymbol{e}_k
\end{aligned}
$$

Thesis

Human Operator Identification in the LPV System Framework

by

R.F.M. Duarte

to obtain the degree of Master of Science in Aerospace Engineering
at the Delft University of Technology,
to be defended publicly on Thursday August 25, 2016 at 14:00 PM.

Student number:	4508629	
Project duration:	March 1, 2016 – August 25, 2016	
Thesis committee:	Prof. dr. ir. M. Mulder,	TU Delft, AE
	Dr. ir. M. M. van Paassen,	TU Delft, AE
	Dr. ir. D. Pool,	TU Delft, AE, supervisor
	Dr. ir. H. Vallery,	TU Delft, 3ME

This thesis is confidential and cannot be made public until August 25, 2019.

TU Delft

Preface

An unbelievably long time ago, I initiated the long winding process which would eventually culminate in the final days of this month. As I write these final lines and take a step back to look at the big picture, I wonder: how did I accomplish all of this?

In true fairness, the work contained in this MSc. Thesis is not an individual effort. Far from it, in fact. It was developed with the strength of those who directly helped me with valuable technical advice, those that accompanied me on a daily basis in this Delftian journey, and those who, despite being really far away from me, were always on my mind.

Accordingly, I'd like to thank my main supervisor, dr. ir. Daan Pool, for having the time (and patience) to painstakingly instructing me with valuable technical knowledge, and for showing me the correct way when I was most lost within my research. Dank je hartelijk Daan!

Also, I'd like to thank dr. ir. René van Paassen, whom provided me with various suggestions and insight regarding LPV identification and prof. dr. ir. Max Mulder, whose contagious enthusiasm towards the work developed by Portuguese students boosted my self-confidence.

Of course, I could not forget the small (in number, but HUGE in spirit) Portuguese community in Delft that I'm part of. So, a big thanks to the *Tugas em Delft*, my second family. I may be a handful at times, but they were always there for me.

Last but not least, my family in Portugal, whom I miss everyday. So many people to individually mention! I hope they forgive me for not doing so and, instead, offer a general thank you for supporting me in my decisions and in following this path I tread. However, a special shout-out to Fátima, Paulo Zé and Joana: the single three most important people in my life.

Obrigado por tudo e até já!

R.F.M. Duarte
Delft, August 2016

Contents

List of Figures

List of Symbols

A	Dynamics Matrix
A_t	Sinusoidal amplitude, deg
a_k	Continuous model denominator coefficients
B	Input matrix
b_k	Continuous model numerator coefficients
C	Output matrix
C_{xy}	Covariance matrix
D	Feed-through matrix
e	Operator input, deg
F	PBSID future window, samples
f_c	LTI capturing frequency, Hz
f_s	Sampling frequency, Hz
f_t	Forcing function target signal, deg
G	Sigmoid function maximum rate of change, s^{-1}
H_c	Controlled element dynamics
H_n	Human operator remnant filter
H_p	Human operator dynamics
K	Innovation matrix
K_{xy}	Correlation matrix
K_c	Controlled system gain
K_n	Remnant filter gain
K_v	Human operator visual gain
k	Discrete sample
M	Sigmoid function time of maximum rate of change, s
N	PBSID system order
n	Human operator remnant signal
P_n	Relative remnant signal power
p	PBSID past window, samples
s	Laplace unit, rad s^{-1}
T_L	Operator lead time constant, s
T_m	Measurement time (sinusoidal period), s
T_{sim}	Simulation time, s
t	Time, s
u	Human operator control signal, deg
W	Pseudo-white noise signal
y	Controlled element output signal, deg

ζ_{nm}	Operator neuromuscular damping ratio
μ	Sigmoid function
σ	Standard deviation
σ^2	Variance
τ_v	Operator visual delay, s
ϕ_t	Sinusoidal phase, rad
ω_b	Controlled system break frequency, rad s^{-1}
ω_{nm}	Operator neuromuscular natural frequency, rad s^{-1}
ω_t	Sinusoidal frequency, rad s^{-1}

Introduction

Since the very beginnings of human-machine interaction studies carried out by McRuer et al. [3, 10], extensive research has been done on the identification of the human operator in-the-loop, in both simple single-channel and complex multi-channel tracking tasks [11, 15, 17, 21], in most cases with constant controlled element dynamics. An increased interest in time-varying modelling of the human operator has been emerging, supported on the reasoning that in real-life situations very frequently the human operator needs to adapt himself, either due to a controlled element dynamics that changes over time [25, 27], or simply due to accumulated fatigue, momentary distractions or even the boredom experienced while performing a task.

Numerous time-varying identification techniques have been proposed and previously tested, not only for human-in-the-loop identification purposes, but also for industrial-oriented applications [4, 8, 9, 12, 19, 20, 22, 26]. Within the human in-the-loop framework, recursive least-squares, maximum likelihood and wavelet transform analysis algorithms have been implemented [8, 12, 19, 26]. On the other hand, the Linear Parameter-Varying (LPV) system class has been successfully studied and applied for the identification of time-varying behaviour of industrial systems [4, 9, 20, 22]. An LPV model assumes the system dynamics change over time and depend exclusively on measurable external variables, called Scheduling Variables. A promising example of an LPV system identification technique is the Predictor-Based Subspace Identification (PBSID) algorithm, developed and optimized by Chiuso and van Wingerden et al. [2, 23], and having already been applied for wind turbine LPV identification, with encouraging results. [22]

The innovative use of the LPV framework to identify time-varying human neuromuscular admittance dynamics for steering tasks using the grip force on the steering wheel as Scheduling Variable has recently yielded promising results [16]. However, LPV framework application and subsequent identification of the human in-the-loop operator in aeronautical tracking tasks is still fairly unexplored territory, as selecting an appropriate Scheduling Variable in this context is not a trivial problem.

Most recently, Zaal used a genetic MSE algorithm to identify the adaptation of test subjects to changing controlled element dynamics [27]. However, the assumption that the human operator dynamics change according to the same function that defines the controlled element dynamics change limits the potential for capturing possible non-mapped details and nuances in the evolution of the operator dynamics over time. Due to its reliance on an explicit and possibly experimental Scheduling Variable, the LPV framework could provide an increased freedom in human operator identification, by not requiring strict assumptions on the time-variation of the parameters of the human operator.

The goal of this MSc. Thesis thus is to assess the viability of adapting and applying the LPV framework and PBSID algorithm to the time-varying human operator identification problem. This is achieved by recreating the experiment by Zaal [27] in offline conditions using `Matlab` and identifying the simulated human operator in offline Monte Carlo simulations using an adaptation of the PBSID algorithm. This is accomplished in three phases. Firstly, the tunable settings of the PBSID algorithm are selected based on the comparison of the quality of the models obtained for different algorithm settings. Secondly, using the previously selected PBSID settings, the models are identified using a selection of four Scheduling Variables (two analytical and two experimental) and two conditions (with and without perturbation of the human operator dynamics). These results are then analysed regarding model quality and parametric estimation bias, to evaluate the performance of the candidate experimental Scheduling Variables. Finally, a sensitivity analysis is performed, where the estimated parameters of the identified model are varied in an isolated fashion. The subsequently induced variations in estimated model quality provide a backdrop for analysing the parametric estimation performance, by indicating the expected parameter estimation precision achievable with this framework.

This MSc. Thesis has the following structure. The Scientific Paper that forms the basis of this Thesis is presented in Part I. Part II comprises the appendixes, subdivided in four sections. Appendix A explores the process for obtaining the experimental Scheduling Function μ_{E_1}. Analogously, Appendix B presents the process for obtaining the experimental Scheduling Function μ_{E_2}. In Appendix C, the experimental Scheduling Functions obtained from the Monte Carlo simulations are presented. Finally, Appendix D provides an extensive overlook to the parametric estimation distribution obtained from the Monte Carlo simulations.

Scientific Paper

Implementation of Predictor-Based Subspace Methodology for Human Operator Identification in the LPV System Framework

R.F.M. Duarte [*] , D.M. Pool [†] , M.M. van Paassen [‡] , M. Mulder [§]

Delft University of Technology, Delft, The Netherlands

Human interaction with time-varying systems is becoming commonplace. Driving a car or piloting an aircraft are examples of such human-machine interactions, where the human operator must adapt his control strategy to the changing dynamics of the system. However, changes in control dynamics of the human operator may also be induced by internal factors, namely fatigue, boredom or even a sudden scare. Therefore, the search for a suitable method that can correctly and sharply identify these changes in the operator dynamics is of the highest importance. In this paper, a novel application of the Linear Parameter Varying (LPV) framework to the human operator in single-loop time-varying tracking tasks and subsequent time-varying identification with a Predictor-Based Subspace Identification (PBSID) algorithm is tested. Additionally, two experimentally determined Scheduling Functions, derived from the measured output of the human operator, are tested regarding their model identification performance. An offline simulation analysis based on a recent experimental study was setup and the PBSID algorithm was used to identify the human operator model in different conditions. The results obtained from offline Monte Carlo simulations show good overall model identification, but high noise realization sensitivity. The results further show the possibility of increased freedom in human operator parameter evolution over time when using the LPV framework. An experimental Scheduling Function obtained from zero-phase filtering the second derivative of the human operator output signal was found to capture the time-variation in human operator dynamics, with equivalent accuracy as obtained with analytical Scheduling Functions.

Nomenclature

A	Dynamics Matrix
A_t	Sinusoidal amplitude, deg
a_k	Continuous model denominator coefficients
B	Input matrix
b_k	Continuous model numerator coefficients
C	Output matrix
C_{xy}	Covariance matrix
D	Feed-through matrix
e	Operator input, deg
F	PBSID future window, samples
f_c	LTI capturing frequency, Hz
f_s	Sampling frequency, Hz
f_t	Forcing function target signal, deg
G	Sigmoid function maximum rate of change, s^{-1}
H_c	Controlled element dynamics
H_n	Human operator remnant filter
H_p	Human operator dynamics
K	Innovation matrix
K_{xy}	Correlation matrix
K_c	Controlled system gain
K_n	Remnant filter gain
K_v	Human operator visual gain

[*] MSc Student, Control and Simulation Section, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, The Netherlands; r.f.marquesduarte@student.tudelft.nl.

[†] Assistant Professor, Control and Simulation Section, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, The Netherlands; d.m.pool@tudelft.nl. Member 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.

k	Discrete sample
M	Sigmoid function time of maximum rate of change, s
N	PBSID system order
n	Human operator remnant signal
P_n	Relative remnant signal power
p	PBSID past window, samples
s	Laplace unit, rad s^{-1}
T_L	Operator lead time constant, s
T_m	Measurement time (sinusoidal period), s
T_{sim}	Simulation time, s
t	Time, s
u	Human operator control signal, deg
W	Pseudo-white noise signal
y	Controlled element output signal, deg

Symbols

ζ_{nm}	Operator neuromuscular damping ratio
μ	Sigmoid function
σ	Standard deviation
σ^2	Variance
τ_v	Operator visual delay, s
ϕ_t	Sinusoidal phase, rad
ω_b	Controlled system break frequency, rad s^{-1}
ω_{nm}	Operator neuromuscular natural frequency, rad s^{-1}
ω_t	Sinusoidal frequency, rad s^{-1}

I. Introduction

Since the very beginnings of human-machine interaction studies carried out by McRuer et al.,[1,2] extensive research has been done on the identification of the human operator in-the-loop, in both simple single-channel and complex multi-channel tracking tasks,[3–6] in most cases with constant controlled element dynamics. An increased interest in time-varying modeling of the human operator has been emerging, supported on the reasoning that in real-life situations very frequently the human operator needs to adapt himself, either due to a controlled element dynamics that changes over time[7,8] (for example, an aircraft that stalls, or suffers structural damage), or simply due to accumulated fatigue, momentary distractions or even the boredom experienced while performing a task.

Numerous time-varying identification techniques have been proposed and previously tested, not only for human-in-the-loop identification purposes, but also for industrial-oriented applications.[9–17] Within the human in-the-loop framework, recursive least-squares, maximum likelihood and wavelet transform analysis algorithms have been implemented.[9–12] On the other hand, the Linear Parameter-Varying (LPV) system class has been successfully studied and applied for the identification of time-varying behavior of industrial systems.[13–17] An LPV model assumes the system dynamics change over time and depend exclusively on measurable external variables, called Scheduling Variables. A promising example of an LPV system identification technique is the Predictor-Based Subspace Identification (PBSID) algorithm, developed and optimized by Chiuso and Van Wingerden et al.,[18,19] and having already been applied for wind turbine LPV identification, with encouraging results.[14]

The innovative use of the LPV framework to identify time-varying human neuromuscular admittance dynamics for steering tasks using the grip force on the steering wheel as Scheduling Variable has recently yielded promising results.[20] However, LPV framework application and subsequent identification of the human in-the-loop operator in aeronautical tracking tasks is still fairly unexplored territory, as selecting an appropriate[a] Scheduling Variable in this context is not a trivial problem.

Most recently, Zaal used a genetic MSE algorithm to identify the adaptation of test subjects to changing controlled element dynamics.[8] However, the assumption that the human operator dynamics change according to the same function that defines the controlled element dynamics change limits the potential for capturing possible non-mapped details and nuances in the evolution of the operator dynamics over time. Due to its reliance on an explicit and possibly experimental Scheduling Variable, the LPV framework could provide an increased freedom in human operator identification, by not requiring strict assumptions on the time-variation of the parameters of the human operator.

The goal of this paper thus is to assess the viability of adapting and applying the LPV framework and PBSID algorithm to the time-varying human operator identification problem. This is achieved by recreating the experiment by Zaal[8] in offline conditions using `Matlab` and identifying the simulated human operator in offline Monte Carlo simulations using an adaptation of the PBSID algorithm. This is accomplished in three phases. Firstly, the tunable settings of the PBSID algorithm[b] are selected based on the comparison of the quality of the models obtained for different algorithm settings. Secondly, using the previously selected

[a] Accurate, low-noise and sensitive to changes in system dynamics.

[b] System order N, Past window p and Future Window F.

PBSID settings, the models are identified using a selection of four Scheduling Variables (two analytical and two experimental) and two conditions (with and without perturbation of the human operator dynamics). These results are then analysed regarding model quality and parametric estimation bias, to evaluate the performance of the candidate experimental Scheduling Variables. Finally, a sensitivity analysis is performed, where the estimated parameters of the identified model are varied in an isolated fashion. The subsequently induced variations in estimated model quality provide a backdrop for analysing the parametric estimation performance, by indicating the expected parameter estimation precision achievable with this framework.

Therefore, the paper has the following structure. In Section II, the fundamental mathematical basis of the PBSID algorithm is described. Section III introduces the Zaal setup, identification methodology, performance metrics and the offline testing conditions that are used to achieve the goal of the paper. In Section IV, the results of the PBSID algorithm tunning and offline Monte Carlo simulations for all testing conditions are presented and analyzed. Finally, a discussion on the relevance of the results and subsequent conclusions are presented in Sections V and VI.

II. The PBSID Algorithm

The chosen algorithm for the identification of the human operator LPV model is the Predictor-Based Subspace Identification (PBSID).[14] The reasoning behind this choice lies on the fact that the PBSID is a **global** LPV identification algorithm. Unlike a **local** algorithm it does not interpolate a collection of identified LTI models at isolated instants in time, but rather identifies the system as an explicit LPV model. The lack of model interpolation in the identification process allows for a precise representation of the system dynamics at any given instant in time, which is advantageous for identifying the possible details and nuances in the evolution of the operator dynamics over time.

This section explores the mathematical fundamentals of the PBSID algorithm. The following description is based on Refs. 14 and 19.

A Linear Parameter-Varying (LPV) system state-space model with parameter-independent output equation may be modeled by Eqs. (1).

$$x_{k+1} = \sum_{i=1}^{m} \mu_k^{(i)} \left(A^{(i)} x_k + B^{(i)} u_k + K^{(i)} e_k \right) \tag{1a}$$

$$y_k = C x_k + D u_k + e_k \tag{1b}$$

where k represents the discrete time unit and m is the number of considered Scheduling Variables; $x_k \in \Re^n$, $u_k \in \Re^r$ and $y_k \in \Re^l$ are, respectively, the state, input and output vectors. The white innovation process, $e_k \in \Re^l$, has zero mean and accounts for the error committed when approximating the output y_k with its prediction $\hat{y}_k$, in a one step ahead predictor framework (Eq.(2)).

$$x_{k+1} = \sum_{i=1}^{m} \mu_k^{(i)} \left(\underbrace{\left(A^{(i)} - K^{(i)} C\right)}_{\tilde{A}^{(i)}} x_k + \underbrace{\left(B^{(i)} - K^{(i)} D\right)}_{\tilde{B}^{(i)}} u_k + K^{(i)} y_k \right) \tag{2a}$$

$$y_k = C x_k + D u_k + e_k \tag{2b}$$

The matrices $A^{(i)} \in \Re^{n \times n}$, $B^{(i)} \in \Re^{n \times r}$, $C \in \Re^{l \times n}$, $D \in \Re^{l \times r}$ and $K^{(i)} \in \Re^{n \times l}$ are, respectively, the dynamics, input, output, feedthrough and error intensity matrices. The vector $\mu^{(i)} \in \Re$ is the Scheduling Function (SF), and is comprised of the timed samples k of the i^{th} Scheduling Variable (SV). The Scheduling Function is, in an LPV framework, responsible for changing the system dynamics over time by effectively working as a local weight in the matrices $\tilde{A}^{(i)}$, $\tilde{B}^{(i)}$ and $K^{(i)}$. These matrices depend linearly on the scheduling sequence, which comprises all SFs and is assumed to have an affine dependence: $\mu_k = \begin{bmatrix} 1 & \mu_k^{(2)} & \dots & \mu_k^{(m)} \end{bmatrix}^T$. As a consequence, the time-varying A matrix, for example, becomes:

$$A_k = \sum_{i=1}^{m} \mu_k^{(i)} A^{(i)} \tag{3}$$

The main goal of the algorithm is to introduce a factorization which separates the to-be-identified system matrices from the to-be-assumed scheduling sequence. Before achieving it, however, some definitions need to be introduced:

- **The extended time-varying controllability matrix**

  The transition matrix for discrete-time time-varying systems may be defined as in Ref. 21:

$$\phi_{p,k} = \tilde{A}_{k+p-1}...\tilde{A}_{k+1}\tilde{A}_k \tag{4}$$

  where p is defined as the past window of collected data.

  Using Eq. (4), and grouping the matrices $\tilde{B}$ and K as $\breve{\mathrm{B}}_k = \begin{bmatrix}\tilde{B}_k & K_k\end{bmatrix}^T$, $\bar{B}^{(i)} = \begin{bmatrix}\tilde{B}^{(i)} & K^{(i)}\end{bmatrix}$, the extended time-varying controllability matrix is defined as:

$$\bar{\mathcal{K}}_k^p = \left[\left(\phi_{p-1,k+1}\breve{\mathrm{B}}_k\right), \quad \ldots, \quad \left(\phi_{1,k+p-1}\breve{\mathrm{B}}_{k+p-2}\right), \quad \left(\breve{\mathrm{B}}_{k+p-1}\right)\right] \tag{5}$$

- **The extended time-invariant controllability matrix**

  The operator $\mathcal{L}$ is introduced, which allows for the systematic multiplication of matrices $\tilde{A}$ and $\bar{B}$, in permutations:

$$\mathcal{L}_1 = \begin{bmatrix}\bar{B}^{(1)}, & \ldots, & \bar{B}^{(m)}\end{bmatrix} \tag{6}$$

$$\mathcal{L}_p = \begin{bmatrix}\tilde{A}^{(1)}\mathcal{L}_{p-1}, & \ldots, & \tilde{A}^{(m)}\mathcal{L}_{p-1}\end{bmatrix} \tag{7}$$

  where m represents the number of SVs. Using this operator, the LPV extended time-invariant controllability matrix may be constructed (Eq. (8)).

$$\mathcal{K}^p = \begin{bmatrix}\mathcal{L}_p, & \mathcal{L}_{p-1}, & \ldots, & \mathcal{L}_1\end{bmatrix} \tag{8}$$

- **The Scheduling matrix**

  The Scheduling matrix $\mathcal{N}$ is introduced, aggregating the scheduling sequence within the range of the past window p:

$$\mathcal{N}_k^p = \begin{bmatrix} \mathcal{P}_{p|k} & & & 0 \\ & \mathcal{P}_{p-1|k+1} & & \\ & & \ddots & \\ 0 & & & \mathcal{P}_{1|k+p-1} \end{bmatrix} \tag{9}$$

  where $\mathcal{P}_{p|k} = \mu_{k+p-1} \otimes \mu_{k+p-2} \otimes \cdots \otimes \mu_k \otimes I$, and $\otimes$ represents the Kronecker matrix product.

Finally, the desired factorization of the time-varying extended controllability matrix is achieved:

$$\bar{\mathcal{K}}_k^p = \underbrace{\mathcal{K}^p}_{\text{unknown}} \cdot \overbrace{\mathcal{N}_k^p}^{\text{known}} \tag{10}$$

Note that $\mathcal{K}^p$ depends on the unknown system matrices and $\mathcal{N}_k^p$ depends exclusively on the known scheduling sequence. Furthermore, the number of columns of $\bar{\mathcal{K}}_k^p$ grows with p, through the relation $(r+l)\sum_{j=1}^{p} m^j$. Please recall that r, l, p and m represent, respectively, the number of inputs, number of outputs, past window and number of SVs.

Consequently, the state equation for the modelled states may now be written as:

$$x_{k+p} = \phi_{p,k}x_k + \underbrace{\mathcal{K}^p\mathcal{N}_k^p}_{\bar{\mathcal{K}}_k^p}\,\bar{z}_k^p \tag{11}$$

where $\bar{z}_k^p = \begin{bmatrix} z_k & z_{k+1} & \dots & z_{k+p-1} \end{bmatrix}^T$ and $z_k = \begin{bmatrix} u_k^T & y_k^T \end{bmatrix}^T$. If the system is uniformly exponentially stable, $\phi_{j,k} \approx 0$ for all $j > p$, as the influence of the transition matrix outside the considered past window is assumed to be negligible. This approximation becomes perfect for $p \to \infty$, but might result in biased estimates for finite p.[14] Consequently:

$$x_{k+p} \approx \mathcal{K}^p \mathcal{N}_k^p \bar{z}_k^p \tag{12a}$$

$$y_{k+p} \approx C\mathcal{K}^p \mathcal{N}_k^p \bar{z}_k^p + D u_{k+p} + e_{k+p} \tag{12b}$$

Now, the known data is stacked in matrices:

$\mathcal{U}_i = [u_{p+i}, \quad \dots, \quad u_{N-f+i+1}]\,, \quad \mathcal{Y}_i = [y_{p+i}, \quad \dots, \quad y_{N-f+i+1}]\,, \quad \mathcal{Z}_i = \begin{bmatrix} \mathcal{N}_0^{p+i} \bar{z}_0^{p+i}, & \dots, & \mathcal{N}_{N-p-f}^{p+i} \bar{z}_{N-p-f}^{p+i} \end{bmatrix}.$

If the matrix $\begin{bmatrix} \mathcal{Z}^T & \mathcal{U}^T \end{bmatrix}^T$ has full rank the following linear regression may be solved, to estimate $C\mathcal{K}^{p+i}$ and D:

$$\min_{C\mathcal{K}^{p+i},D} \| \, \mathcal{Y}_i - \left(C\mathcal{K}^{p+i} \mathcal{Z}_i + D\,\mathcal{U}_i \right) \|_F^2 \tag{13}$$

However, the state sequence can not be directly estimated. So, firstly the extended observability matrix is constructed:

$$\Gamma^F = \begin{bmatrix} C \\ C\tilde{A}^{(i)} \\ \vdots \\ C\left(\tilde{A}^{(i)}\right)^{F-1} \end{bmatrix} \tag{14}$$

where F denotes the future window of predicted data. The product of the extended observability and controllability matrices is then computed (Eq. (15)), using Eq. (14), relation (16) and the estimation of $C\mathcal{K}^{p+i}$, previously obtained solving Eq. (13).

$$\Gamma^F \mathcal{K}^p = \begin{bmatrix} C\mathcal{L}_p & C\mathcal{L}_{p-1} & \cdots & C\mathcal{L}_1 \\ C\tilde{A}^{(1)}\mathcal{L}_p & C\tilde{A}^{(1)}\mathcal{L}_{p-1} & \cdots & C\tilde{A}^{(1)}\mathcal{L}_1 \\ \vdots & \vdots & \ddots & \vdots \\ C\left(\tilde{A}^{(1)}\right)^{F-1}\mathcal{L}_p & C\left(\tilde{A}^{(1)}\right)^{F-1}\mathcal{L}_{p-1} & \cdots & C\left(\tilde{A}^{(1)}\right)^{F-1}\mathcal{L}_1 \end{bmatrix} \tag{15}$$

$$C\mathcal{K}^p = [C\mathcal{L}_p, \;\; C\mathcal{L}_{p-1}, \;\; \dots, \;\; C\mathcal{L}_1] \tag{16a}$$

$$C\mathcal{L}_p = \begin{bmatrix} C\tilde{A}^{(1)}\mathcal{L}_{p-1}, & \dots, & C\tilde{A}^{(m)}\mathcal{L}_{p-1} \end{bmatrix} \tag{16b}$$

The extended observability matrix multiplied by the state sequence ($\Gamma^F \mathcal{K}^p \mathcal{Z}$) is then computed, and the state sequence $\hat{X} = \mathcal{K}^p \mathcal{Z}$ is estimated by performing a Singular Value Decomposition:

$$\widehat{\Gamma^F \mathcal{K}^p} \mathcal{Z} = \begin{bmatrix} \mathcal{U} & \mathcal{U}_\perp \end{bmatrix} \begin{bmatrix} \Sigma_N & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} \mathcal{V} \\ \mathcal{V}_\perp \end{bmatrix} \tag{17}$$

where Σ_N denotes a diagonal matrix containing the N largest singular values and $\mathcal{V}$ are the corresponding singular vectors. By selecting only the largest singular values, a reduced number of states can be estimated:

$$\hat{X}_r = \Sigma_N \mathcal{V} \tag{18}$$

Since the states, the input, the output, the feedthrough and the scheduling sequence are now known, the remaining system matrices can be determined from Eq. (1), yielding matrices of the form:

$$A^{(i)} = \begin{bmatrix} a_{11}^{(i)} & a_{12}^{(i)} \\ a_{21}^{(i)} & a_{22}^{(i)} \end{bmatrix} \tag{19}$$

as an example for a generic second order ($N = 2$) $A^{(i)}$ matrix.

