Identifying Feedback- and Feedforward Manual Control Behavior Using Subsystem Identification

Master of Science Thesis

P.J.C. Bentinck

TU Delft

Identifying Feedback- and Feedforward Manual Control Behavior Using Subsystem Identification

Master of Science Thesis

by

P.J.C. Bentinck

to obtain the degree of Master of Science
at the Delft University of Technology.

January 15, 2021

Student number: 4229827
Thesis committee: Dr. ir. D. M. Pool, TU Delft
Dr. ir. K. van der El, TU Delft
Prof. dr. ir. M. Mulder, TU Delft
Dr. ir. W. Mugge , TU Delft

Cover figure retrieved from: https://www.explorekentuckylake.com/_images/header/lbl-scenic-drives.jpg

TU Delft

Department of Control & Operations, Section Control & Simulation
Faculty of Aerospace Engineering
Delft University of Technology

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled **Identifying Feedback- and Feedforward Manual Control Behavior Using Subsystem Identification** by P. J. C. Bentinck in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: _____

Assessment Committee:

Professor and Chair: _____
Prof. dr. ir. M. Mulder

Supervisors: _____
Dr. ir. D. M. Pool

_____
Dr. ir. K. van der El

External Examiner: _____
Dr. ir. W. Mugge

Preface

This report marks the end of the Master of Science thesis project that I have been working on over the past year and a bit. Looking back, it has been an odd time, starting with my return from my internship abroad, to performing 'regular' thesis work at the university to spending the last 10 months at home due to a COVID-19 pandemic. Although such a period naturally contains ups and downs, I have enjoyed working on this project and will keep good memories about it.
There are a number of people I would like to thank for their efforts over this period, starting with those

that have helped me writing this thesis. Daan, thank you for your committed guidance in my thesis project. Thanks to your extensive guidance I never felt like working completely 'alone', even despite being at home for the majority of the project. It is unfortunate that the trip from Nashville to Lexington, to meet Jesse and his department, has not taken place, but I hope this will happen somewhere in the future! Kasper, thank you for all your feedback and inspiration for my thesis. Not only your feedback in person, but also your work itself has been a great source for me. You can clearly see that the copy of your dissertation, which Max gave to me at the start of my thesis, has been opened and used more than a few times. Max, thank you for all your enthusiasm and feedback. One of the reasons for choosing C&S has been your enthusiasm and I am very glad with my decision. Your comments always helped me keep an eye on the bigger picture of my work and I enjoyed the meetings with the stories about trips to the United States and other places around the world.
Besides the people that helped me write my thesis, I would like the persons that helped me personally.

First, I would like to thank the people from 2.44, who made the thesis work a lot more fun during the first months of the project. Furthermore, I would like to thank my roommates who were always able to make me laugh and take a break from all the serious work. To my friends and family, thank you for supporting me in any way possible over the past year, whether it was to find support or just to have good time. Finally, mom, dad and Thom, thanks for everything you have done for me over the past year, it meant a lot to me.
With that, an end has also come to my 7 and a half years of time at- and around TU Delft. It has been

a great time and I will definitely miss and remember both the work at the university and the life around it. For now, it is on to a new adventure, but Delft will always be a special place for me. Thank you.

Pieter-Bas Bentinck
Delft, January 2020

Contents

Report Overview ix

List of Figures xi

List of Tables xv

List of Abbreviations xvii

List of Symbols xix

I Paper I: On the Application of a Subsystem Identification Technique to a Manual Control Pursuit- and Preview Model 1

II Paper II: Comparison of Identification Techniques used to Identify Feedback- and Feedforward Control Behaviour in Preview Tracking Tasks 15

III Appendices 31

1 Candidate Pool Limit Selection 33

2 Candidate Pool Densities Paper I 37

3 Forcing Function Comparison 39
3.1 Forcing Function Description . . . 39
3.1.1 Sheffler et al. . . . 39
3.1.2 Van der El et al. . . . 39
3.2 Comparison . . . 39

4 Control Output Signal Simulation 43
4.1 Remnant Model Validation . . . 43
4.2 Open- and Closed Loop Simulation Comparison . . . 44

5 Human Operator Model Parameter Sensitivity 45
5.1 Method . . . 45
5.2 Results and Discussion . . . 46

6 Experiment Plan 49
6.1 Experiment Setup . . . 49
6.1.1 Experiment variables . . . 49
6.1.2 Experiment matrix . . . 49
6.2 Procedures . . . 50

7 Experiment Results 51
7.1 Signal Variances . . . 51
7.2 Crossover Frequency and Phase Margin . . . 52

8 Comparison U-Methods 55
8.1 Mean Absolute Error . . . 55
8.2 Parameter Fits . . . 56
8.3 Variance-Accounted-For . . . 57
8.4 Frequency Domain Error . . . 57

IV Preliminary Report **61**

1 Introduction **63**

2 Modelling and identification of feedback and feedforward in manual control **65**
2.1 Modelling the human controller . . . 65
2.1.1 Successive Organisation of Perception . . . 65
2.1.2 Model structures . . . 67
2.2 Tracking tasks including feedback- and feedforward . . . 68
2.2.1 Pursuit- and preview tracking tasks . . . 68
2.2.2 Predictable target signals . . . 70
2.3 Identification methods . . . 70
2.3.1 Multi-loop Fourier Coefficients method . . . 70
2.3.2 Identification using linear time-invariant models . . . 71
2.3.3 Time domain maximum likelihood estimation . . . 71
2.3.4 Kalman filtering . . . 71
2.3.5 Subsystem identification . . . 72
2.3.6 Predictor-Based Subspace Identification . . . 72
2.4 Synthesis . . . 72