While the LPV model structure and PBSID algorithm have been successfully used for the identification of wind turbines,[14,19] this paper focuses on their usage for the purpose of human operator identification in a simple single-axis tracking task.

III. Methods

A. Tracking Task

The manual control scenario which serves as the basis for the identification algorithm testing is that presented in Ref. 8, with minor adjustments to better fit the purpose of LPV algorithm testing. This allowed the algorithm to be tested in a familiar human in-the-loop framework.

In Ref. 8, human control behaviour in a dual-axis roll-pitch control task was experimentally evaluated. For the purpose of simplicity, only a single-axis, i.e., the pitch control task was considered in this paper. The block diagram of the task is shown in Figure 1.

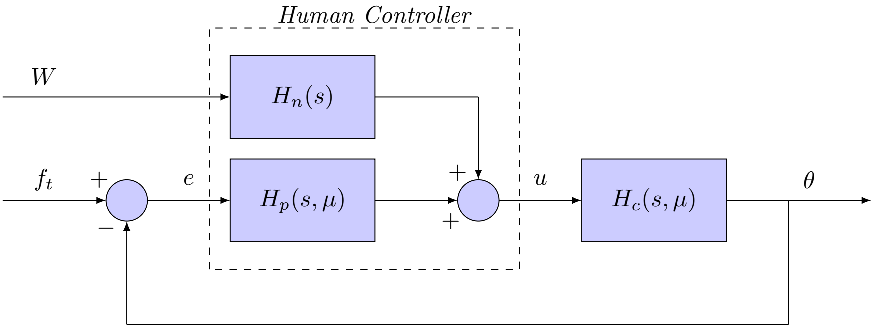


Figure 1. The pitch control task used for testing.

The simplified aircraft pitch dynamics (H_c) and the linear human operator controller and neuromuscular dynamics (H_p) are assumed to have time-varying parameters according to a predefined Scheduling Variable μ, while the human operator remnant filter (H_n) is assumed to be time-invariant.

B. Scheduling Function

In Ref. 8, the time-varying parameters were changed over time according to the sigmoid function:

$$P(t) = P_1 + \frac{P_2 - P_1}{1 + e^{-G(t-M)}} \tag{20}$$

where P_1 represents the initial value of the generic parameter P and P_2 its final value. G is the maximum rate of change of the parameter and M is defined as the time (in seconds) at which it occurs.

For testing purposes, this sigmoid function was defined as the reference analytical scheduling function μ_{A_1}, by introducing the following relation:

$$\mu_{A_1}(t) = \frac{1}{1 + e^{-G(t-M)}} \quad \Longrightarrow \quad P(t) = P_1 + (P_2 - P_1) \cdot \mu_{A_1}(t) \tag{21}$$

where μ_{A_1} denotes, from now on, the reference analytical Scheduling Function within the LPV framework. When $\mu_{A_1} = 0$ (or $t \to -\infty$), $P = P_1$, and when $\mu_{A_1} = 1$ (or $t \to +\infty$), $P = P_2$. Setting $G = 0.5\ s^{-1}$, $M = 50\ s$ and using $t = 0\ s$ and $t = 100\ s$ yields, respectively, $\mu_{A_1} = 1.4 \times 10^{-11}$ and $\mu_{A_1} = 1$, which represent good approximations. In Ref. 8, two values of G were tested to assess the effect of the rate of change of the controlling dynamics in the human operator. In this paper, the focus is on evaluating the performance of the PBSID algorithm for a human in-the-loop task. Hence, fixed $G = 0.5\ s^{-1}$ and $M = 50\ s$ were used. The corresponding time variation of μ_{A_1} is represented in Figure 2.

When performing LPV identification, a single Scheduling Function was always considered, to further simplify the problem.

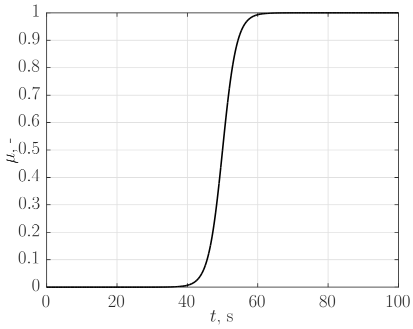


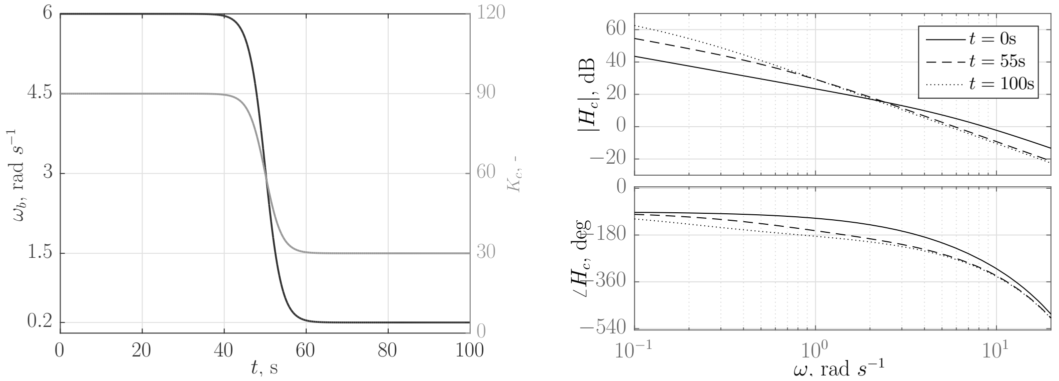
Figure 2. The sigmoid scheduling function μ_{A_1}, used for reference testing purposes.

C. Controlled System Dynamics

The controlled dynamics, represented in Figure 1 as H_c, is defined by the transfer function:

$$H_c(s, \mu) = \frac{K_c(\mu)}{s^2 + \omega_b(\mu)s} \tag{22}$$

where both the break frequency w_b and the static gain K_c change over time according to the relation presented in Eq. (21). The initial and final parameter values were set according to Ref. 8, and Figure 3(a) represents the time variation of both parameters throughout a simulation. The initial and final values of the parameters of H_c are also compiled in Table 1. The controlled system dynamics changes, thus, from mainly single-integrator dynamics to double-integrator dynamics (Figure 3(b)).



(a) Time variation of ω_b, from 6 to 0.2 rads^{-1} and K_c, from 90 to 30.

(b) Frequency and phase response of H_c over time.

Figure 3. Controlled element dynamics variation over time.

D. Linear Human Operator Dynamics

According to the human operator model developed by McRuer et al.,[1] the human operator is expected to close the loop in a way that the closed-loop frequency response approximates that of a single-integrator system, around the crossover frequency. In this case, the human operator compensation dynamics is expected to be mostly a gain in the early stages of the simulation, and both a gain and a lead component in the later stages of the simulation: a result verified in Ref. 8.

Hence, for the linear human operator model H_p, a time-varying visual gain K_v, a time-varying lead term T_L, a constant visual delay τ_v and neuromuscular limitations ω_{nm} and ζ_{nm} are assumed, as shown in

Eq. (23).[8]

$$H_p(s,\mu) = K_v(\mu) \cdot [1 + T_L(\mu)s] \cdot e^{-s\tau_v} \cdot \frac{\omega_{nm}^2}{s^2 + 2\zeta_{nm}\omega_{nm}s + \omega_{nm}^2} \tag{23}$$

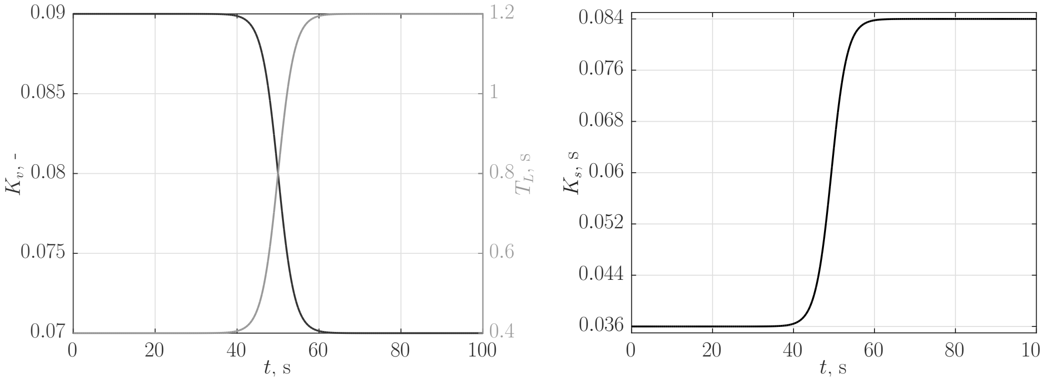
where both the human operator gain K_v and the lead term T_L change over time according to Eq.(21), with initial and final conditions according to the results obtained by Zaal,[8] as shown in Figure 4(a). The initial and final values of the parameters of H_p are compiled in Table 1.

1. *Alternative Parametrization*

The initial phases of testing showed the estimation of the human operator equalisation parameters K_v and T_L was abnormally biased, as evidenced by the strong correlation between K_v and T_L in Eq. 23, for dominant T_L. This was solved by utilizing an alternative parametrization:

$$H_p(s,\mu) = [K_s(\mu)s + K_v(\mu)] \cdot e^{-s\tau_v} \cdot \frac{\omega_{nm}^2}{s^2 + 2\zeta_{nm}\omega_{nm}s + \omega_{nm}^2} \tag{24}$$

where $K_s(\mu) = K_v(\mu) \cdot T_L(\mu)$. The evolution of the parameter K_s over time is represented in Figure 4(b), and its initial and final values are compiled in Table 1.



(a) Time variation of K_v and T_L.

(b) Time variation of K_s.

Figure 4. Evolution of the human operator equalisation parameters over time.

Table 1. Initial and final values for the parameters of H_c and H_p.

	H_c		H_p					
	K_c	ω_b, rad s^{-1}	K_v	T_L, s	K_s, s[†]	ω_{nm}, rad s^{-1}	ζ_{nm}	τ_v, s
$t = 0$ s	90	6	0.09	0.4	0.036	11.25	0.35	0.28
$t = 100$ s	30	0.2	0.07	1.2	0.084			

[†] $K_s = K_v \cdot T_L$.

E. Non-Linear Human Operator Dynamics

The non-linear human operator dynamics was simulated by having a pseudo-white noise signal W passing through a low-pass remnant filter H_n. This filter was defined as $H_n = K_n/(0.2s+1)$, and the gain K_n was chosen so that the power of the remnant signal in the initial single-integrator dynamics phase of the simulation, $P_n = \sigma_n^2/\sigma_u^2$, is as desired. Since no time variation of K_n is assumed, a lower remnant signal power towards the end of the simulation ensues, as the controlling activity increases. The resulting filtered white-noise signal is then added to the linear response of the human operator, to produce its output u.

F. Forcing Function

THE forcing function was defined in an analogous way as in Ref. 8. Although the simulation time was increased from 81.92 to 100 seconds[c], the sinusoidal period was kept at $T_m = 81.92$ seconds, which allowed the use of the same forcing function as in Ref. 8.

The forcing function was obtained by summing ten individual sinusoidal functions:

$$f_t(t) = \sum_{i=1}^{10} A_t(i) sin\left[\omega_t(i)t + \phi_t(i)\right] \tag{25}$$

with their respective amplitudes (A_t), frequencies (w_t) and phases (ϕ_t) listed in Table 2. The length of the simulation was defined at $T_{sim} = 100$ s, and the sampling frequency at $f_s = 100$ Hz.

For each sinusoidal component, the frequency was defined as an integer multiple of the measurement-time base frequency: $\omega_t(i) = n_t(i)\frac{2\pi}{T_m}$.

The amplitudes of the sinusoidal components were defined as the absolute value of a second-order low-pass filter at a specific sinusoid frequency:

$$A_t(i) = \left| \frac{\left[1 + 0.1j\omega_t(i)\right]^2}{\left[1 + 0.8j\omega_t(i)\right]^2} \right| \tag{26}$$

This ensures that the high frequency sinusoids have lower amplitude. The obtained amplitudes were then scaled down to obtain a forcing function with standard deviation of 1.5 deg.[8]

The sinusoidal phases were obtained by generating a large number of random phase sets. The set that yields a probability distribution closest to a Gaussian distribution and without excessive peaks was selected.[8]

Table 2. Forcing function parameters.

i	w_t, rad s^{-1}	A_t, deg	ϕ_t, rad
1	0.230	1.186	−0.753
2	0.384	1.121	1.564
3	0.614	0.991	0.588
4	0.997	0.756	−0.546
5	1.687	0.447	0.674
6	2.608	0.245	−1.724
7	4.065	0.123	−1.963
8	6.596	0.061	−2.189
9	10.661	0.036	0.875
10	17.564	0.025	0.604

G. Off-the-loop delay estimation

SINCE the used LPV model has a discrete state-space model and the considered value of τ_v was 0.28 seconds, with the sampling frequency of $f_s = 100$ Hz, this yields around 28 samples of delay between input and output. In order to capture this delay, the estimated LPV model would need to be of at least 28^{th} discrete order ($N > 28$), which is inefficient and requires a significant computational effort. The solution was to pre-estimate the delay by iteratively estimating lower order LPV models with different amounts of input-output delay compensation. The delay setting which yielded the highest VAF value was then chosen as the estimated $\hat{\tau}_v$. By removing the corresponding input-output delay, the system could be effectively estimated by a lower order discrete model, with $\hat{\tau}_v$ added a posteriori, after the formal identification process. Figure 5 provides a schematic overview of the entire identification process.

[c] to provide approximately 40 seconds of simulation data for both initial and final system conditions, while maintaining $M = 50$ seconds.

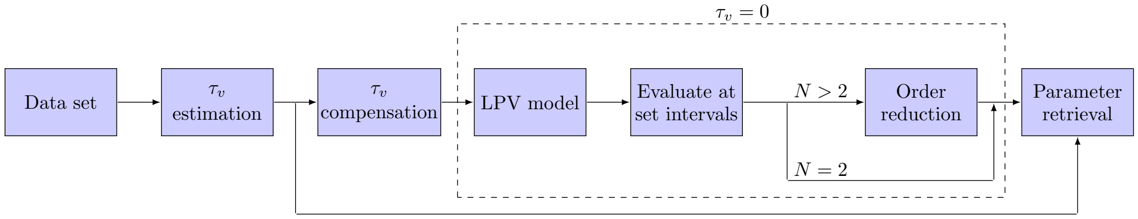


Figure 5. General overview of the parameter estimation process (N represents the identified system order).

H. Retrieving the model parameters

In the testing conditions used, the PBSID provides the human operator model estimation, H_p, as presented in Eq. (1). The corresponding LTI models can be obtained by fixing values of the Scheduling Function μ, obtaining:

$$x_{k+1} = A_1 x_k + (B_1 + B_2\mu_k)\, u_k \tag{27a}$$

$$y_k = C x_k + D u_k \tag{27b}$$

where the matrices A_1 and B_1 are the dynamics and input matrices of the initial system, at $\mu = 0$. The matrix B_2 is the time-varying input matrix, which is multiplied with the Scheduling Variable μ and induces a variation in the system dynamics according to the Scheduling Function. The A matrix does not depend on the Scheduling Function ($A_2 = 0$), as the neuromuscular parameters are not expected to vary over time. Considering only the visual gain and lead time constant are time-varying, and to alleviate the computational burden, only the B matrix was considered time-varying.

The defined model for H_p is SISO and of second-order (Eq. (24)), thus a series of intermediate steps were required in order to obtain an estimation of the model parameters from Eq. (27):

- The obtained set of local LTI models (Eq. (27)) was converted from discrete to continuous-time, using *zero-order hold*.
- The continuous-time state-space models were then converted to the SISO transfer-function form.
- If $N > 2$, a balanced order reduction to second order with matching DC gain was performed (Figure 5).[22]

The final estimated local LTI continuous-time models, with added $\hat{\tau}_v$ have the following structure:

$$\hat{H}_p^i = \frac{\mathcal{U}(s)}{\mathcal{E}(s)} = \frac{b_0^i + b_1^i s}{a_0^i + a_1^i s + s^2} \cdot e^{-\hat{\tau}_v s} \tag{28}$$

with respect to the i^{th} local LTI model. Comparing Eq. (28) with Eq. (24), it is possible to write:

$$\hat{\omega}_{nm}^i = \sqrt{a_0^i} \qquad \hat{\zeta}_{nm}^i = \frac{a_1^i}{2\sqrt{a_0^i}} \qquad \hat{K}_v^i = \frac{b_0^i}{a_0^i} \qquad \hat{K}_s^i = \frac{b_1^i}{b_0^i} \tag{29}$$

I. Performance Metrics

The quality of the estimated second-order model depends not only on how good its model parameters are estimated, but also on the quality of the corresponding Scheduling Function. In order to assess the inherent quality of the obtained models with respect to the reference model, some metrics are introduced.

These performance metrics were applied divided into regions of interest, to better assess the performance of the system identification with respect to the changing human operator dynamics. These regions are represented in Figure 6.

In Region I, the human operator equalization is predominantly a gain; Region II is the transition region; in Region III, the human operator dynamics is the combination of a gain and lead compensation.

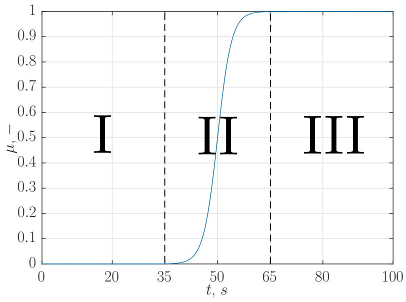


Figure 6. The three regions of testing assessment, with μ represented.

As Region II is the crucial transition phase in between dynamics, the VAF and Correlation between Scheduling Functions results were obtained for the full simulation and for the transition Region II. The parameter estimation bias results were obtained for all of the regions (I - III).

1. Variance Accounted For

The VAF is a metric that indicates the percentage of identification data that is explained by the estimated model, and it is given by Eq. (30).

$$\text{VAF} = \left(1 - \frac{\sum_{k=1}^{N} \left(\hat{u}(k) - u(k)\right)^2}{\sum_{k=1}^{N} \left(u(k)\right)^2}\right) \times 100\% \tag{30}$$

where $\hat{u}(k)$ and $u(k)$ are the estimated and real output of H_p, at sample k. The higher the VAF value, the better the model fit to the data.

2. Average Bias of Parameter Estimation

In order to obtain insight as to how good an estimated model is, the VAF alone is not sufficient, as it gives little information regarding the quality of the model parameters themselves. For the purpose of evaluating the parameter estimation, the average absolute bias over time provides a simple and meaningful metric:

$$\bar{B}_{\hat{\theta}} = \frac{1}{N} \sum_{k=1}^{N} \left| \frac{\theta(k) - \hat{\theta}(k)}{\theta(k)} \right| \tag{31}$$

where $\theta(k)$ is the real value of the generic parameter θ (e.g., K_v) at sample k, and $\hat{\theta}(k)$ is the estimated value of the same parameter, at the same instant. $\bar{B}_{\hat{\theta}}$ is the average bias of the estimation of parameter θ over a single noise realization.

3. Correlation between Scheduling Functions

The Correlation between the reference analytical Scheduling Function μ_A and the to-be-tested experimental Scheduling Function μ_E is introduced as a simple indicator of the quality of μ_E. This allows for an objective comparison between potential experimental Scheduling Function candidates, and can be computed as in Eq. (32).[23]

$$K_{\mu_A,\mu_E} = \frac{C_{\mu_A,\mu_E}}{\sigma_{\mu_A}\sigma_{\mu_E}} \tag{32}$$

where K_{μ_A,μ_E} is the correlation matrix, of which the off-diagonal element is taken as the correlation metric. C_{μ_A,μ_E} is the covariance matrix and σ represents the standard deviation. A value of $K_{\mu_A,\mu_E} = 1$ indicates the experimental Scheduling Function is perfect, while a value of $K_{\mu_A,\mu_E} = 0$ indicates a completely unusable Scheduling Function.

J. Off-line Testing

In order to test the PBSID algorithm for the time-varying pitch-control task, a series of three tests were devised:

- PBSID Tuning Tests;
- Scheduling Function Tests;
- Sensitivity Analysis.

All were offline simulation scenarios performed with three different levels of human operator remnant power, as specified in Section III-E: $P_n = 0.05, 0.15, 0.25$, and with the same forcing function for all conditions and realizations (see Section III-F).

1. PBSID Tuning Tests

The goal of these tests was to objectively choose the optimal settings for the PBSID algorithm, namely the past window P, the future window F and the estimated LPV system order N. The chosen tuning would therefore be used for all the remaining tests.

Several Monte Carlo simulations with 100 different noise realizations were performed, using different combinations of algorithm settings. The tunable parameters and their respective range of testing are compiled in Table 3.

Table 3. Testing range of the tunable parameters of the PBSID alogrithm.

PBSID Parameter	Range	Step
p	$[15, 120]$	$30^{\dagger}$
N	$[2, 14]$	2

† A step of 15 was used, to go from $p = 15$ to $p = 30$. A step of 30 was subsequently used until the end of the range.

In Table 3, p is the past window of the algorithm (given in samples). The parameter N represents the desired order of the identified system. It was decided that the desired system order and the Future Window F would always share the same value, as it significantly reduces the number of testing combinations and decreases the computational effort.[14] The main drawback of using a small F is the truncation of the extended observability matrix (Eq. 14). Through Eq. 15 and Eq. 17, the state sequence $\hat{X}$ will be reduced, but the number of reduced states determined by the Singular Value Decomposition (Eq. 18) is expected to be low anyway.

The definitive tuning is the one that guarantees that the VAF of the identified model is high, but more importantly, that there is no drop in VAF after the model order reduction.

2. Scheduling Function Tests

In the LPV framework, the Scheduling Function is critical, as it directly drives the time variation of the modelled system dynamics.[19] For the LPV identification of a human controller, a suitable, measurable Scheduling Variable must be found, so that a Scheduling Function that reflects the change in dynamics of the human operator can be constructed.

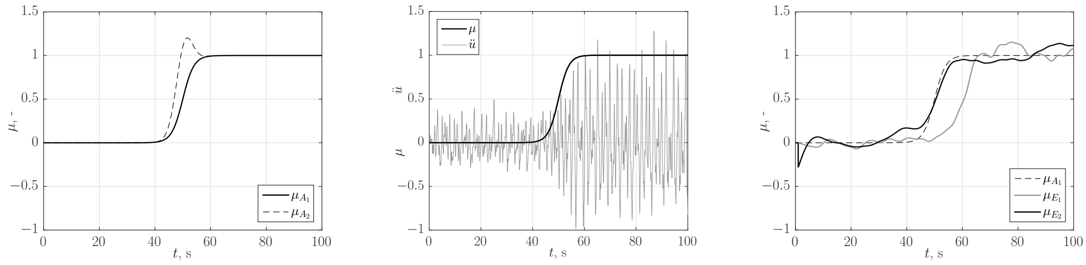
Therefore, the Scheduling Function tests focus on comparing different Scheduling Functions regarding their performance on the identification of the LPV model, with data collected from Monte Carlo runs of 100 noise realizations each.

Four Scheduling Functions were selected for testing: two of analytical origin (μ_A) and two of experimental origin (μ_E). Table 4 compiles the testing conditions.

Table 4. Scheduling Function testing conditions.

H_p Condition	Scheduling Function			
	Analytical		**E**xperimental	
	μ_{A_1}	μ_{A_2}	μ_{E_1}	μ_{E_2}
$\bar{P}$	$\bar{P}_{A_1}$	$\bar{P}_{A_2}$	$\bar{P}_{E_1}$	$\bar{P}_{E_2}$
P	P_{A_1}	P_{A_2}	P_{E_1}	P_{E_2}

- **Analytical Scheduling Function** μ_{A_1} (Figure 7(a)) is the reference Scheduling Function, previously defined in Section III-B.
- **Analytical Scheduling Function** μ_{A_2} (Figure 7(a)) consists on the Scheduling Function μ_{A_1} with a "perturbation" added to it. This perturbation is a Gaussian curve with average at 50 seconds, standard deviation of eight seconds and amplitude of 0.6.
- **Experimental Scheduling Function** μ_{E_1} (Figure 7(c)) is based on the second derivative of the human operator output, $\ddot{u}$. Figure 7(b) indicates that $\ddot{u}$ might hold crucial information about the evolution of the human operator system dynamics over time. However, to be usable as a Scheduling Function, the raw data of $\ddot{u}$ had to undergo some post-processing. Here, $\ddot{u}$ was treated with successive RMS filterings,[24] where a movable filtering window ten samples wide was used. The high amount of RMS filtering was necessary to guarantee the usability of μ_{E_1} as Scheduling Function, specially in high human operator remnant conditions. Afterwards, the filtered $\ddot{u}$ was normalized, to ensure a fair comparison between Scheduling Functions.
- **Experimental Scheduling Function** μ_{E_2} (Figure 7(c)) treats $\ddot{u}$ with a zero-phase digital filtering, processing the data with a linearly optimized Butterworth low-pass filter in both forward and reverse directions.[25] The optimization process makes use of a cost function which compares the resulting μ_{E_2} with the corresponding experimental Scheduling Function μ_A and strives to minimize the error by tweaking the filter coefficients.



(a) The analytical Scheduling Functions μ_{A_1} and μ_{A_2}.

(b) μ_{A_1} and $\ddot{u}$ over time. The change in control activity coincides with the change in dynamics.

(c) Time traces of μ_{E_1} and μ_{E_2}, for a single realization with $P_n = 0.05$.

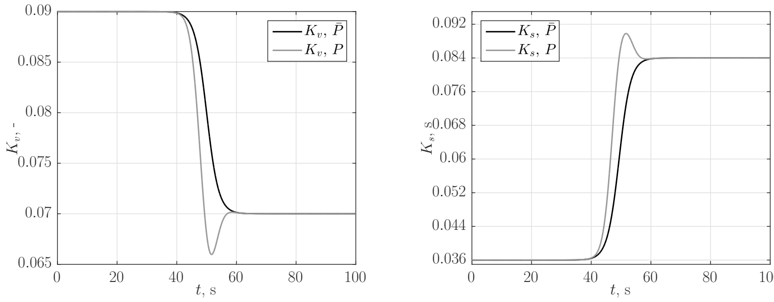
Figure 7. Analytical and experimental Scheduling Functions.

The four Scheduling Functions were tested by performing the identification of H_p for two different conditions:

- **Condition $\bar{P}$:** without the Gaussian perturbation in H_p parameters K_v and K_s.
- **Condition P:** The same Gaussian curve that is included in μ_{A_2} is added to K_v and K_s.

The time traces of K_v and K_s for both $\bar{P}$ and P conditions are represented in Figure 8.

A consequence of this setup is that the Scheduling Function μ_{A_1} serves as the reference for the H_p condition $\bar{P}$, while μA_2 is the reference for condition P.



(a) Time variation of K_v, with conditions $\bar{P}$ and P.

(b) Time variation of K_s, with conditions $\bar{P}$ and P.

Figure 8. H_p parameters for the two testing conditions.

3. *Sensitivity Analysis*

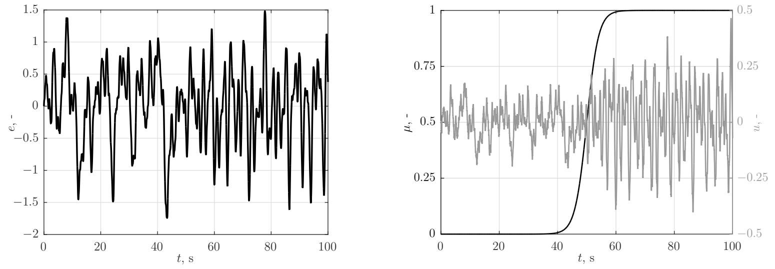
Finally, a sensitivity analysis was performed in the estimated human operator model parameters, consisting of a single PBSID model estimation using a unique realization of noise, followed by the calculation of the VAF as a function of variations in each of the parameters (while maintaining the others constant and equal to the result of the PBSID estimation), for Region I and Region III.

This analysis was important for deducing the parameter estimation precision expectable from the algorithm, for a given parametrization. This analysis will further show that the change in model parametrization (K_v, K_s) presented in Section D-1 delivers better estimation accuracy than the traditional human control model definition (K_v, T_L).

IV. Results

In this section, the results for the three tests described in Section III-J are presented. Section IV-A presents the results of the PBSID Tuning Tests; the results of the Scheduling Function Tests are presented in Section IV-B, and Section IV-C presents the results of the Sensitivity Analysis.

As an example of the data used for the human operator identification, Figure 9 shows the evolution of the human operator tracking error (input), as well as its output over time. Please note how the control activity amplitude changes with μ.



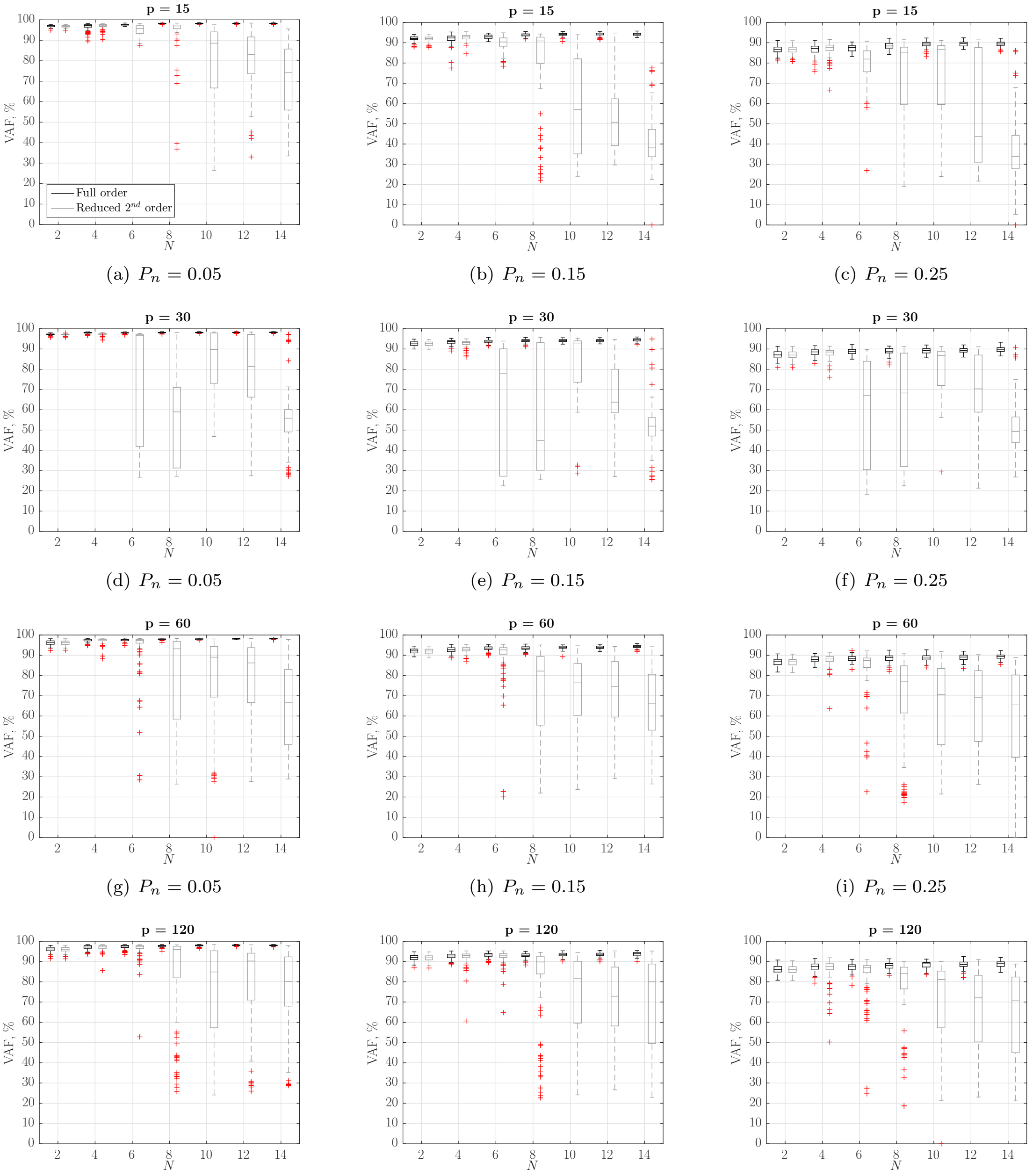
(a) Time trace of human operator tracking error e.

(b) Evolution of the human operator output u, and μ_{A_1}.

Figure 9. Evolution of the human operator performance over time.

A. PBSID Tuning

Figure 10 shows a comparison between the VAF obtained for the high-order identified LPV models and their respective balanced order reductions to second order, for different values of P_n, p and N.



(a) $P_n = 0.05$ (b) $P_n = 0.15$ (c) $P_n = 0.25$

(d) $P_n = 0.05$ (e) $P_n = 0.15$ (f) $P_n = 0.25$

(g) $P_n = 0.05$ (h) $P_n = 0.15$ (i) $P_n = 0.25$

(j) $P_n = 0.05$ (k) $P_n = 0.15$ (l) $P_n = 0.25$

Figure 10. VAF of identified and reduced LPV models for different N, p and P_n.

As expected, it is possible to identify an increasing tendency for the VAF with increasing model order. On the other hand, there is a clear break point after which VAF is lost due to the order reduction. This point is always identified as beginning with the reduction from sixth to second order. The VAF difference between using fourth or sixth models is generally small, but for $P_n = 0.25$, $N = 4$ yields better results for the reduced model. However, the difference in VAF between $N = 4$ and $N = 2$ is not significant. Furthermore, by using $N = 2$, the system order reduction step is not needed, eliminating the errors in parameter estimation potentially introduced by the system order reduction.

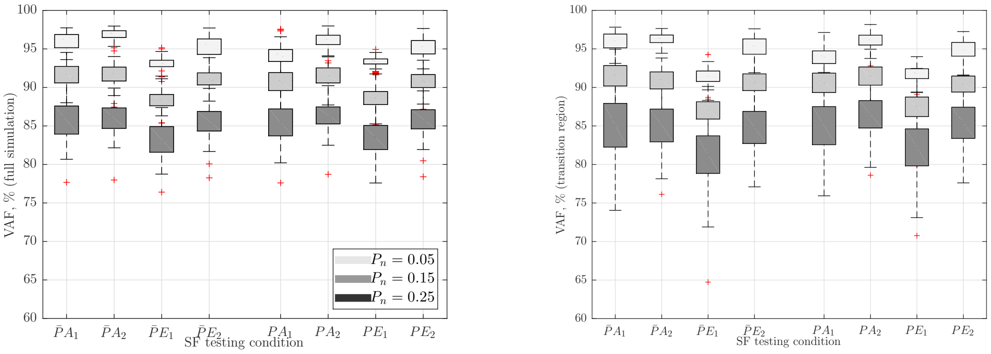
The comparison between different p values shows firstly a tendency for high order models to be more successfully reduced with higher p. There seems not to be any visible trend for the VAF to further increase with p, and the computational burden increases with p.[19] However, the higher the past window value p, the lower the truncation error in the state predictions.[14] A compromise was made and $p = 120$ was selected to both reduce parameter estimation bias while maintaining the computation time at acceptable levels.

From this point forward, all testing was performed with PBSID settings $F = N = 2$ and $p = 120$.

B. Scheduling Functions

The results obtained in this section are based on the models obtained using $F = N = 2$ and $p = 120$ as PBSID settings.

Figure 11 reveals once more that as the human operator remnant power P_n increases, the VAF decreases. This is a general and natural trend. However, for both figures, the maximum attainable VAF values for $P_n = 0.05$, 0.15 and 0.25 are higher than the normally expected values of 95, 85 and 75%, respectively. This is due to the time-invariance of the human remnant filter, which yields comparatively lower P_n values for Regions II and III, and, consequently, overall higher VAF values.



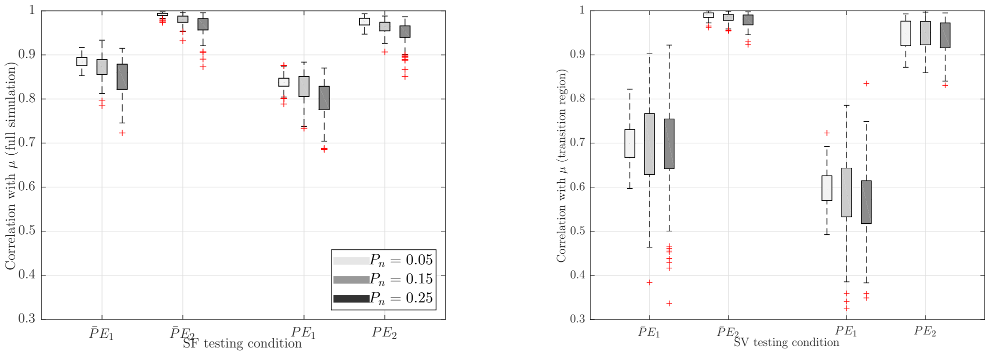
(a) Full simulations.

(b) Region II.

Figure 11. VAF of identified reduced models, for different SF conditions and $P_n = 0.05$, 0.15 and 0.25.

In further detail, Figure 11(a) represents the distribution of the VAF values attained in the Monte Carlo runs for the different SF and P_n conditions. For no H_p perturbation conditions ($\bar{P}$), the SF μ_{A_1} yields the highest VAF values for every P_n level bar $P_n = 0.05$, being the reference SF for this condition. Remarkably, the SF μ_{E_2} presents very similar results for every P_n value, providing good indications about the usefulness of this experimental SF in the identification process. The SF μ_{A_2} is very similar to μ_{A_1}, the difference being the addition of the perturbation in Region II. As such, the results are quite similar between the two analytical SFs, with both obtaining quite similar scores for condition $\bar{P}$ and μ_{A_2} obtaining higher scores for condition P - where μ_{A_2} is the reference. The VAF values of the models identified with μ_{E_1} are generally lower than with μ_{E_2}, for both conditions. These trends are maintained throughout Region II (Figure 11(b)), but the differences between μ_{A_1}, μ_{A_2} and μ_{E_1}, μ_{E_2} are now more pronounced. In fact, this is a critical region: the perturbation that differentiates μ_{A_1} and μ_{A_2} is defined in this region, and the penalization of using an unsuitable SF for identification is clearly visible. Regarding the experimental SFs, the heavy RMS filtering needed to make $\ddot{u}$ usable as an experimental SF introduces a delay in μ_{E_1}, which is most clearly perceptible in Region II. This in turn accentuates the VAF difference between μ_{E_1} and μ_{E_2} in this region and fleshes out the dangers of phase-altering filtering Scheduling Functions.

Figure 12 further explores the comparison between the experimental Scheduling Functions by showing their correlation with the respective reference analytical SF, for different P_n levels. It is perceptible that for every condition, μ_{E_2} is better correlated with either μ_{A_1} or μ_{A_2} than μ_{E_1}. Furthermore, the correlation values concerning μ_{E_2} are consistently above 0.9, a value μ_{E_1} only seldom achieves. It is also noticeable a general trend for the correlation to decrease as the condition changes from $\bar{P}$ to P, which suggests both experimental SFs have some trouble detecting small perturbations in the human operator dynamics. Focusing on Region II, it is possible to observe a high correlation variance concerning μ_{E_1}, which indicates a severe lack of consistency. When comparing the results of the full simulation with those of Region II, a general drop in correlation becomes apparent. This drop is strong for μ_{E_1}, due to its relative delay; it is, however, much smaller for μ_{E_2}, further evidencing the superior SF quality of μ_{E_2} over μ_{E_1}.



(a) Full simulations.

(b) Region II.

Figure 12. Correlation of experimental SFs with their analytical counterparts, for different SF conditions and $P_n = 0.05$, 0.15 and 0.25.

Figure 13 concerns the normalized bias of the parameters estimated using different SFs, H_p conditions and P_n values in Monte Carlo simulations of 100 runs.

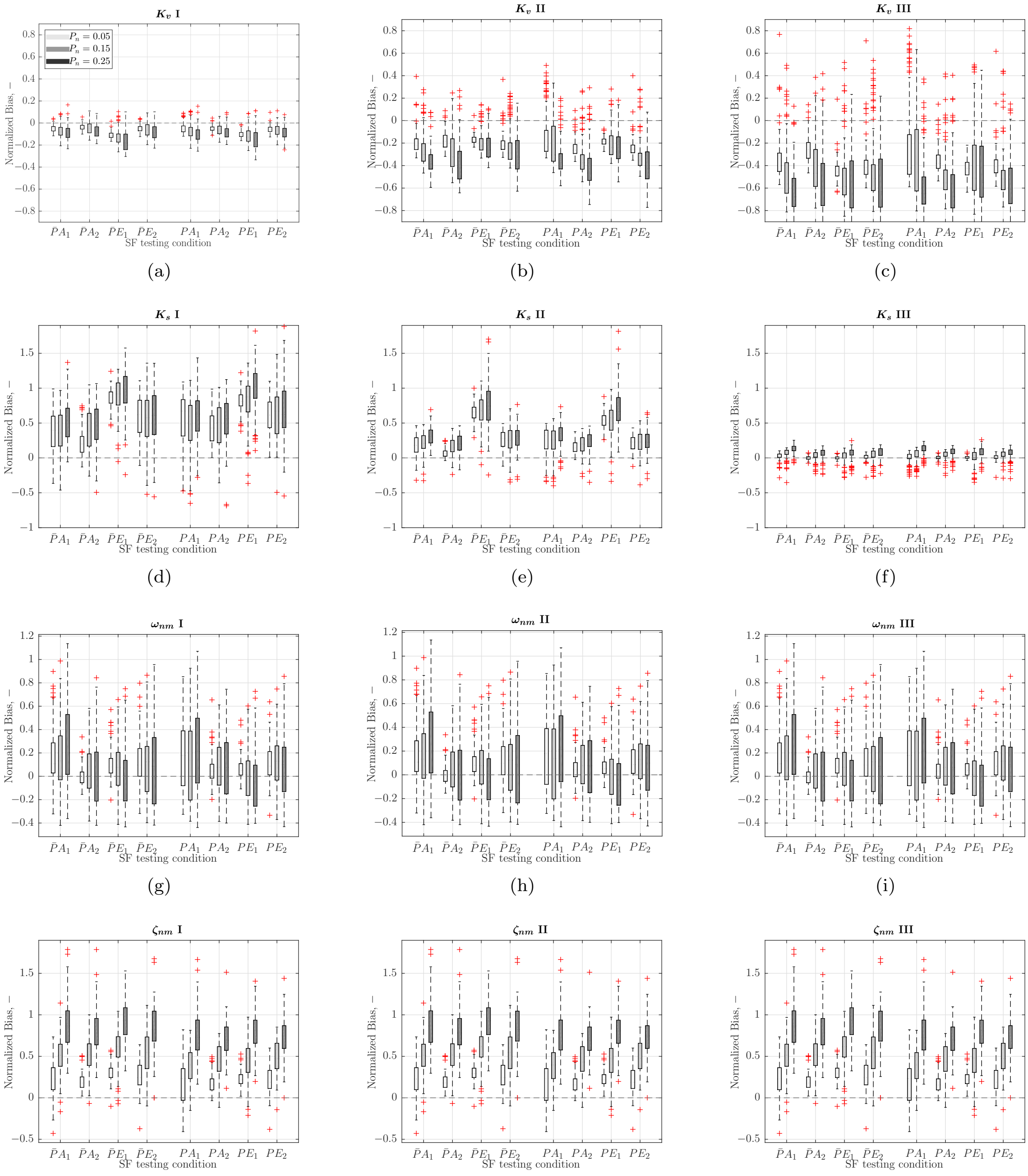
Through analysis of Figures 13(a)-(c), it is possible to observe a much more accurate K_v estimation for all testing conditions in Region I when compared with Region III. In fact, the absolute value of the normalized bias for Region I is rarely above 20%, even for $P_n = 0.25$. On the other hand, Region III K_v estimations seem to be worse, with some runs of $P_n = 0.25$ achieving a normalized bias as high as 80%. Throughout the different testing regions, a distinct tendency for undershooting the estimation of K_v is present, as evidenced by the median of the normalized biases of the different conditions. Finally, there is not much difference in K_v estimation between the considered SFs, apart from a slightly worse performance of μ_{E_1} in Regions I and II. Analysing these plots with the help of Figure 11, it may be concluded that the parameter K_v has relatively small influence in the estimated overall model quality.

In turn, the subsequent trio of sub-figures (Figures 13(d)-(f)) presents much more conclusive results regarding the fundamental differences in VAF witnessed in Figure 11. In fact, the K_s estimation using μ_{E_1} proves to be worse than with μ_{E_2}, especially concerning Regions I and II. This result combined with Figure 11 indicates that, in the considered framework, the lead term K_s is more important than K_v in obtaining good-fitting models. There is not a significant difference in estimation bias between different human operator remnant intensities, and the overall bias reduces in Region III, evidencing very good performance in K_s estimation for this particular region.

Figures 13(g)-(i) concern ω_{nm}, which is supposed to be non time-varying. Accordingly, the normalized bias results are the same throughout the different regions. A very high variance is observed, with a tendency to overshoot the real value. This unusually high variance contributes to a lack of ω_{nm} estimation precision, and gets worse as P_n increases. No significant differences between SFs or H_p conditions are detected.

For ζ_{nm}, Figures 13(j)-(l) show no bias variation between regions, as expected. However, much like ω_{nm}, the normalized bias variance is still very high. As P_n increases, the estimation gets gradually worse, and there are no significant differences between SFs or H_p conditions.

Figure 14 represents the normalized bias of the visual delay τ_v. Since its estimation was performed off-the-loop (Section III-G), the results are not presented separated in regions, as the estimation is valid



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 13. Bias in parameter estimation, for different regions and $P_n = 0.05$, 0.15 and 0.25.

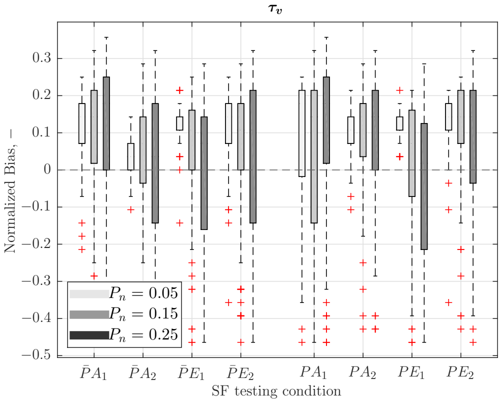


Figure 14. Bias in visual delay τ_v estimation.

for the entire simulation length. The estimations have high bias variances for any condition but seem to be generally more biased for the experimental Scheduling Functions. Furthermore, despite displaying high variances for $P_n = 0.15$ and 0.25, the experimental Scheduling Functions have a performance comparable to the analytical ones, for these P_n levels.

C. Sensitivity Analysis

THE results of the offline parameter sensitivity analysis are presented in Figure 15. This analysis was not performed for the visual delay parameter τ_v, since its estimation was not explicitly performed by the algorithm (see Section III-G). This analysis is intended to support and further explain the results obtained in the Scheduling Function tests of Section IV-B.

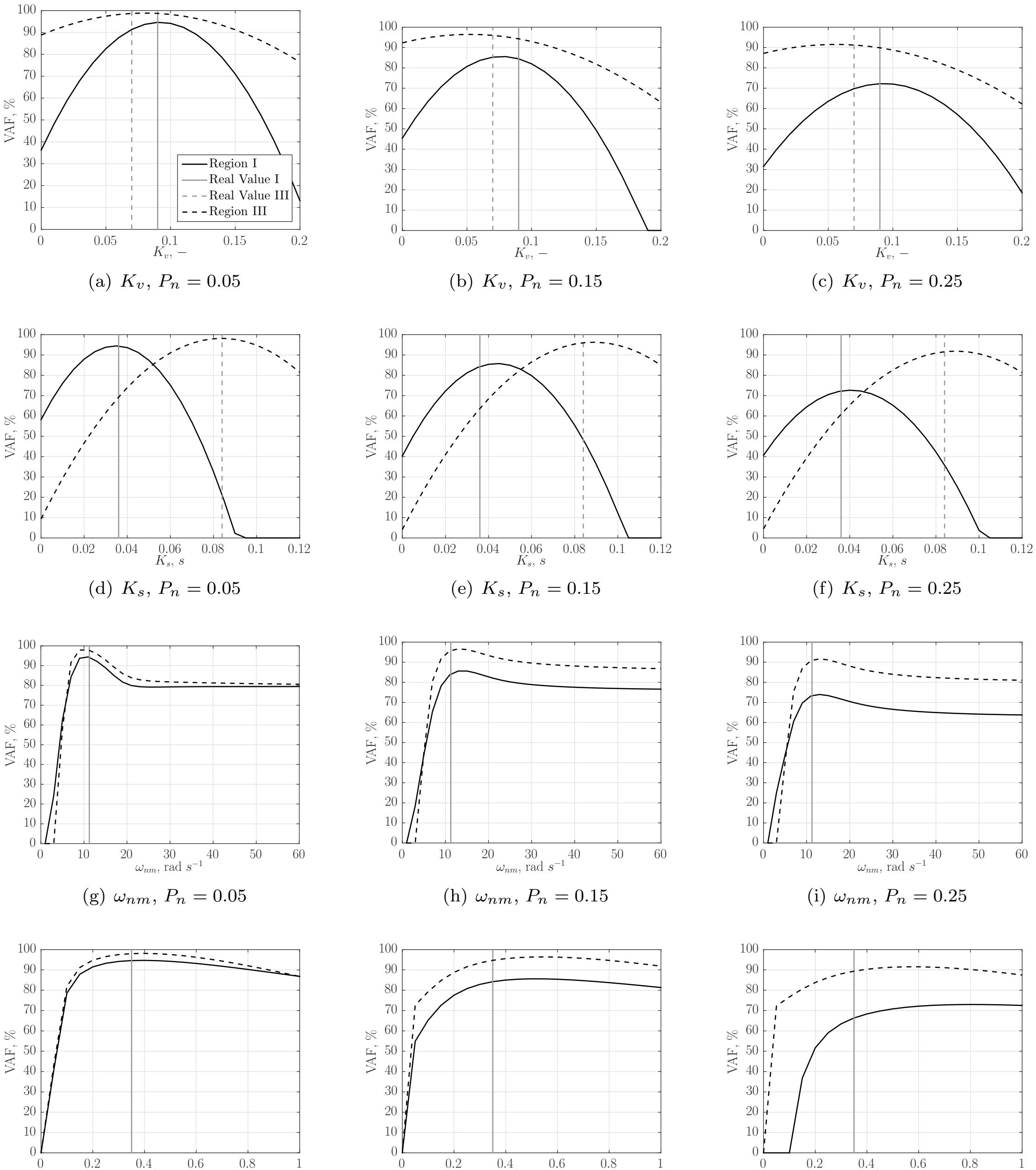
Most of the plots in Figure 15 show a convex parabola-like shape, indicating the presence of a single VAF maximum. This maximum should, in theory, be reached at the real value of the parameter being estimated.

Throughout Figure 15, it is possible to observe that the VAF values for Region III are consistently much higher than those for Region I. A direct comparison between the VAF of these two conditions can not be considered, as the human operator remnant power is time-invariant.

Figure 15(a) shows the VAF value is much more sensitive to K_v variations in Region I than in Region III, where a large offset in K_v leads to relatively small VAF drops. This result supports the large estimation bias observed for K_v in Region III (Figure 13(a)). The probable cause is the high-importance status the visual gain has in Region I human operator dynamics (where the human equalization is, excluding limitations, the visual gain itself) versus the low importance K_v acquires, when in Region III the human operator equalization becomes predominantly a lead term.[1] As the remnant power increases, the curves flatten and the bias is bound to increase, because a high VAF model is equally obtainable for a broader range of K_v.

Figures 15(d)-(f) showcase the performance of the alternative parametrization presented in Section III-D-1, where it is possible to observe less bias for Region III than for Region I, confirming the results from Figure 13(f).

The last two sets of plots deal with the non time-varying parameters of H_p, ω_{nm} and ζ_{nm}. In Figures 15(g)-(i), it is possible to observe a lack of VAF sensitivity for estimated ω_{nm} values above 20 rad s^{-1}: an expected result, since f_t does not have input power at those frequencies. Furthermore, as P_n increases, the well-defined VAF peak close to 10 rad s^{-1} erodes considerably. Finally, Figures 15(j)-(l) contain information about ζ_{nm}, where it is possible to observe an almost flat VAF curve throughout the neuromuscular damping range. This explains the high bias variances observed in Figure 13(g)-(l) for the estimated neuromuscular parameters.



(a) K_v, $P_n = 0.05$ (b) K_v, $P_n = 0.15$ (c) K_v, $P_n = 0.25$

(d) K_s, $P_n = 0.05$ (e) K_s, $P_n = 0.15$ (f) K_s, $P_n = 0.25$

(g) ω_{nm}, $P_n = 0.05$ (h) ω_{nm}, $P_n = 0.15$ (i) ω_{nm}, $P_n = 0.25$

(j) ζ_{nm}, $P_n = 0.05$ (k) ζ_{nm}, $P_n = 0.15$ (l) ζ_{nm}, $P_n = 0.25$

Figure 15. Changes in VAF induced by alternate variations of the model parameters.

V. Discussion

In this paper a Predictor-Based Subspace routine was proposed for time-varying human-operator identification in a single-loop pitch-tracking task with time-varying controlled element dynamics. The algorithm was implemented with the help of Van Wingerden's PBSID Toolbox.[19,26] To assess the usefulness of the PBSID algorithm in the identification of the human controller, data collected from offline Monte Carlo simulations of the control task performed experimentally in Ref. 8 were used to identify several LPV models. These were then converted to local LTI models and evaluated. This analysis involved three steps. Firstly, the tunable settings of the PBSID algorithm[d] were selected based on the comparison of the quality of the models obtained for different algorithm settings (Section IV-A). Secondly, using the previously selected PBSID settings, the human operator model was identified using a selection of four Scheduling Variables and two H_p conditions. These results were then analysed regarding model quality and parametric estimation bias, to evaluate the performance of the candidate experimental Scheduling Variables (Section IV-B). Finally, a sensitivity analysis was performed, where the estimated parameters of the identified model were varied one at a time. The subsequently induced variations in estimated model quality provided a backdrop for analysing the parametric estimation performance, by indicating the expected parameter estimation precision achievable with this framework (Section IV-C).

The PBSID Tuning test showed a positive VAF effect on the increase of the identified model order. However, the improvement is marginal and after sixth order, the balanced order reduction needed to reduce the identified model to second order and retrieve the Precision Pilot Model[1] parameters induces a sharp decrease in model quality. An increasing past window p was verified to only marginally increase the overall VAF of the models. A more refined test would provide more meaningful results regarding the precise effects of p on the model quality. Furthermore, the effects of the Future Window F were not considered, to decrease the computation time. A study on the effect of F in model quality would add value to the PBSID parameter tuning. The selected tuning of $N = 2$, $F = 2$ and $p = 120$ provided satisfactory results. When comparing these settings with the ones used in the PBSID identification example presented in Ref.19, it is possible to observe a considerable difference in the selected size of the Past Window. Figure 3 in Ref.19 reveals that the overall model quality for that specific application example starts sharply decreasing for $p > 10$, so an appropriate choice of p would be between five and ten samples. On the other hand, the results presented in Section IV-A of the present paper indicate no adverse effects in model quality on increasing p until 120 samples. It is worth noting, however, the identification data used in this paper contained 10000 samples, while in Ref. 19 only 2000 samples were used.

Unfortunately, τ_v was pre-estimated and its estimation was not part of the main identification process. While the results obtained were satisfactory, this method does not allow for the estimation of a time-varying human operator visual delay.

The Scheduling Function tests revealed the potential of $\ddot{u}$ to be used as an experimental Scheduling Function, by carefully filtering the high-frequency oscillations. A phase-shifting filtering method, like RMS, was shown to be especially problematic in Region II, where the dynamics transition occurs. The zero-phase filtering of $\ddot{u}$ (μ_{E_2}) presented better VAF results, in both the full simulation and Region II, yielding estimated models with very close VAF values to the ones generated by the analytical sigmoid-based SFs. However, both experimental SFs fail to accurately capture the perturbation introduced for condition P, as evidenced by the decrease in SF correlation for this condition (Figure 12).

The PBSID algorithm itself is relatively consistent, its high-order LPV discrete state-space model estimates showing high VAF with low variance. In fact, if an appropriate Scheduling Function can be chosen, then the identified LPV model is of high quality. The usage of $N = 2$ eliminates the need of performing a balanced order reduction to second order for parametric retrieval, which would introduce a factor of error and uncertainty. Notably, the bias and variance of K_v in Region I and K_s in Region III are much lower than in any other combination of parameters and regions, as supported by the results of the Sensitivity Analysis. The estimation of the neuromuscular parameters was found to be inconsistent, a result also supported by the Sensitivity Analysis.

Still, the parameter estimation bias is generally higher than that obtained in Ref 8. Furthermore, a high number of parameter estimates turned out to be outliers, which shows the high-sensibility nature of the algorithm to different noise realizations. A possible solution to this problem might be averaging the identification data of the Monte Carlo runs, so that the human operator remnant is diluted and reduced.

[d] System order N, Past window p and Future Window F.

Future work in the application of the LPV framework for the identification of the human operator in tracking tasks may be developed. Experimental data testing using the time-varying controlled element of Ref. 8 and human test subjects to validate μ_{E_2} as a suitable SF for LPV identification is a natural continuation to the work developed in this paper. Within the PBSID algorithm, a thorough analysis on the effects of the p and F in model estimation quality would add a more precise understanding on the optimal PBSID settings for the specific application presented in this paper. The key aspect of the LPV framework lies in the Scheduling Variable. Therefore, a research on suitable candidate Scheduling Variables to be used for time-varying human operator identification in tracking tasks is of high importance. An interesting research topic also lies on the possibility of using multiple Scheduling Variables for LPV identification, and their effects on model quality and parameter estimation. The LPV framework was proved promising for modelling time-varying human operator behaviour, so new LPV identification methods (using Orthonormal Basis Functions,[15] for example) may continue to be developed and tested in the future. Finally, a new, unambiguous human operator parametrization, more suitable for LPV identification may be explored in the future.

VI. Conclusion

This paper had the goals of assessing the viability of using the LPV framework and the PBSID algorithm to solve the time-varying human in-the-loop identification problem, and to find a suitable experimental Scheduling Variable that can be used for the LPV identification. To accomplish this, an offline recreation of the experiment in Ref 8 was used as a testing bed. Monte Carlo runs with representative human operator remnant noise realizations were performed, and the human operator dynamics was identified as an LPV model using the PBSID algorithm, with different testing conditions and Scheduling Functions. The identified models were compared using VAF and the relative bias of the model parameters. The PBSID algorithm was found to have difficulties in estimating the high visual delay of the human operator. When delay compensation was used, the algorithm yielded very high VAF models, of high fitting quality. The comparison between analytical and experimental Scheduling Functions yielded encouraging results regarding the Scheduling Function obtained from zero-phase filtering the second derivative of the human operator output signal (μ_{E_2}). In fact, the models identified with this experimental SF produced VAF values very close to the models identified with the analytical Scheduling Functions for condition $\bar{P}$ (no perturbation in H_p dynamics). Furthermore, for condition P (perturbation in H_p dynamics), the μ_{E_2} models had slightly higher VAF values than the models obtained through the reference analytical SF taken from Ref. 8, over which represents an improvement.

References

[1] McRuer, D. T. and Jex, H. R., "A Review of Quasi-Linear Pilot Models," *IEEE Transactions on Human Factors in Electronics*, Vol. HFE-8, No. 3, sep 1967.

[2] Duane T. McRuer, R. E. and Moore, G. P., "A Neuromuscular Actuation System Model," *IEEE Transactions on Man-Machine Systems*, Vol. MMS-9, No. 3, sep 1968.

[3] Pool, D. M., *Pilot Control in Compensatory Roll Axis Tracking Tasks*, Master's thesis, Delft University of Technology, 2007.

[4] van der El, K., Pool, D. M., Damveld, H. J., van Paassen, M. M., and Mulder, M., "An Empirical Human Controller Model for Preview Tracking Tasks," *IEEE Transactions on Cybernetics*, 2015.

[5] Nieuwenhuizen, F. M., Zaal, P. M. T., Mulder, M., van Paassen, M. M., and Mulder, J. A., "Modeling Human Multichannel Perception and Control Using Linear Time-Invariant Models," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, July-August 2008, pp. 999–1013.

[6] Roggenkämper, N., Pool, D. M., Drop, F. M., van Paassen, M. M., and Mulder, M., "Objective ARX Model Order Selection for Multi-Channel Human Operator Identification," *Proceedings of the AIAA Modeling and Simulation Technologies Conference, Washington, D.C.*, No. AIAA-2016-4299, June 2016.

[7] Zaal, P. M. T. and Pool, D. M., "Multimodal Pilot Behavior in Multi-Axis Tracking Tasks with Time-Varying Motion Cueing Gains," *Proceedings of the AIAA Modeling and Simulation Technologies Conference, National Harbor (MD)*, No. AIAA-2014-0810, Jan. 2014.

[8] Zaal, P. M. T., "Manual Control Adaptation to Changing Vehicle Dynamics in Roll-Pitch Control Tasks," *Journal of Guidance, Control and Dynamics*, Vol. 39, No. 5, 2016.

[9] 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," *Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics, San Diego (CA)*, Oct. 2014, pp. 3573–3578.

[10] Kers, M., *Maximum Likelihood Estimation of Linear Time-Varying Pilot-Vehicle System Parameters*, Master's thesis, Delft University of Technology, Aug. 2012.

[11] Thompson, P. M., Klyde, D. H., and Brenner, M. J., "Wavelet-Based Time-Varying Human Operator Models," *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit, Montreal (CA)*, No. AIAA-2001-4009, 2001.

[12] Zaal, P. M. T. and Sweet, B. T., "Estimation of Time-Varying Pilot Model Parameters," *Proceedings of the AIAA Modeling and Simulation Technologies Conference, Portland, Oregon, Aug. 8-11*, No. AIAA-2011-6474, 2011.

[13] Giarr, L., Bauso, D., Falugi, P., and Bamieh, B., "{LPV} model identification for gain scheduling control: An application to rotating stall and surge control problem," *Control Engineering Practice*, Vol. 14, No. 4, 2006, pp. 351 – 361.

[14] van Wingerden, J. W., *Control of Wind Turbines with Smart Rotors: Proof of Concept & LPV Subspace Identifcation*, Ph.D. thesis, Technische Universiteit Delft, 2008.

[15] Tóth, R., "Modeling and identification of linear parameter-varying systems," 2010.

[16] Lovera, G. M. M. and Laroche, E., "Identification of a Flexible Robot Manipulator Using a Linear Parameter-Varying Descriptor State-Space Structure," *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011.

[17] Tóth, R., Laurain, V., Gilson, M., and Garnier, H., "Instrumental variable scheme for closed-loop LPV model identification," *Automatica*, Vol. 48, 2012, pp. 2314–2320.

[18] Chiuso, A., "The role of vector autoregressive modeling in predictor-based subspace identification," *Automatica*, Vol. 43, No. 6, 2007, pp. 1034 – 1048.

[19] van Wingerden, J. and Verhaegen, M., "Subspace identification of Bilinear and LPV systems for open and closed loop data," *Automatica*, Vol. 45, No. 2, 2009, pp. 372–381.

[20] Pronker, A., *Estimating Time-Varying Neuromuscular Admitance During Steering Tasks (Preliminary MSc. Thesis)*, Master's thesis, Faculty of Aerospace Engineering - Delft University of Technology, 2015.

[21] Rugh, W., *Linear system theory*, Upper Saddle River, 1996.

[22] Varga, A., "Balancing-Free Square-Root Algorithm for Computing Singular Perturbation Approximations," *Proc. of 30th IEEE CDC*, Brighton, UK, 1991, pp. 1062–1065.

[23] J.A. Mulder, J.C. van der Vaart, W. v. S. . M. M., *Lecture Notes AE4304 - Aircraft Responses to Atmospheric Turbulence*, Delft University of Technology - Faculty of Aerospace Engineering, 2015.

[24] IEEE, "Standard on Transitions, Pulses, and Related Waveforms," 2003.

[25] Oppenheim, A. V., Schafer, R. W., and Buck, J. R., *Discrete-Time Signal Processing*, Upper Saddle River, NJ, 1999.

[26] "PBSID Toolbox," http://www.dcsc.tudelft.nl/~jwvanwingerden/pbsid/pbsidtoolbox_product_page.html.

II

Appendixes

A

Scheduling Function E_1

An LPV system depends on Scheduling Functions that are explicitly time dependent and steer the system's dynamics by weighing the system matrices.

In the case of the LPV identification setting of a human controller controlling simple pitch dynamics, the choice of Scheduling Variable is not as trivial as in natural LPV systems. Indeed, the human operator is a non-linear, time varying system, whose conversion to an LPV framework depends on the use of a SF that is both measurable and coupled with the varying system dynamics. In this appendix, the retrieval and subsequent treatment of the SF E_1 used in testing is explored.

A.1. Human operator output and its derivatives

The output of the human controller, u, was verified be a good candidate for SF: it is measurable (through the deflection of the joystick element) and, as supported by Figure A.1, presents a good coupling to the overall changing dynamics, in the form of an increased activity in u, as the workload increases.

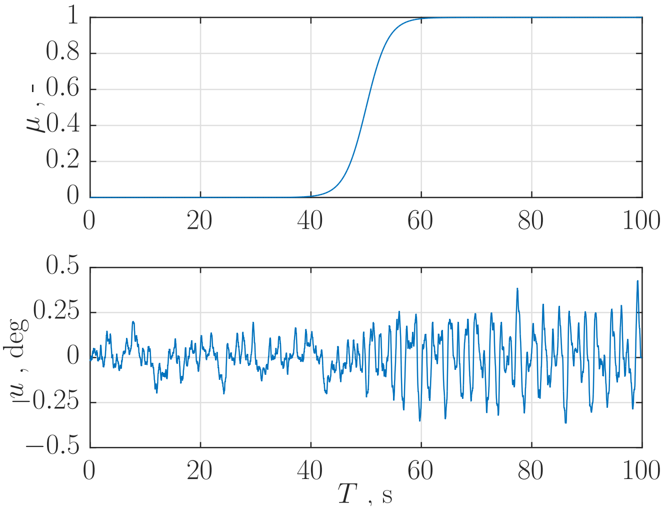


Figure A.1: Comparison between the sigmoid baseline SF μ and the human operator output (u).

The output u, however, can not be directly used as a Scheduling Function, because the changes in dynamics it indicates need to be represented in the form of amplitude variations. The controlled system double integrator forces the human operator to adopt a lead term in its own dynamics, which is expected to generate more aggressive changes in u, as the human operator tries to predict the input signal. These changes are best picked up in the amplitude of $\ddot{u}$. We can observe the coupling between $\ddot{u}$ and μ in Figure A.2.

The difference between the $\mu = 0$ dynamics and the $\mu = 1$ dynamics is now much more noticeable in $\ddot{u}$ than in u.

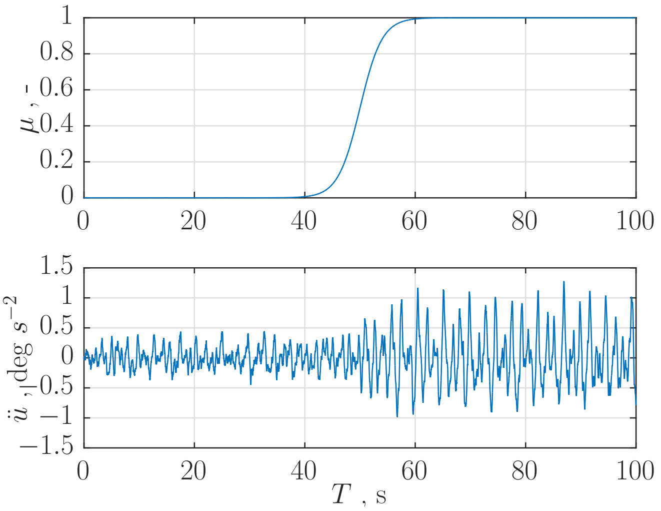


Figure A.2: Comparison between the sigmoid baseline SF μ and the double time derivative of the human operator output ($\ddot{u}$).

The retrieval of $\ddot{u}$ from the offline simulation was performed using an approximation: a derivative term with a low-pass filter with a low cutoff frequency, so that the noise is filtered out, to avoid its accidental differentiation:

$$\frac{\dot{u}(s)}{u(s)} = \frac{s}{0.5s+1} \tag{A.1}$$

A.2. Pre-processing

Before $\ddot{u}$ can be used as an experimental SF for the PBSID testing, it needs to be pre-processed, in order to eliminate the noisy oscillations and hopefully extract a clean signal trend, similar in character to the reference μ.

A.2.1. Step 1

The signal has currently negative and positive values, with an average close to zero. Since the reference SF μ is always positive, E_1 has to follow the same trend, to provide a fair comparison. There are two main possibilities: take the absolute value of $\ddot{u}$, or exponentiate it with an even power.

The absolute value of $\ddot{u}$ was then taken: $E_{1_{S1}} = |\ddot{u}|$. While an even exponentiation would yield a further accentuated difference between the two main human operator dynamics in play, it would also amplify any disturbance in the signal.

Figure A.3 shows the resulting $E_{1_{S1}}$ signal. Notice the trending is still present, only the average became positive.

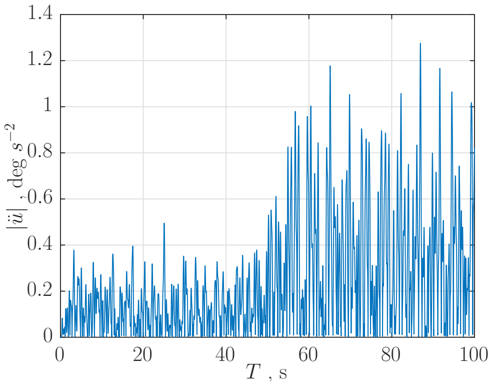


Figure A.3: Time trace of $E_{1_{S1}} = |\ddot{u}|$.

A.2.2. Step 2

Now that $E_{1_{S1}} = |\ddot{u}|$ has been obtained, a filtering process can be applied to the signal. The main goal is to remove the high frequency oscillations of the signal and extract the trend component of $E_{1_{S1}}$. This filtering was performed by defining a moving window in which the Root Mean Square (RMS) of the corresponding portion of signal is calculated. [5]

Different window sizes and number of successive filterings were tested. The obtained correlation values between the reference SF μ and the post-filtering signal $E_{1_{S2}}$ are compiled in table A.1 and showed in Figure A.4. These values were obtained for no human operator remnant ($P_n = 0$) and for a wide range of window sizes and successive RMS filterings.

		Number of successive RMS filterings						
		1	2	3	4	5	6	7
RMS filter window size (samples)	4	0.6957	0.7808	0.8309	0.8643	0.8883	0.9064	0.9207
	6	0.8352	0.9261	0.9598	0.9747	0.9819	0.9856	0.9876
	8	0.9169	0.9790	0.9916	0.9951	0.9960	0.9959	0.9952
	10	0.9572	0.9939	0.9976	0.9973	0.9955	0.9929	0.9894
	12	0.9764	0.9974	0.9969	0.9937	0.9888	0.9826	0.9751
	14	0.9847	0.9963	0.9926	0.9859	0.9770	0.9662	0.9539

Table A.1: Correlation between μ and $E_{1_{S2}}$, $P_n = 0$.

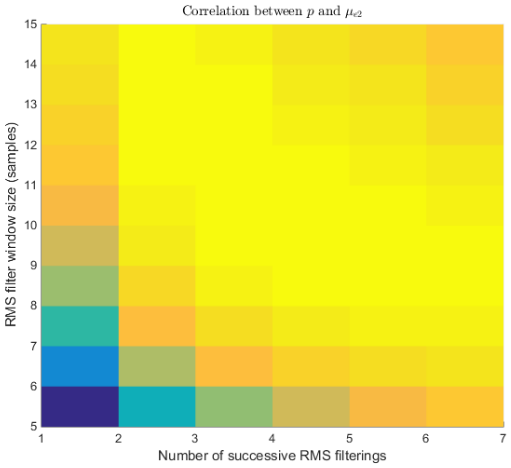


Figure A.4: Correlation map between μ and $E_{1_{S2}}$, $P_n = 0$.

From Figure A.4, it is observable that large correlation values between μ and $E_{1_{S2}}$ are obtained for a wide range of window sizes and number of successive filterings. It is noticeable, nevertheless, that as the number of successive filterings increases, the optimal window size decreases. Although a high correlation is good, meaning both SV's are compatible, it doesn't necessarily guarantee a good performance when identifying the LPV system: let us consider the filter settings which yielded the highest correlation value: window size of 10 samples and 3 successive RMS filterings (Table A.1). The resulting $E_{1_{S2}}$ is presented in Figure A.5.

It is possible to observe some high-frequency oscillations of relatively low amplitude. For higher values of P_n, the oscillations could be severe and negatively impact the convergence of the algorithm. This possibility will be investigated in Section A.3.

A.2.3. Step 3

A final step of translating and scaling is required. This ensures the final SF stays approximately normalized within the interval $[0, 1]$, to facilitate the comparison between different SFs. This was done by first subtracting the initial value of $E_{1_{S2}}$ to the entire scheduling array, followed by dividing the scheduling array with the average value of $E_{1_{S2}}$ in the last forty seconds of simulation. This results in the final E_1 SF, presented in Figure A.6.

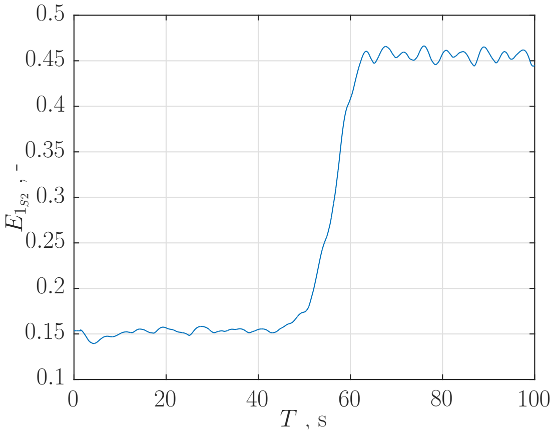


Figure A.5: $E_{1_{S2}}$, obtained filtering $E_{1_{S1}}$ with 3 successive filterings of window size 10, $P_n = 0$.

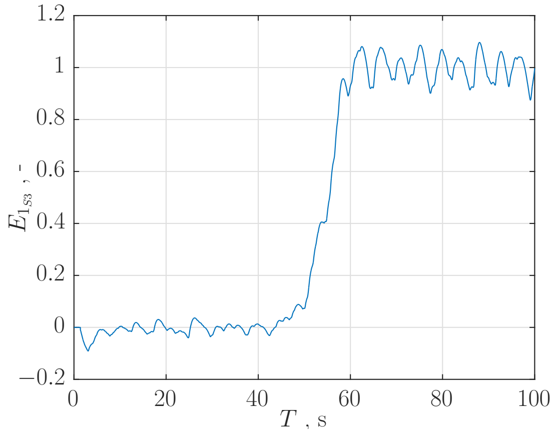


Figure A.6: E_1, obtained by translating and scaling $E_{1_{S2}}$.

A.3. The effect of the human operator remnant

When considering high levels of human operator remnant, an increase in noise on the $\ddot{u}$ signal can be expected, which affect the filtering procedure for obtaining the E_1 Scheduling Function.

For an arbitrary realization $P_n = 0.25$, and using the same settings that yielded the highest correlation for $P_n = 0$, the obtained E_1 SF is represented in Figure A.7.

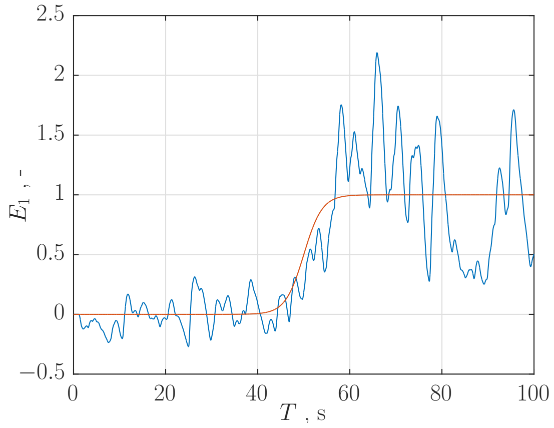


Figure A.7: E_1, obtained by 3 RMS filterings with a 10-sample window size, $P_n = 0.25$.

In this case, the high remnant level ($P_n = 0.25$) induces high-frequency and high-amplitude oscillations that are not sufficiently filtered by the optimal filter settings for $P_n = 0$. By increasing the number of successive filterings, the signal becomes more smooth, but a delay is also introduced:

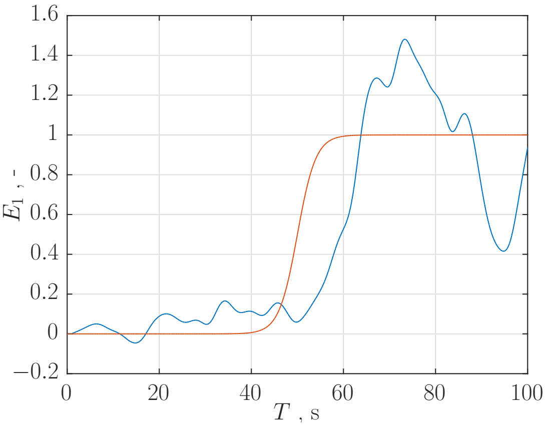


Figure A.8: E_1, obtained by 20 RMS filterings with a 10-sample window size, $P_n = 0.25$.

The higher the number of successive filterings, the higher the delay introduced in E_1, but also the smoother the signal. This limitation of the RMS filtering prompted the search for a different type of filtering in Appendix B.

B

Scheduling Function E_2

Following from the findings of Appendix A, the need for a better filtering of the potential SF that would not introduce phase-shift and work well under different P_n levels became clear. As such, a simple linear optimization of a zero-phase digital Butterworth filtering was chosen to process $\ddot{u}$ and obtain E_2.

B.1. Zero-phase filtering

The zero-phase filtering technique applies a specified filter to the signal twice, once in the forward direction and once in the opposite direction. This operation guarantees that the phase distortion caused by the first filtering is counteract by the second filtering, yielding zero-phase distortion. [13]

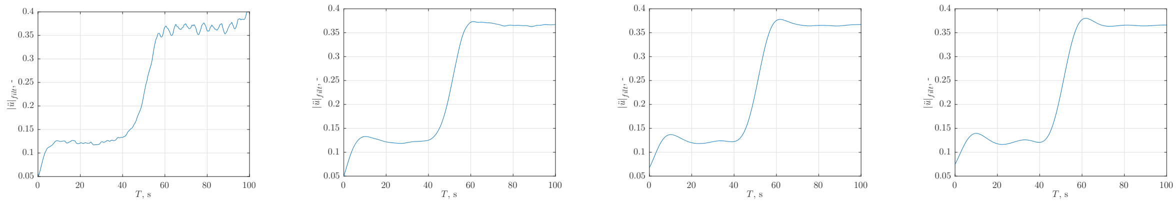
B.2. Pre-processing

The pre-filtering processing starts, similarly to E_1, with the computation of $|\ddot{u}|$ (Figure A.3).

B.2.1. Filter design

The chosen filter to treat the data was a low-pass Butterworth filter. This filter was chosen because it allows for a flat frequency response throughout the low frequency range. The filter is thus expected to eliminate or highly attenuate the oscillations of frequency higher than the specified cut-off frequency. Since the Butterworth filter is applied to $|\ddot{u}|$ using zero-phase digital filtering, the resulting signal is also expected to conserve its phase and, consequently, not suffer from the RMS-induced delay of E_1. Figure B.2 represents a typical frequency response for a generic low-pass Butterworth digital filter.

By specifying the order and the half-power frequency, the transfer function of the filter may be obtained. A second-order filter was selected, as its performance was better than a first-order filter, and less oscillating than higher-order filters, as can be observed in Figures B.1.



(a) First-order filter.

(b) Second-order filter.

(c) Third-order filter.

(d) Fourth-order filter.

Figure B.1: Effects of different order zero-phase Butterworth filtering of $|\ddot{u}|$, for the same specified half-power frequency. $P_n = 0$.

The half-power frequency, however, was chosen to be iteratively optimized for each individual simulation run, in a bid to attenuate the effects of noise-realization variability on the filtered $|\ddot{u}|$.

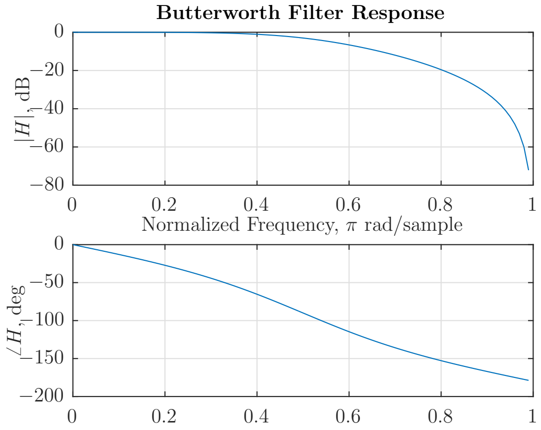


Figure B.2: Frequency response of a second order Butterworth filter, with cut-off frequency $\omega_c = 25$ Hz and sampling frequency $f_s = 100$ Hz, yielding a normalized cut-off frequency of 0.5π rad/sample.

B.2.2. Optimization process

To optimize the half-power frequency, a simple quadratic error cost function was defined:

$$J = \sum (|\ddot{u}|_{\text{filt}} - \mu)^2_{\text{Region II}} \tag{B.1}$$

where $|\ddot{u}_{\text{filt}}|$ is the filtered $\ddot{u}$ signal with the candidate half-power frequency and μ is the analytical SF with respect to the considered condition. Obviously, a better approximation between the two Scheduling Functions results in a lower cost function value J, and the of half-power frequency which yields the minimum cost is chosen to filter the signal. This optimization is performed with respect to Region II, to specifically penalize a bad fit in this region, as this is where the change in dynamics happen and where the perturbations have significant effect in testing condition P.

The full optimization process makes use of a Nelder-Mead Simplex algorithm [6] and has the following steps:

- For the first iteration, initialize the normalized half-power frequency, ω_{hp_0};
- Check if $\omega_{hp} \leq 0$. If true, assign it a low positive value (1×10^{-6});
- Design a low-pass second-order Butterworth digital filter with the specified ω_{hp};
- Process the $|\ddot{u}_{\text{filt}}|$ signal with a zero-phase digital filtering, using the previously obtained Butterworth filter (use `Matlab` function `filtfilt`, for example);
- Translate and scale the obtained $|\ddot{u}|_{\text{filt}}$, as in Section A.2.3;
- Obtain J from Equation B.1;
- Search for new ω_{hp} using the Simplex method;
- Repeat until improvements in J are marginal or the limit of twenty iterations has been reached.

The final value of ω_{hp}, deemed the optimal, is then used to filter $|\ddot{u}|$, which is then translated and scaled to obtain E_2. Figure B.3 represents E_2, obtained after the aforementioned process, for $P_n = 0$. For comparison, μ is also given. It is possible to observe that due to the zero-phase filtering, no delay is observed.

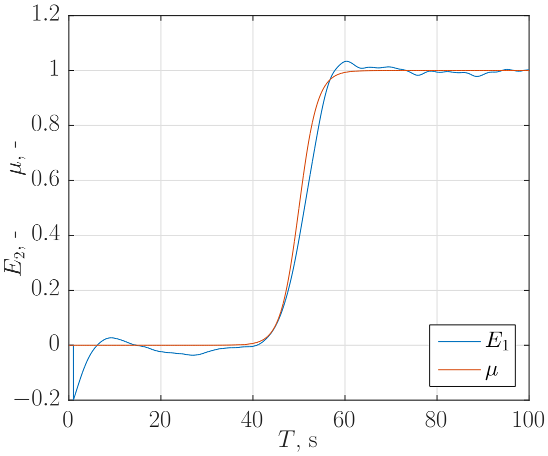
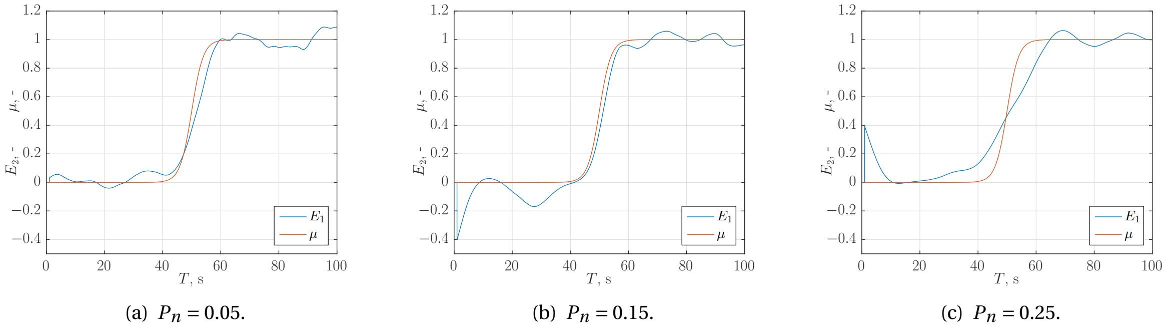


Figure B.3: Comparison between E_2 and μ, for $P_n = 0$.

B.3. The effect of the human operator remnant

Figures B.4 represent the obtained Scheduling Functions E_1, for aleatory realizations of different remnant levels, compared with μ.

Comparatively to the results presented in Appendix A, the results obtained for E_2, with the optimized zero-phase digital filtering appear to be more robust against high human-operator noise. Furthermore, there is no delay in the transition region for $P_n = 0.05$ and $P_n = 0.15$, although E_2 is not able to tightly follow μ during the transition for $P_n = 0.25$. The final value of E_2 seems to be quite tightly around μ for all conditions, with some low-amplitude oscillations. However, there is a ten-second transient of relatively high-amplitude at the beginning of the simulations. This is probably caused by the fact that the initial condition of $|\ddot{u}|$ is zero. The filter matches this initial condition, causing the filtered signal to transition from zero to its initial DC component in the first ten seconds. After the translation and scaling operations are complete, this transient is observed to be modified as well.



(a) $P_n = 0.05$.

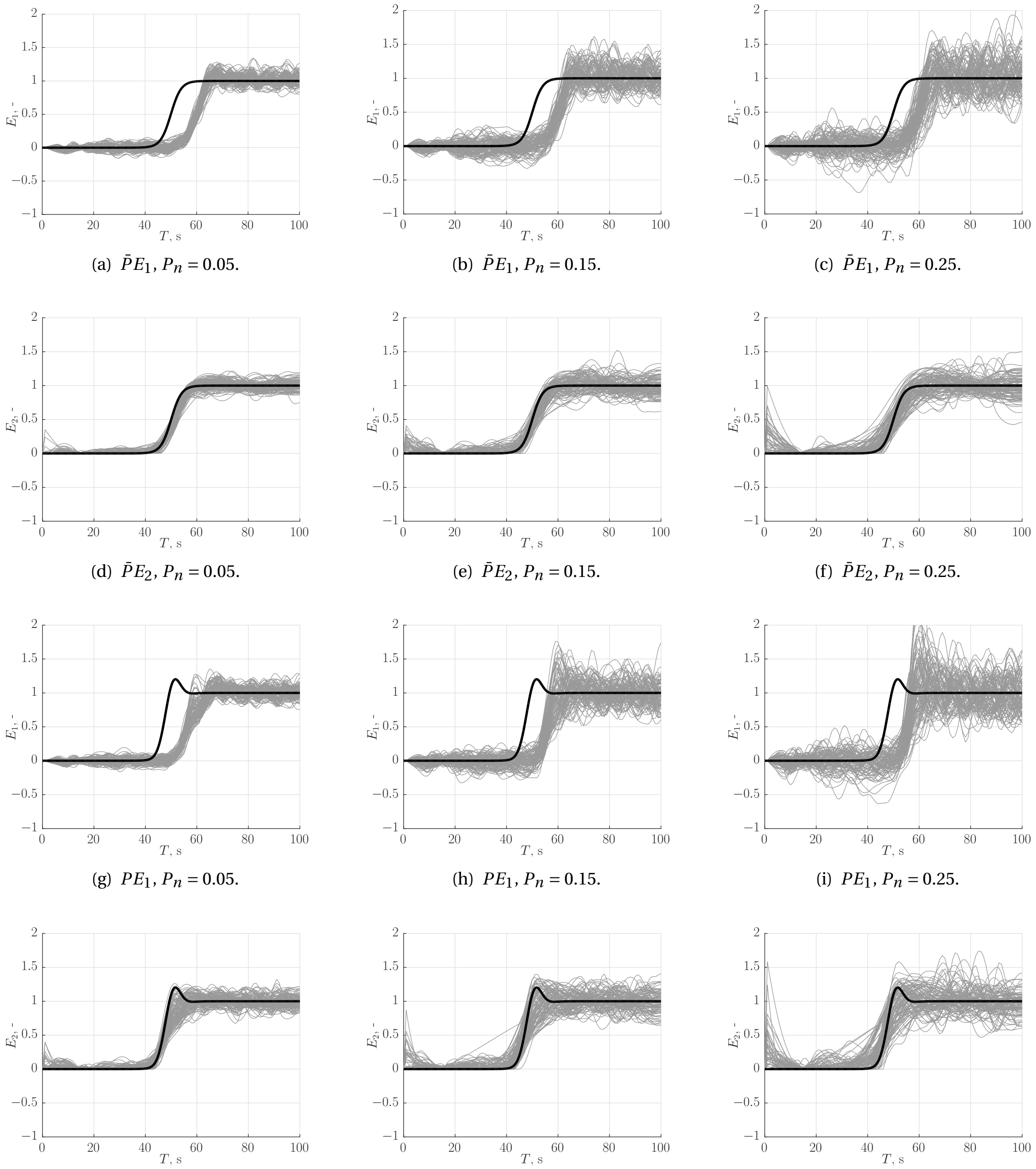
(b) $P_n = 0.15$.

(c) $P_n = 0.25$.

Figure B.4: Effects of different levels of human operator remnant in the SF E_2, for a single realization.

C

Monte Carlo Results: Scheduling Functions



(a) $\bar{P}E_1$, $P_n = 0.05$.

(b) $\bar{P}E_1$, $P_n = 0.15$.

(c) $\bar{P}E_1$, $P_n = 0.25$.

(d) $\bar{P}E_2$, $P_n = 0.05$.

(e) $\bar{P}E_2$, $P_n = 0.15$.

(f) $\bar{P}E_2$, $P_n = 0.25$.

(g) PE_1, $P_n = 0.05$.

(h) PE_1, $P_n = 0.15$.

(i) PE_1, $P_n = 0.25$.

(j) PE_2, $P_n = 0.05$.

(k) PE_2, $P_n = 0.15$.

(l) PE_2, $P_n = 0.25$.

Figure C.1: Experimental SF realizations obtained from 100 Monte Carlo runs, for each condition. Comparison with the corresponding analytical SF.

D

Monte Carlo Results: Parameter Estimation

The considered testing conditions and Scheduling Functions are once again presented in Table D.1.

Table D.1: Scheduling Function testing conditions.

H_p	Scheduling Function			
Condition	Analytical		Experimental	
	μ_{A_1}	μ_{A_2}	μ_{E_1}	μ_{E_2}
$\bar{P}$	$\bar{P}_{A_1}$	$\bar{P}_{A_2}$	$\bar{P}_{E_1}$	$\bar{P}_{E_2}$
P	P_{A_1}	P_{A_2}	P_{E_1}	P_{E_2}

The regions of assessment are defined as in Figure D.1. The results presented in this appendix are for Region I and Region III.

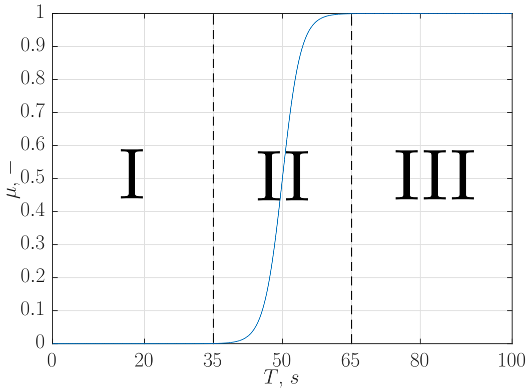
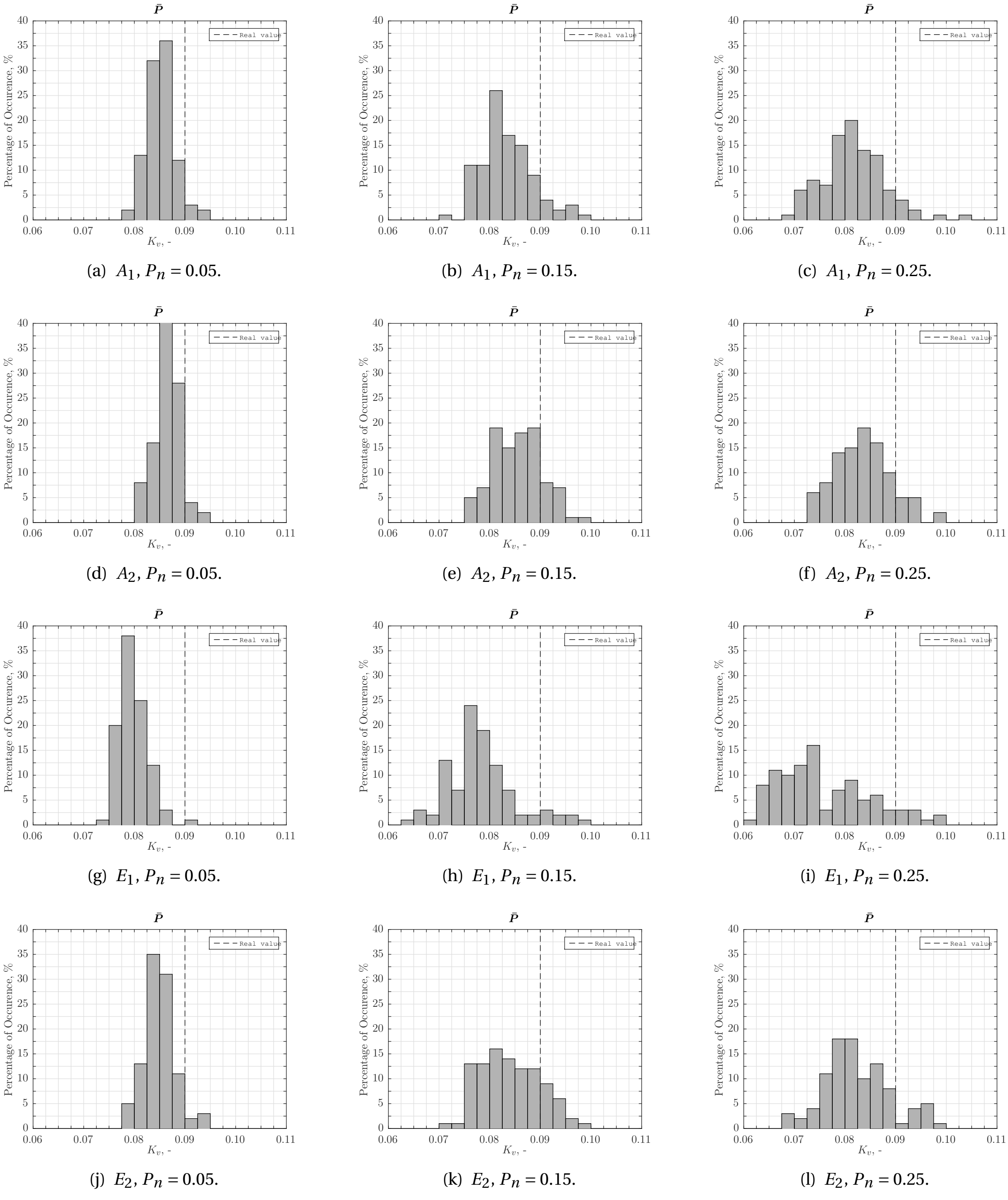


Figure D.1: The three regions of testing assessment, with μ represented.

D.1. K_v

D.1.1. Region I

Condition $\bar{P}$



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

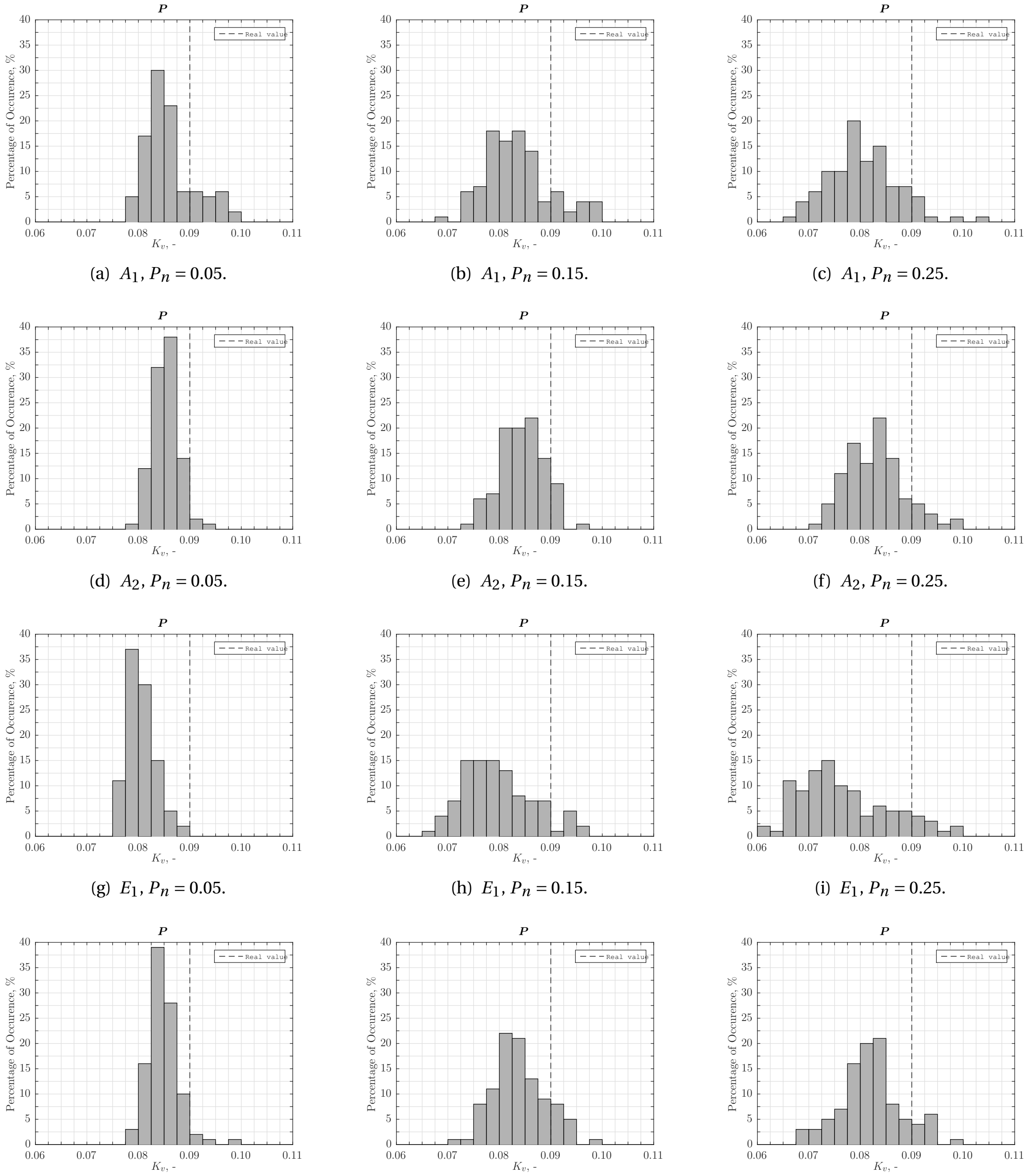
(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.2: Distribution of K_v estimations in 100 Monte Carlo simulations for different SFs. Condition $\bar{P}$, Region I.

Condition P



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

(j) E_2, $P_n = 0.05$.

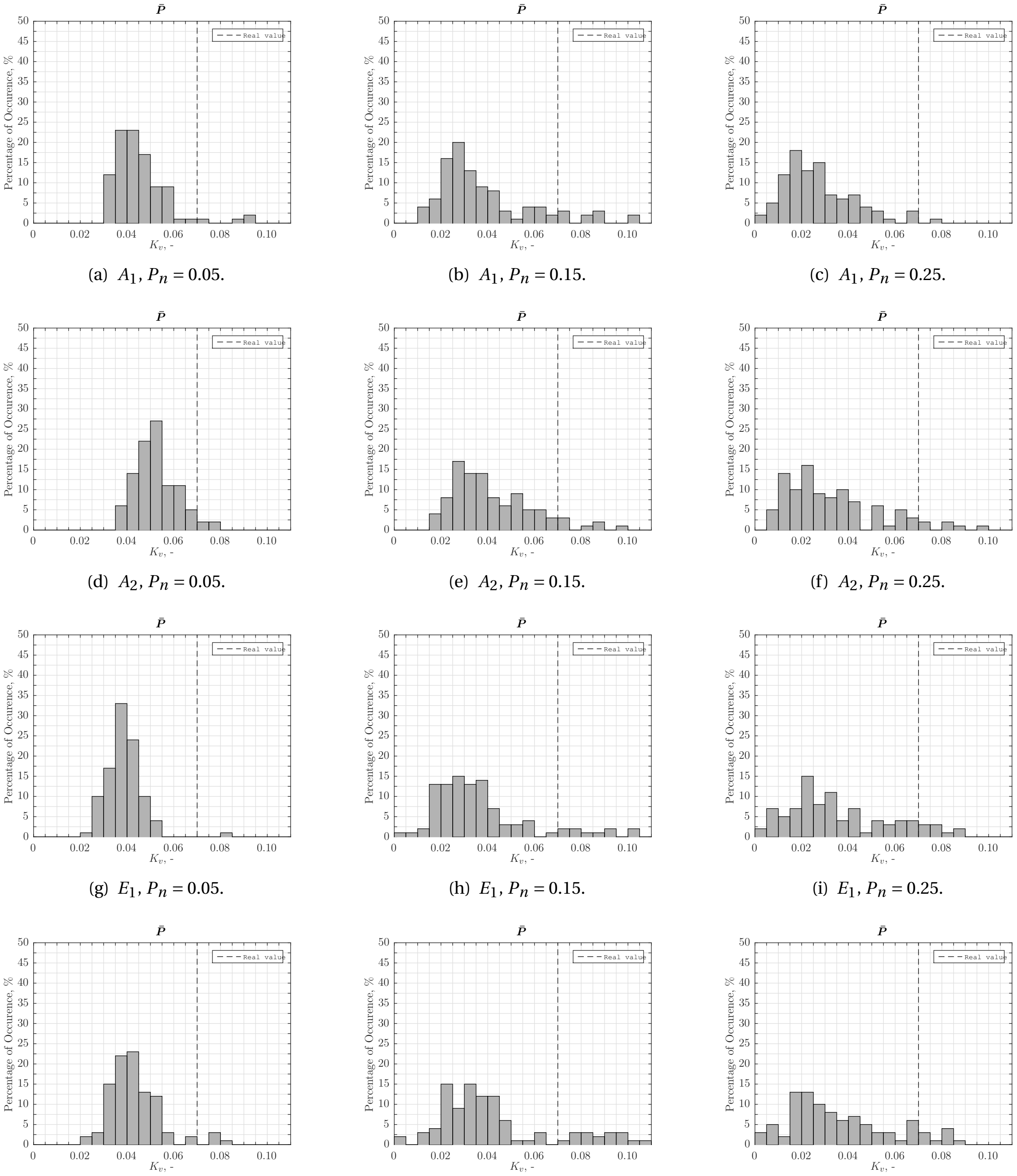
(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.3: Distribution of K_v estimations in 100 Monte Carlo simulations for different SFs. Condition P, Region I.

D.1.2. Region III

Condition $\bar{P}$



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

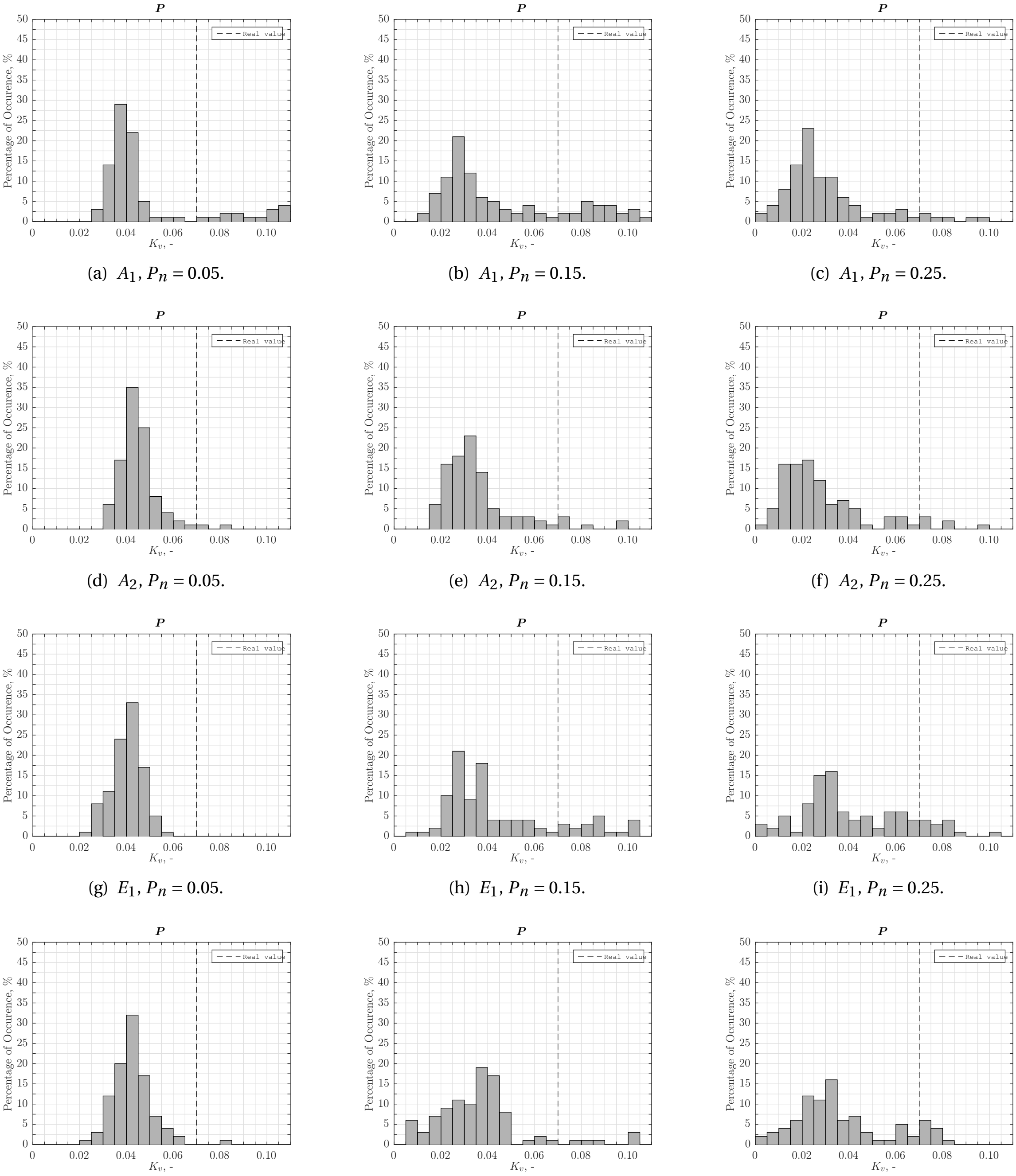
(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.4: Distribution of K_v estimations in 100 Monte Carlo simulations for different SFs. Condition $\bar{P}$, Region III.

Condition P



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

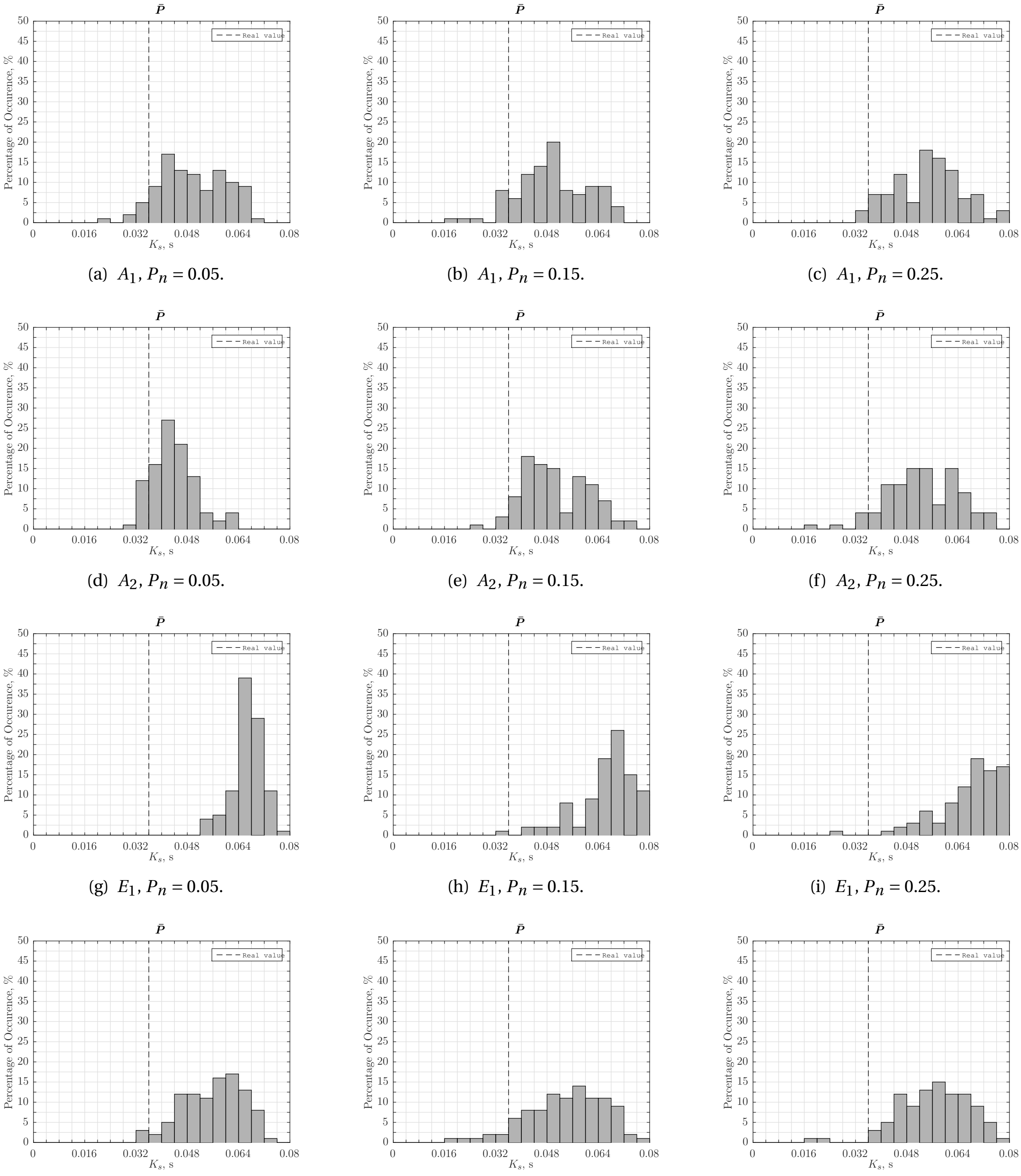
(l) E_2, $P_n = 0.25$.

Figure D.5: Distribution of K_v estimations in 100 Monte Carlo simulations for different SFs. Condition P, Region III.

D.2. K_s

D.2.1. Region I

Condition $\bar{P}$



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

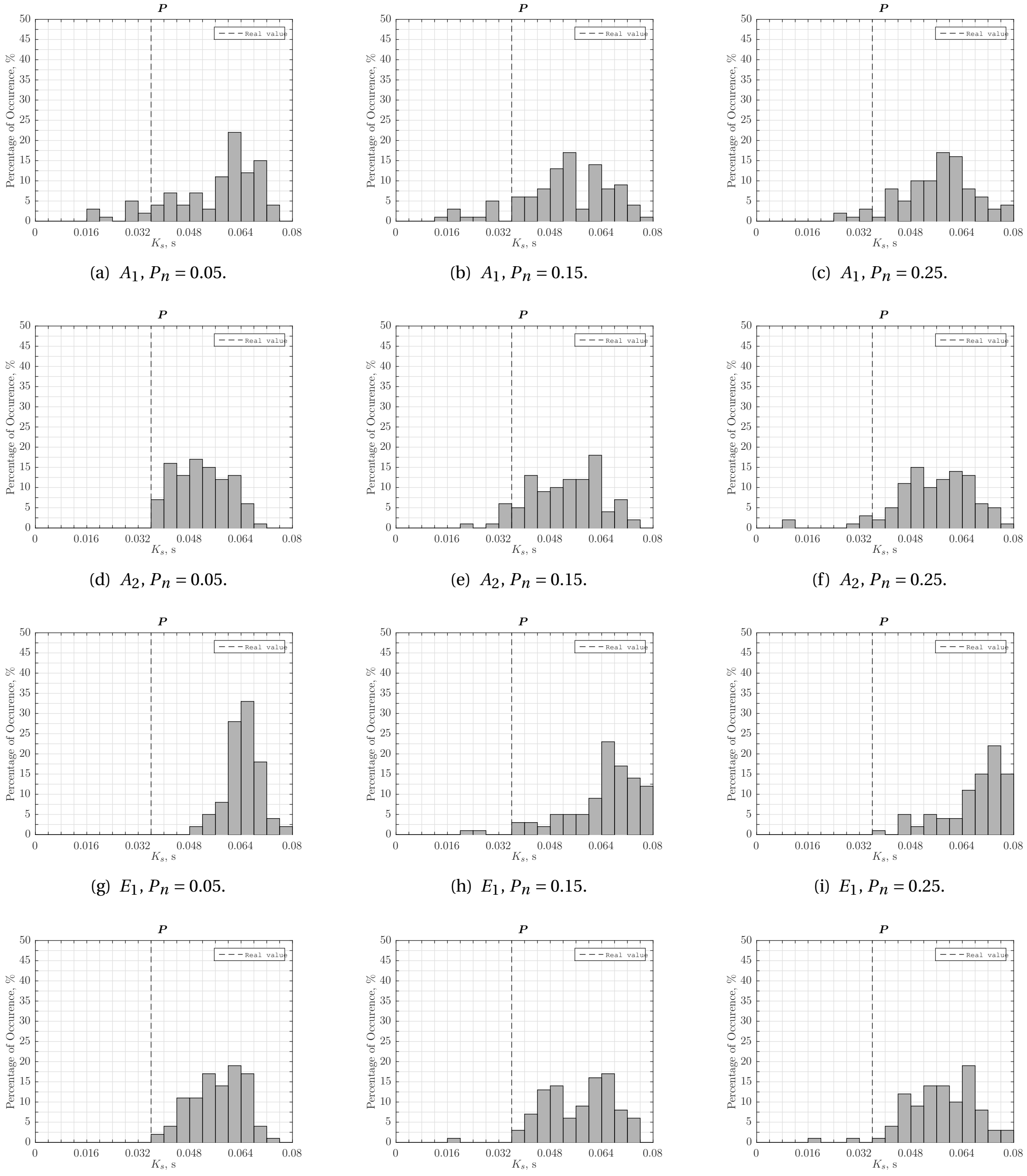
(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.6: Distribution of K_s estimations in 100 Monte Carlo simulations for different SFs. Condition $\bar{P}$, Region I.

Condition P



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

(j) E_2, $P_n = 0.05$.

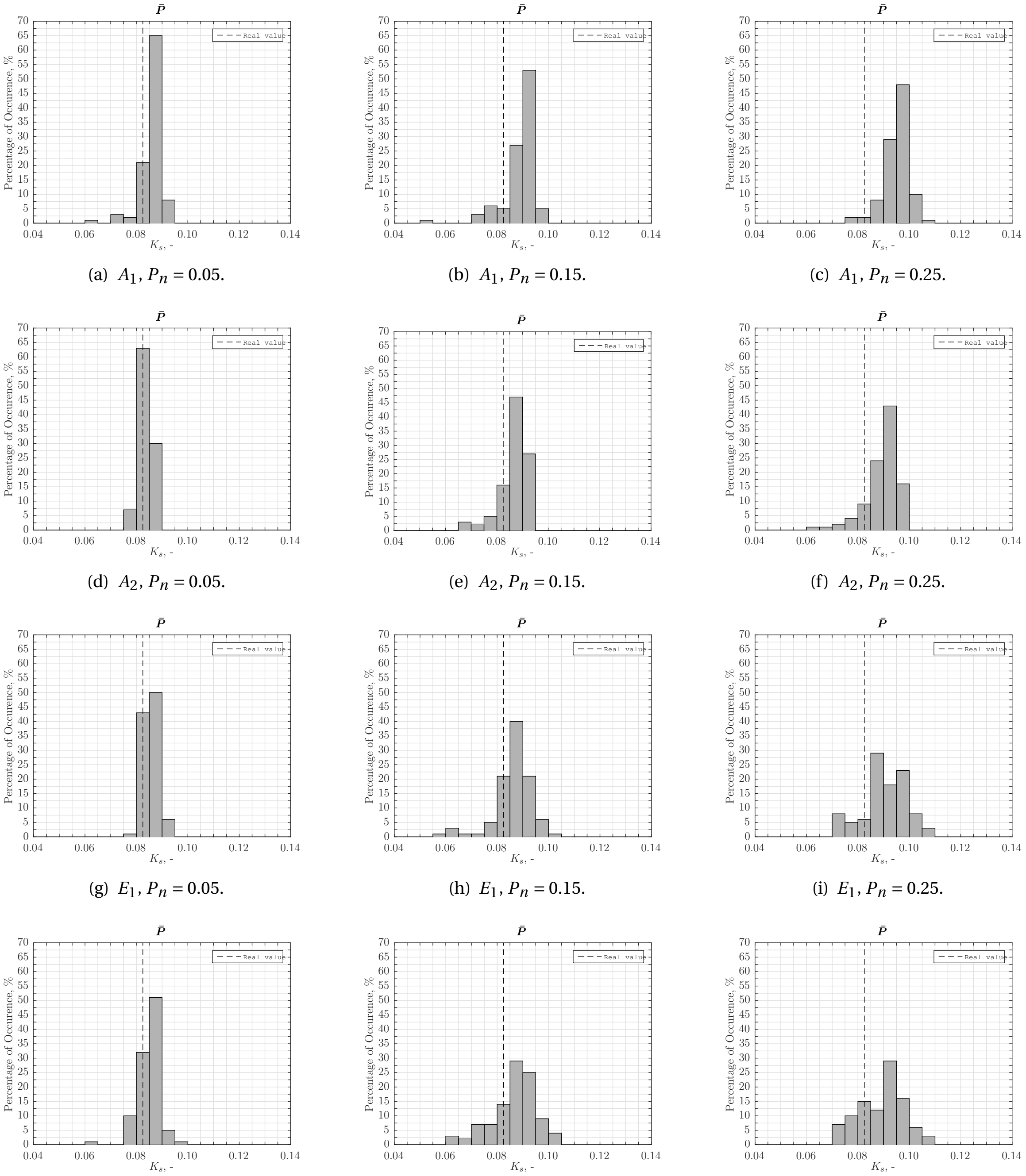
(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.7: Distribution of K_s estimations in 100 Monte Carlo simulations for different SFs. Condition P, Region I.

D.2.2. Region III

Condition $\bar{P}$



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

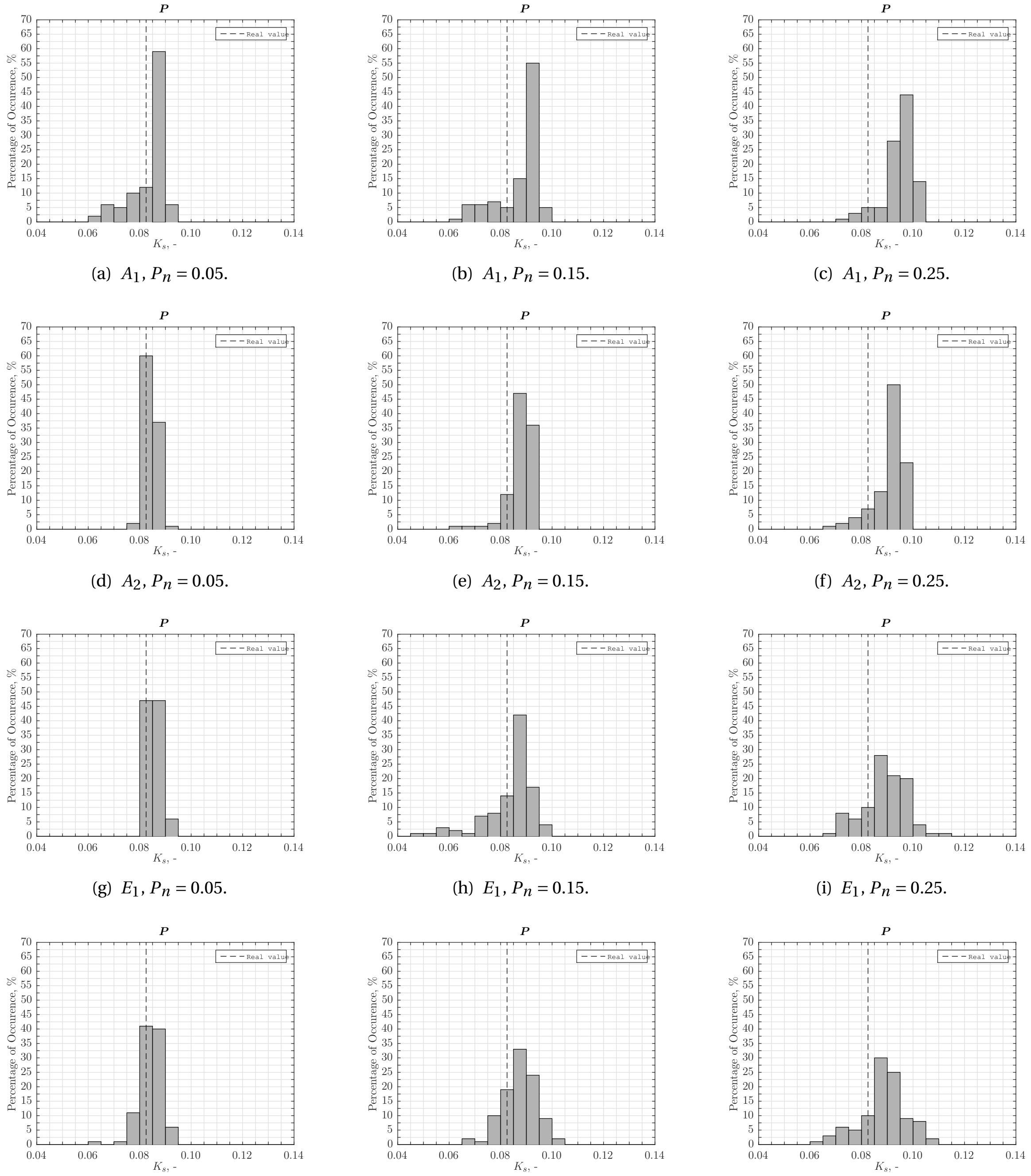
(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.8: Distribution of K_s estimations in 100 Monte Carlo simulations for different SFs. Condition $\bar{P}$, Region III.

Condition P



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

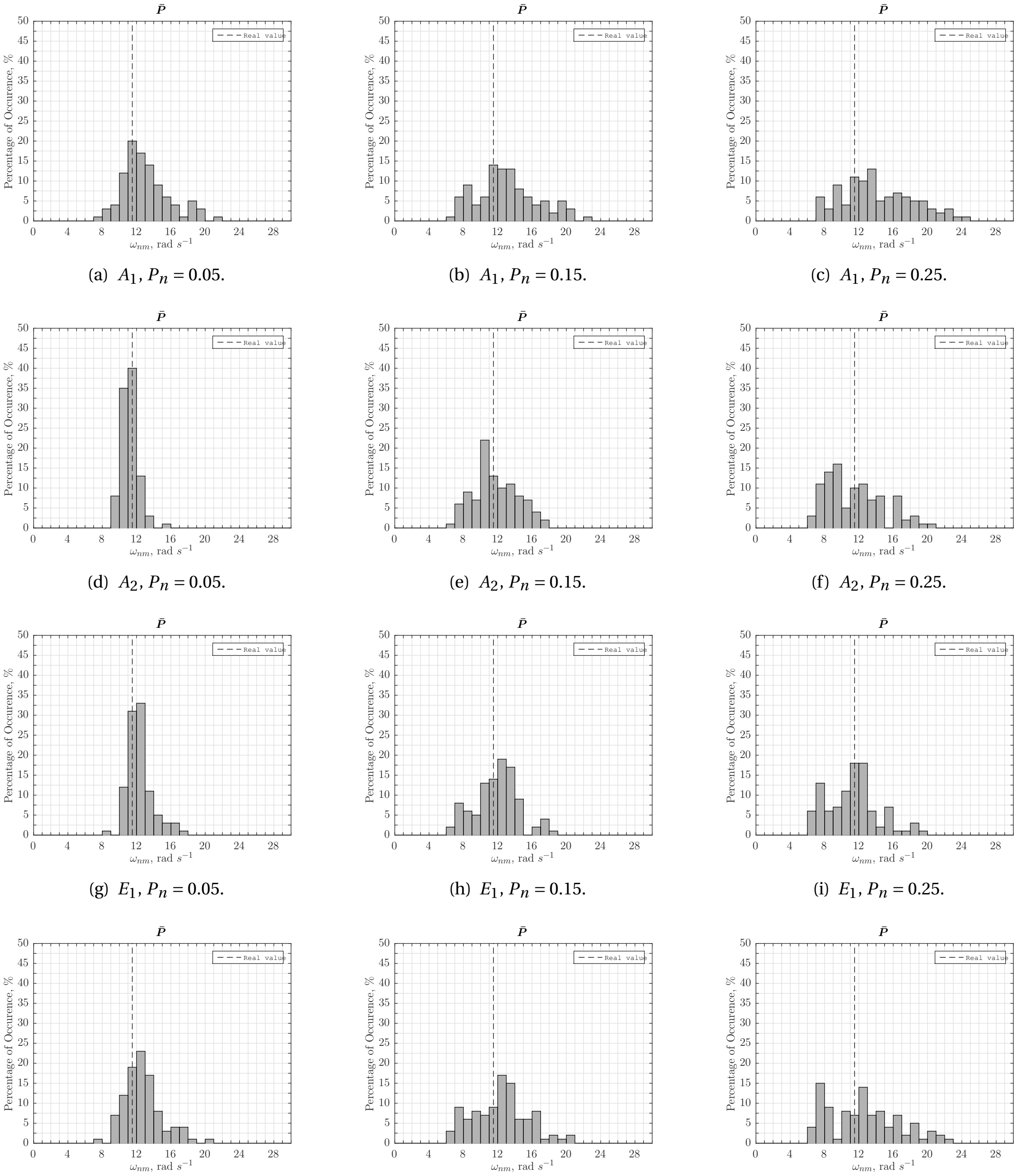
(l) E_2, $P_n = 0.25$.

Figure D.9: Distribution of K_s estimations in 100 Monte Carlo simulations for different SFs. Condition P, Region III.

D.3. ω_{nm}

D.3.1. Region I

Condition $\bar{P}$



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

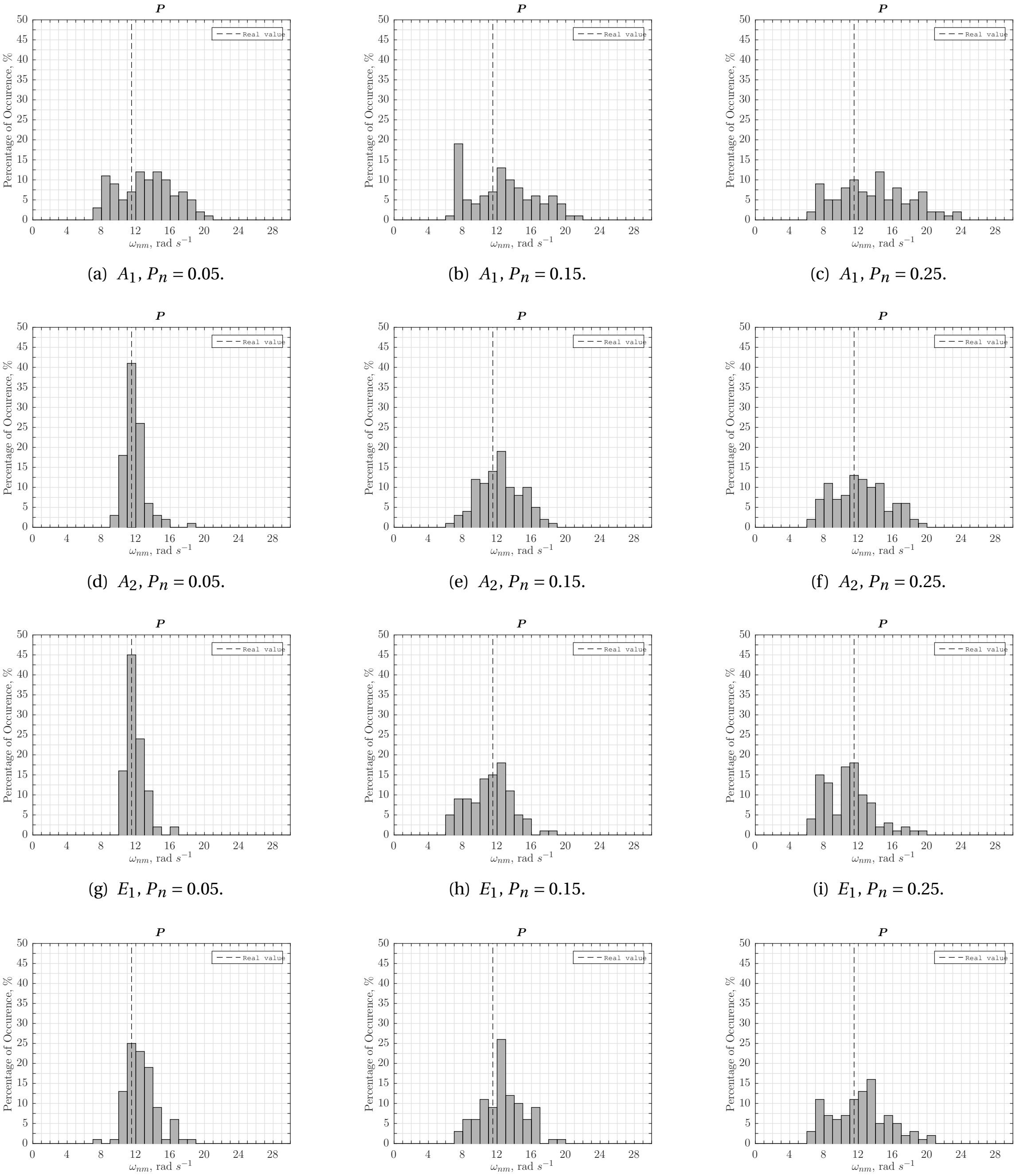
(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.10: Distribution of ω_{nm} estimations in 100 Monte Carlo simulations for different SFs. Condition $\bar{P}$, Region I.

Condition P



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

(j) E_2, $P_n = 0.05$.

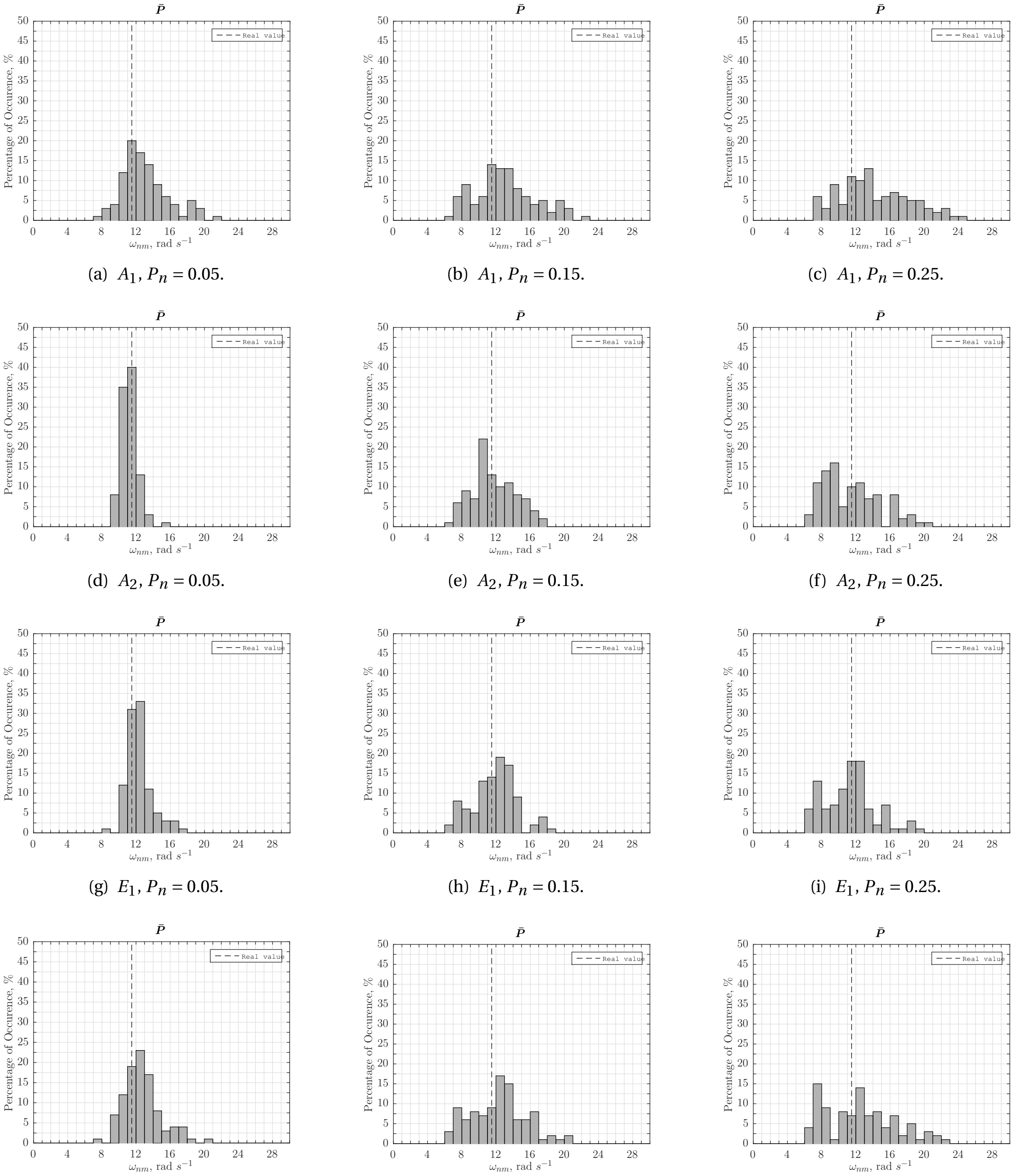
(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.11: Distribution of ω_{nm} estimations in 100 Monte Carlo simulations for different SFs. Condition P, Region I.

D.3.2. Region III

Condition $\bar{P}$



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

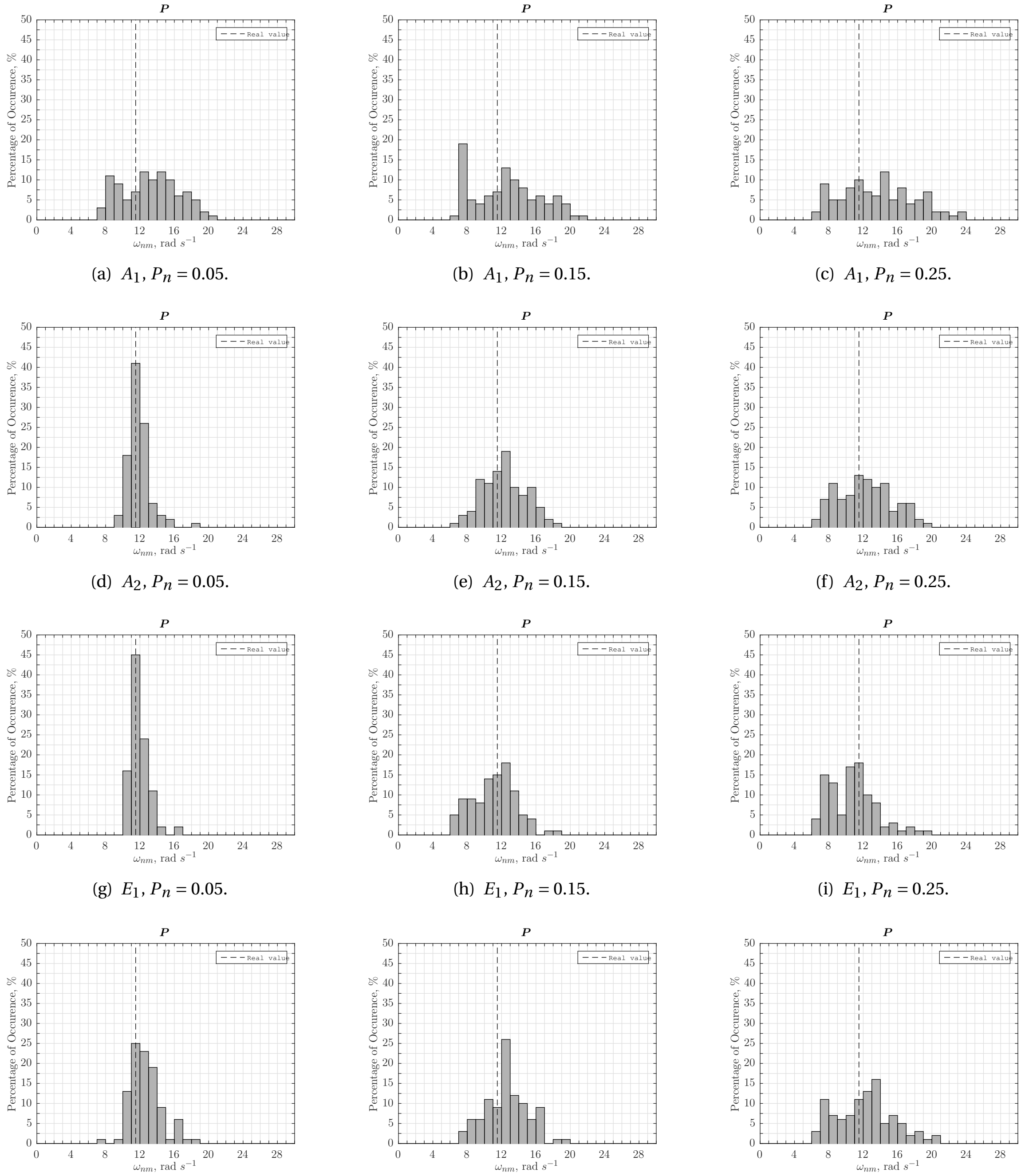
(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.12: Distribution of ω_{nm} estimations in 100 Monte Carlo simulations for different SFs. Condition $\bar{P}$, Region III.

Condition P



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

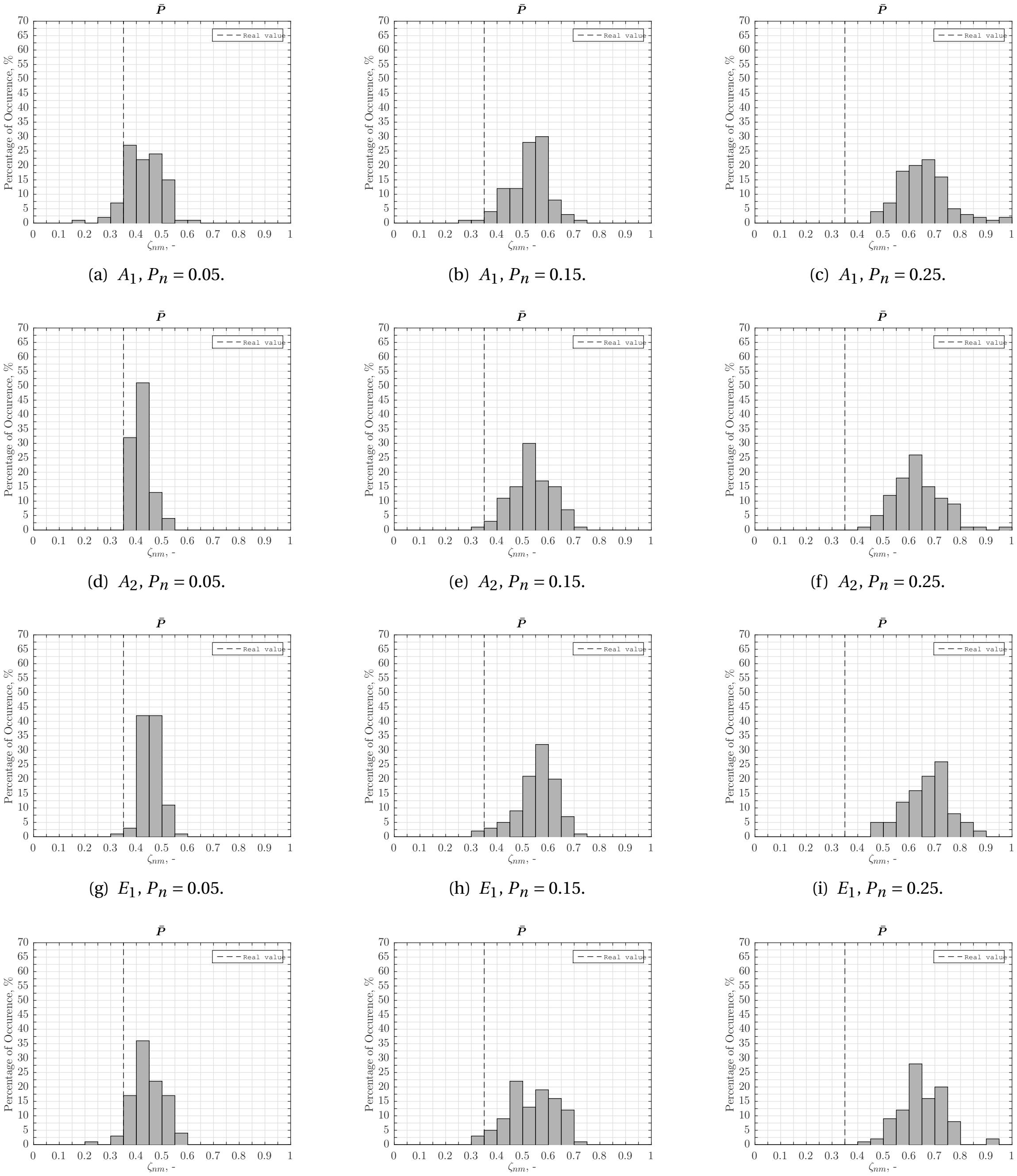
(l) E_2, $P_n = 0.25$.

Figure D.13: Distribution of ω_{nm} estimations in 100 Monte Carlo simulations for different SFs. Condition P, Region III.

D.4. ζ_{nm}

D.4.1. Region I

Condition $\bar{P}$



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

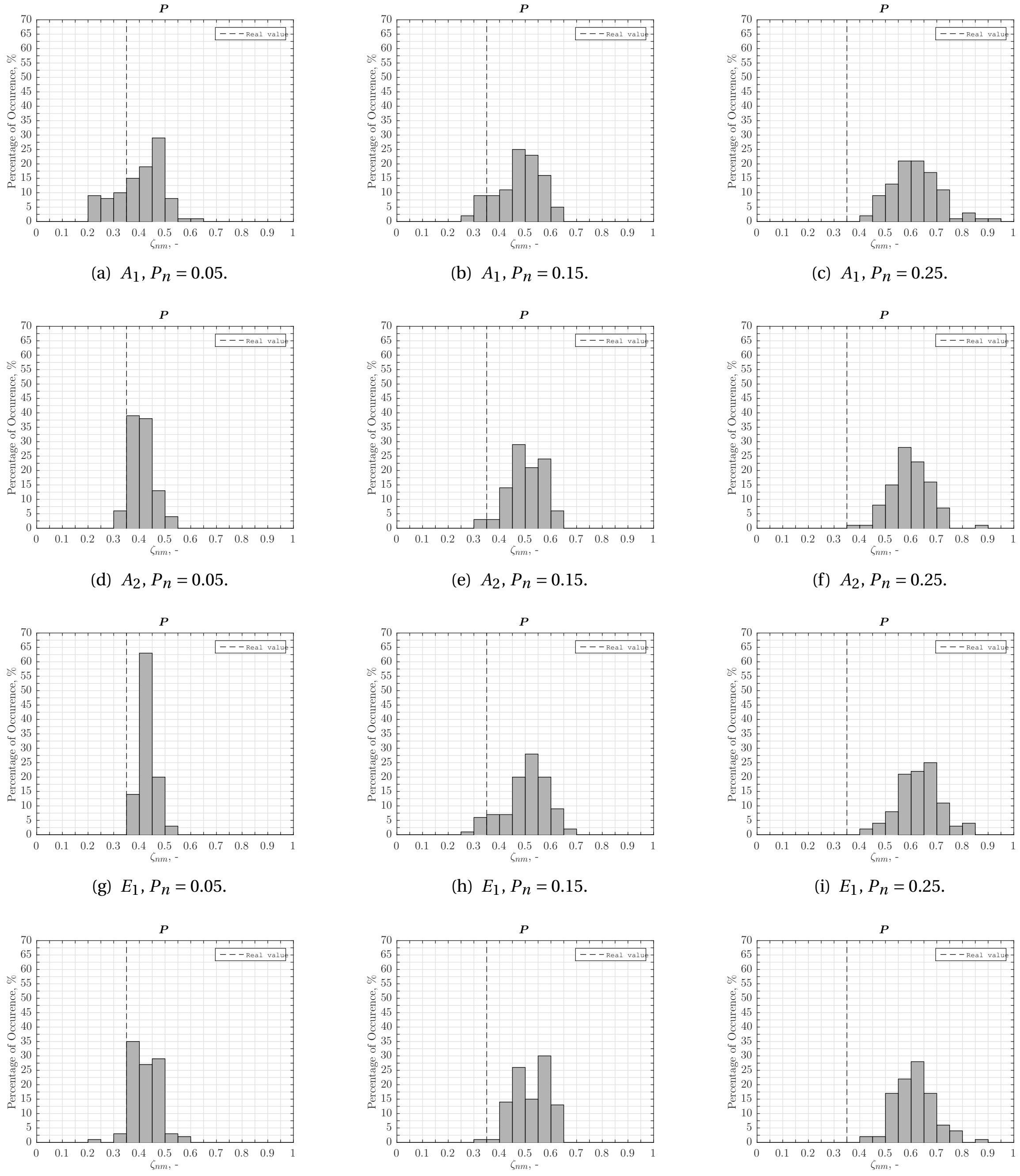
(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.14: Distribution of ζ_{nm} estimations in 100 Monte Carlo simulations for different SFs. Condition $\bar{P}$, Region I.

Condition P



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

(j) E_2, $P_n = 0.05$.

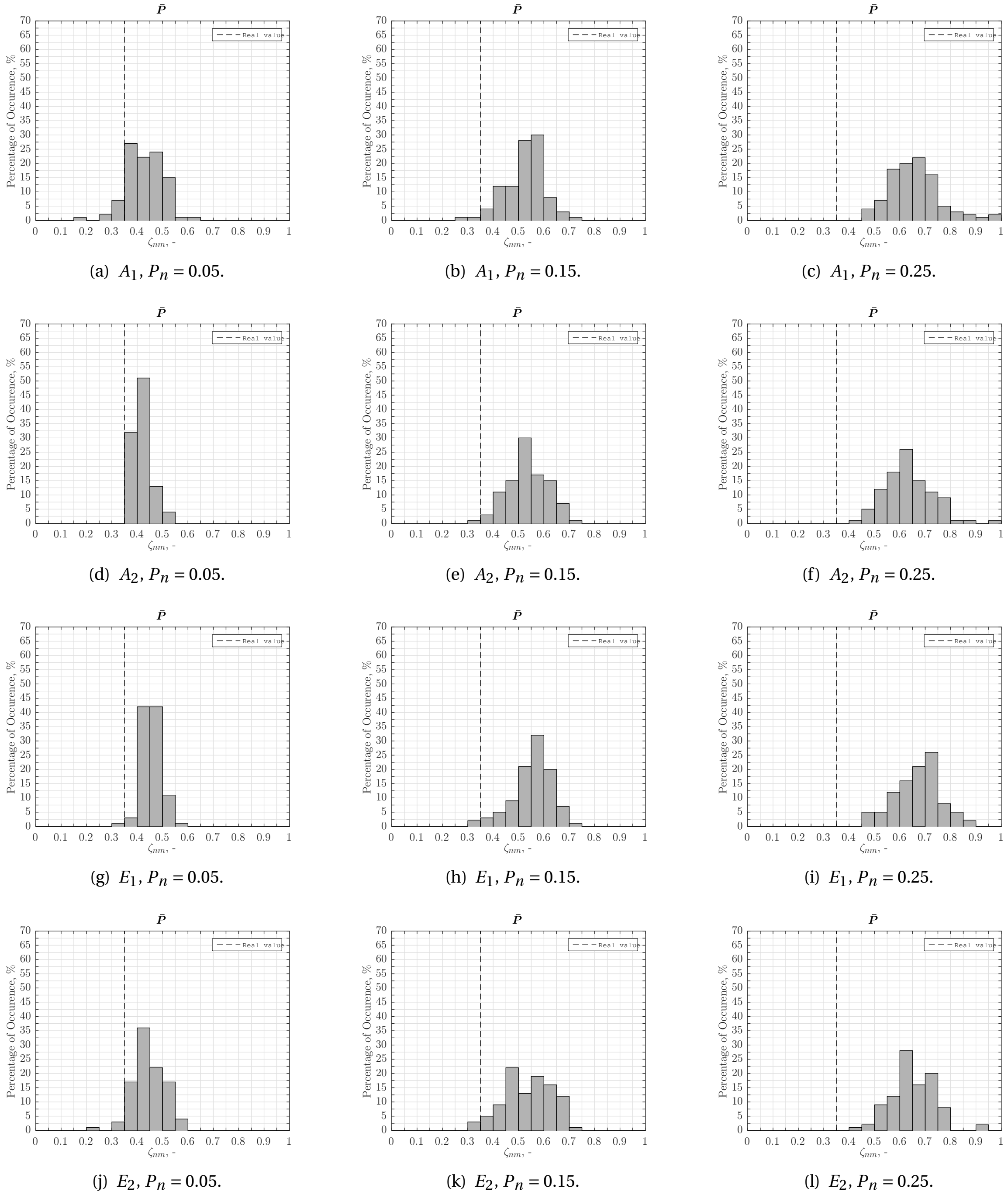
(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.15: Distribution of ζ_{nm} estimations in 100 Monte Carlo simulations for different SFs. Condition P, Region I.

D.4.2. Region III

Condition $\bar{P}$



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

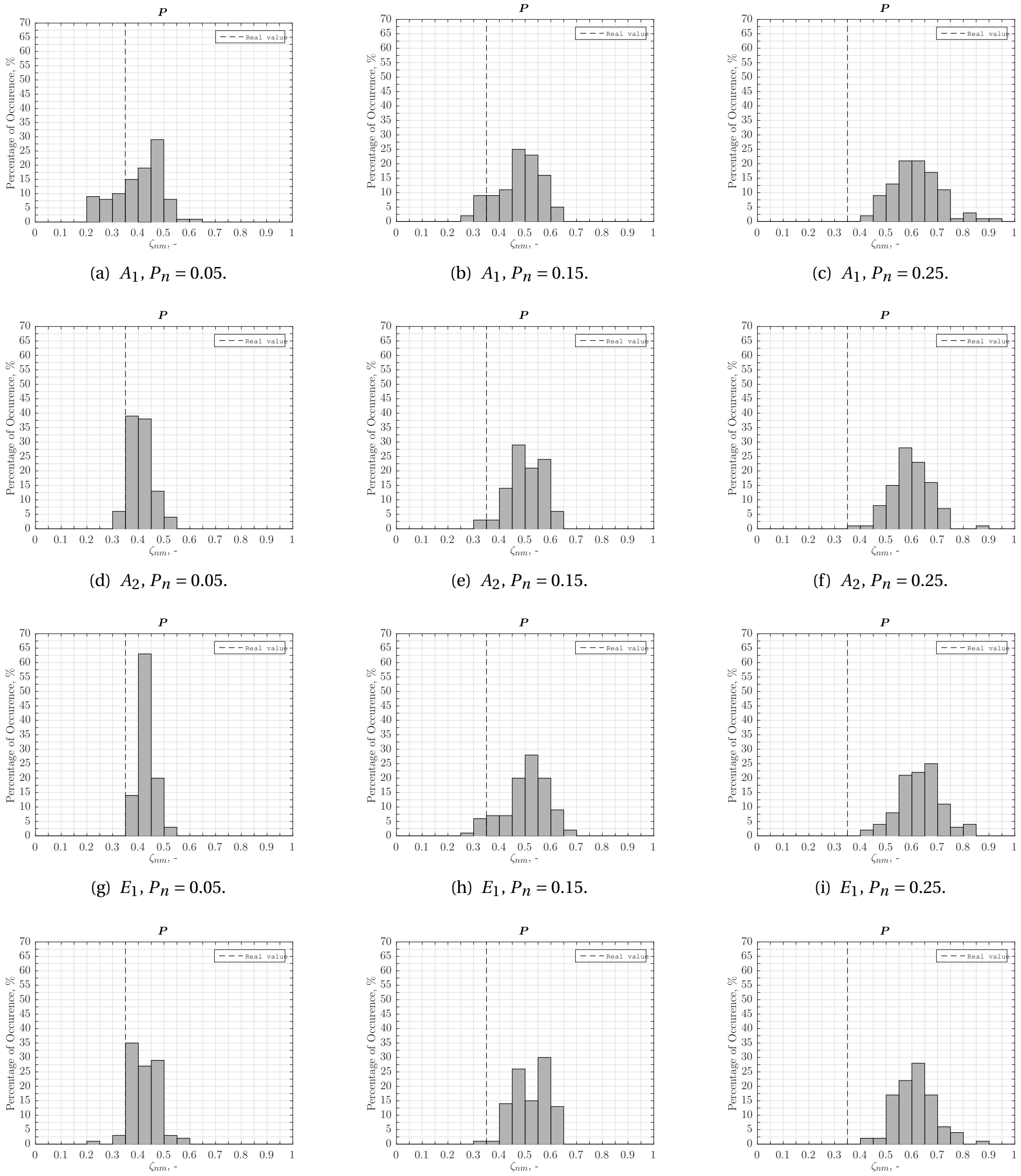
(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.16: Distribution of ζ_{nm} estimations in 100 Monte Carlo simulations for different SFs. Condition $\bar{P}$, Region III.

Condition P



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

(j) E_2, $P_n = 0.05$.

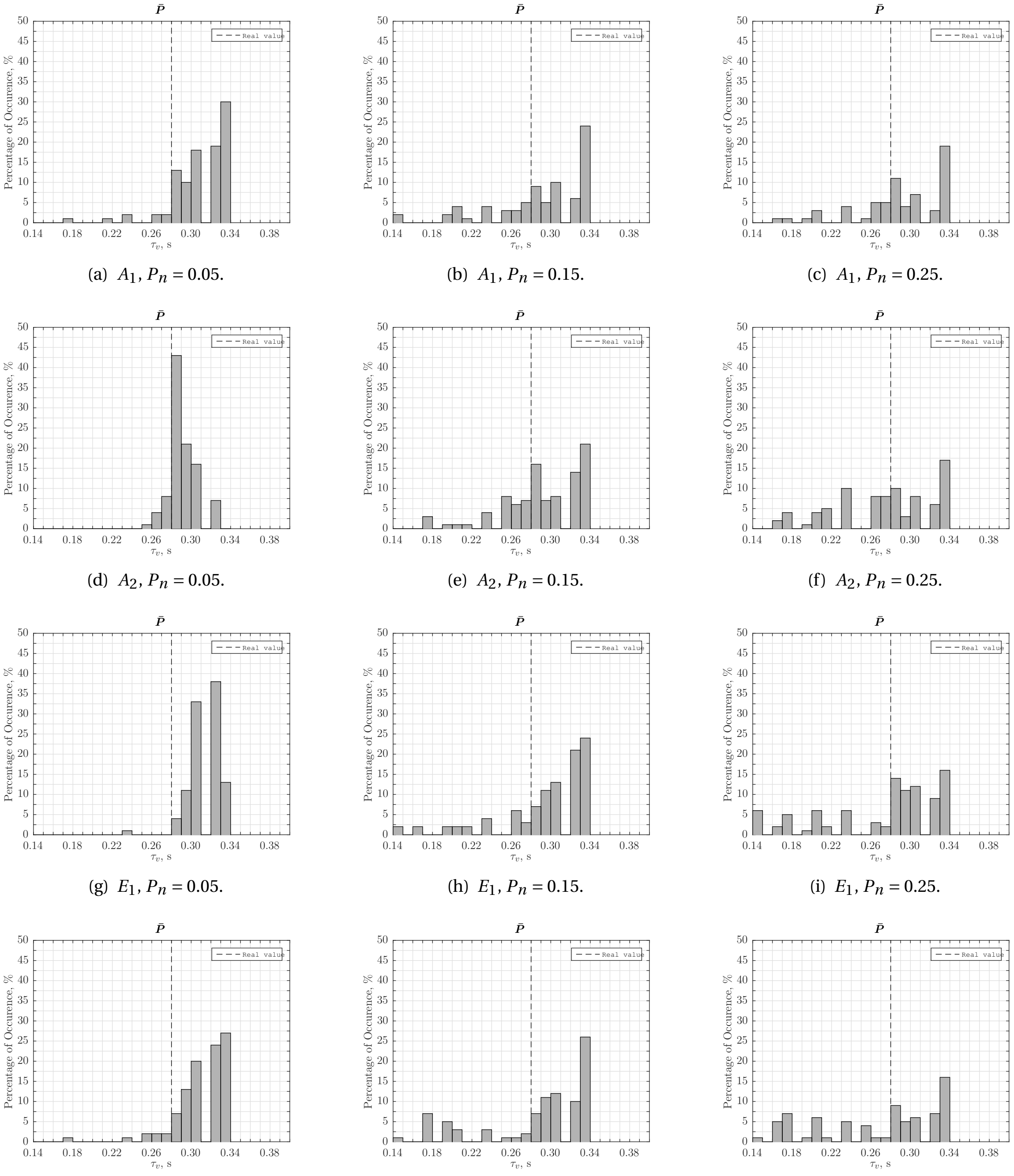
(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.17: Distribution of ζ_{nm} estimations in 100 Monte Carlo simulations for different SFs. Condition P, Region III.

D.5. τ_v

Condition $\bar{P}$



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

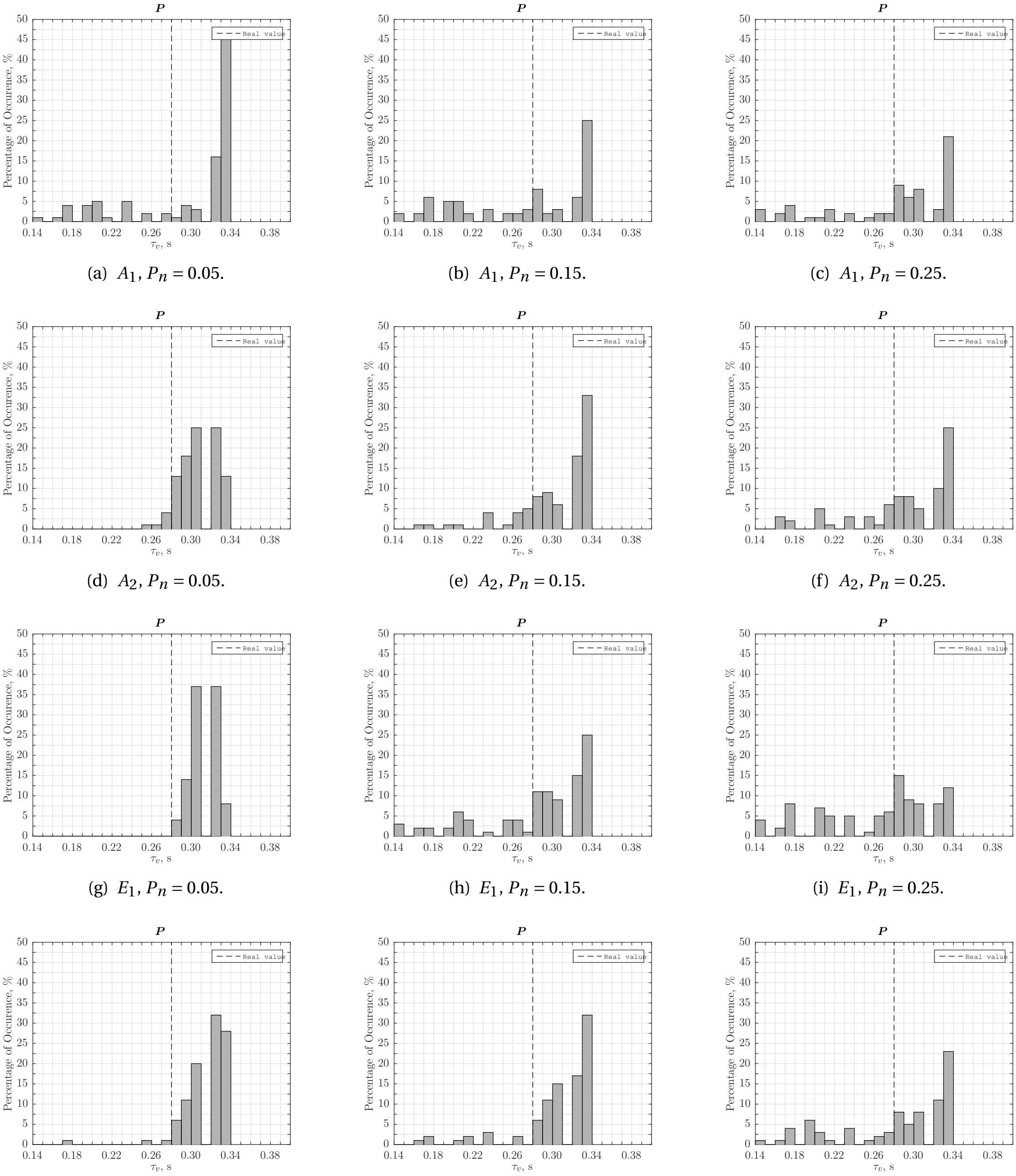
(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.18: Distribution of τ_v estimations in 100 Monte Carlo simulations for different SFs. Condition $\bar{P}$.

Condition P



(a) A_1, $P_n = 0.05$.

(b) A_1, $P_n = 0.15$.

(c) A_1, $P_n = 0.25$.

(d) A_2, $P_n = 0.05$.

(e) A_2, $P_n = 0.15$.

(f) A_2, $P_n = 0.25$.

(g) E_1, $P_n = 0.05$.

(h) E_1, $P_n = 0.15$.

(i) E_1, $P_n = 0.25$.

(j) E_2, $P_n = 0.05$.

(k) E_2, $P_n = 0.15$.

(l) E_2, $P_n = 0.25$.

Figure D.19: Distribution of τ_v estimations in 100 Monte Carlo simulations for different SFs. Condition P.

Bibliography

[1] Pbsid toolbox. `http://www.dcsc.tudelft.nl/~jwvanwingerden/pbsid/pbsidtoolbox_product_page.html`.

[2] Alessandro Chiuso. The role of vector autoregressive modeling in predictor-based subspace identification. *Automatica*, 43(6):1034 – 1048, 2007. ISSN 0005-1098. doi: http://dx.doi.org/10.1016/j.automatica.2006.12.009. URL `http://www.sciencedirect.com/science/article/pii/S000510980700057X`.

[3] Raymond E.Magdaleno Duane T. McRuer and George P. Moore. A neuromuscular actuation system model. *IEEE Transactions on Man-Machine Systems*, MMS-9(3), sep 1968.

[4] L. Giarré, D. Bauso, P. Falugi, and B. Bamieh. {LPV} model identification for gain scheduling control: An application to rotating stall and surge control problem. *Control Engineering Practice*, 14(4):351 – 361, 2006. ISSN 0967-0661. doi: http://dx.doi.org/10.1016/j.conengprac.2005.01.013. URL `http://www.sciencedirect.com/science/article/pii/S0967066105000511`.

[5] IEEE. Standard on transitions, pulses, and related waveforms, 2003.

[6] M. H. Wright J. C. Lagarias, J. A. Reeds and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.

[7] W.H.J.J. van Staveren & M. Mulder J.A. Mulder, J.C. van der Vaart. *Lecture Notes AE4304 - Aircraft Responses to Atmospheric Turbulence*. Delft University of Technology - Faculty of Aerospace Engineering, 2015.

[8] Martin Kers. Maximum Likelihood Estimation of Linear Time-Varying Pilot-Vehicle System Parameters. Master's thesis, Delft University of Technology, August 2012.

[9] G. Mercère; M. Lovera and E. Laroche. Identification of a flexible robot manipulator using a linear parameter-varying descriptor state-space structure. In *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011.

[10] Duane T. McRuer and Henry R. Jex. A review of quasi-linear pilot models. *IEEE Transactions on Human Factors in Electronics*, HFE-8(3), sep 1967.

[11] Frank M. Nieuwenhuizen, Peter M. T. Zaal, Max Mulder, Marinus M. van Paassen, and Jan A. Mulder. Modeling Human Multichannel Perception and Control Using Linear Time-Invariant Models. *Journal of Guidance, Control, and Dynamics*, 31(4):999–1013, July-August 2008. doi: 10.2514/1.32307.

[12] Mario Olivari, Frank M. Nieuwenhuizen, H. H. Bülthoff, and Lorenzo Pollini. Identifying Time-Varying Neuromuscular System with a Recursive Least-Squares Algorithm: a Monte-Carlo Simulation Study. In *Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics, San Diego (CA)*, pages 3573–3578, October 2014. doi: 10.1109/SMC.2014.6974484.

[13] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. Upper Saddle River, NJ, 1999.

[14] A.V. Oppenheim, A.S. Willsky, and S.H. Nawab. *Signals and Systems*. Prentice-Hall signal processing series. Prentice Hall, 1997. ISBN 9780138147570. URL `https://books.google.nl/books?id=LwQqAQAAMAAJ`.

[15] Daan M. Pool. Pilot Control in Compensatory Roll Axis Tracking Tasks. Master's thesis, Delft University of Technology, 2007.

[16] A. Pronker. Estimating time-varying neuromuscular admitance during steering tasks (preliminary msc. thesis). Master's thesis, Faculty of Aerospace Engineering - Delft University of Technology, 2015.

[17] Nicole Roggenkämper, Daan M. Pool, Frank M. Drop, Marinus M. van Paassen, and Max Mulder. Objective ARX Model Order Selection for Multi-Channel Human Operator Identification. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference, Washington, D.C.*, number AIAA-2016-4299, June 2016. doi: 10.2514/6.2016-4299.

[18] W. Rugh. *Linear system theory*. Upper Saddle River, 1996.

[19] Peter M. Thompson, David H. Klyde, and Martin J. Brenner. Wavelet-Based Time-Varying Human Operator Models. In *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit, Montreal (CA)*, number AIAA-2001-4009, 2001.

[20] Roland Tóth, Vincent Laurain, Marion Gilson, and Hugues Garnier. Instrumental variable scheme for closed-loop LPV model identification. *Automatica*, 48:2314–2320, 2012. doi: 10.1016/j.automatica.2012.06.037.

[21] Kasper van der El, Daan M. Pool, Herman J. Damveld, Marinus M. van Paassen, and Max Mulder. An Empirical Human Controller Model for Preview Tracking Tasks. *IEEE Transactions on Cybernetics*, 2015. doi: 10.1109/TCYB.2015.2482984.

[22] J. W. van Wingerden. *Control of Wind Turbines with 'Smart' Rotors: Proof of Concept & LPV Subspace Identifcation*. PhD thesis, Technische Universiteit Delft, 2008.

[23] J.W. van Wingerden and M. Verhaegen. Subspace identification of bilinear and lpv systems for open and closed loop data. *Automatica*, 45(2):372–381, 2009.

[24] A. Varga. Balancing-free square-root algorithm for computing singular perturbation approximations. In *Proc. of 30th IEEE CDC*, pages 1062–1065, Brighton, UK, 1991.

[25] Peter M. T. Zaal and Daan M. Pool. Multimodal Pilot Behavior in Multi-Axis Tracking Tasks with Time-Varying Motion Cueing Gains. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference, National Harbor (MD)*, number AIAA-2014-0810, January 2014. doi: 10.2514/6.2014-0810.

[26] Peter M. T. Zaal and Barbara T. Sweet. Estimation of Time-Varying Pilot Model Parameters. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference, Portland, Oregon, Aug. 8-11*, number AIAA-2011-6474, 2011.

[27] Peter M.T. Zaal. Manual control adaptation to changing vehicle dynamics in roll–pitch control tasks. *Journal of Guidance, Control and Dynamics*, 2016.