3 Subsystem identification **73**
3.1 Problem formulation . . . 73
3.1.1 Sign conventions of subsystems . . . 74
3.2 Basic technique . . . 75
3.2.1 Candidate pool . . . 76
3.2.2 Identification algorithm . . . 77
3.3 Algorithm extensions . . . 78
3.4 SSID including feedback- and feedforward time delays . . . 80
3.4.1 Feedback- and feedforward candidate pool . . . 81
3.4.2 Identification of subsystems and time delays . . . 82
3.5 Summary of SSID . . . 83
3.6 Synthesis . . . 84

4 Application of SSID in examples **85**
4.1 Basic algorithm: general information . . . 85
4.1.1 Candidate pools . . . 85
4.2 Basic algorithm: noisy frequency response data . . . 86
4.2.1 Generating noise realizations . . . 86
4.2.2 Relating noise levels to the human remnant . . . 87
4.3 Basic algorithm: examples and results . . . 88
4.3.1 Example 1: Noiseless data and true parameters in the candidate pool . . . 89
4.3.2 Example 2: Noiseless data and true parameters outside the candidate pool . . . 89
4.3.3 Example 3: Noisy data and true parameters in the candidate pool . . . 90
4.3.4 Example 4: Noisy data and true parameters outside the candidate pool . . . 91
4.4 Extended algorithm . . . 92
4.4.1 Problem setup . . . 92
4.4.2 Noiseless data example . . . 93
4.4.3 Noisy frequency response data . . . 93
4.5 Synthesis . . . 93

5 Research scope **95**
5.1 Research questions . . . 95
5.2 Research approach . . . 96

6 Application of SSID to alternative model structures **99**
6.1 Notations . . . 99
6.2 Problem derivation . . . 99
6.3 Applying SSID to a TX pursuit- and preview model . . . 100
6.3.1 Applying SSID to the full model . . . 101
6.3.2 Applying SSID to the reduced model . . . 101
6.4 Applying SSID to a single loop preview model . . . 103
6.5 Synthesis . . . 104

7 Simulation Setup **105**
7.1 Settings . . . 105
7.1.1 Forcing function design . . . 105
7.1.2 Human operator models . . . 106
7.1.3 Remnant modelling . . . 107
7.2 Signal shift in Simulink implementation . . . 108
7.3 Simulation verification . . . 109
7.3.1 Forcing function power spectra . . . 109
7.3.2 Time domain traces . . . 109
7.3.3 Remnant modelling . . . 109
7.3.4 Multi-loop identification . . . 110
7.4 Recommendations . . . 112

8 Future research **115**
8.1 Refining the research scope . . . 115
8.1.1 Selecting specific tracking task settings . . . 115
8.1.2 Refining the set of independent variables . . . 116
8.2 Modifications to the simulation setup . . . 117
8.3 Running simulations and performing analysis . . . 117
8.4 Comparing the methods with experimental data . . . 117
8.5 Other analyses . . . 118

9 Conclusions **119**

Bibliography **121**

Report Overview

This report is the MSc thesis report of Pieter-Bas Bentinck, entitled *Identifying Feedback- and Feedforward Manual Control Behavior Using Subsystem Identification*. The report consists of four parts. The first two parts contain the core of the research conducted for this thesis project in the form of two papers. The paper in Part I is titled *On the Application of a Subsystem Identification Technique to a Manual Control Pursuit- and Preview Model*. Here, a recently developed subsystem identification (SSID) technique is modified to estimate the parameters of a versatile preview model. Part II consists of the paper *Comparison of Identification Techniques used to Identify Feedback- and Feedforward Control Behaviour in Preview Tracking Tasks*. The paper quantifies the performance of the SSID technique with respect to other techniques that are often used in similar tracking tasks.

Part III contains the appendices to the reports. Here, information and results are presented that were not included in the papers, but are valuable to be documented for future research. The appendices consist of details regarding the setup and the eventual candidate pools used for the SSID technique in this research, a comparison of the forcing functions used in various feedback- and feedforward experiments, the simulation of the control output signal, parameter sensitivity of the human operator model, an experiment plan and finally several additional experiment- and simulation results.

Part IV contains the Preliminary Thesis Report. This report contains a literature study into the subsystem identification technique and a consequent project plan for the thesis. It can be seen as the prequel of the first three parts.

List of Figures

3.1 Original (a) and scaled time-domain forcing functions used by Sheffler et al. (2019) and van der El et al. (2020). In both figures, the forcing function of van der El et al. (2020) is kept constant. . . . 40
3.2 Frequency domain comparison of the forcing functions used by van der El et al. (2020) and McRuer et al. (1965). . . . 40
3.3 Comparison of the controlled element dynamics used by van der El et al. (2020) and McRuer et al. (1965). . . . 41

4.1 Experimentally obtained- and two simulated power spectral densities for SI (a) and DI (b). Data from subject two, obtained using the $U2_{td}$ method is used. . . . 43
4.2 Example of the differences between the measured control output and the open- and closed-loop simulations. Data from the SI task is selected for subject 3, with the parameter fits from the $U2_{td}$ method. . . . 44

5.1 Sensitivity value $s_{norm_{x,\theta_i}}$ for both controlled element dynamics and all values of f_θ. . . . 47

7.1 Total tracking performance and the separate contributions due to the target- and disturbance functions and the remnant. . . . 51
7.2 Total control output variance and the separate contributions due to the target- and disturbance functions and the remnant. . . . 52
7.3 Remnant-to-control-output ratios $\frac{\sigma_{u_n}^2}{\sigma_u^2}$ for the different conditions. . . . 53
7.4 Crossover frequencies and phase margins for the target- and disturbance functions for the different experiment conditions. Results from the raw experimental data, as well as fits from the $U2_n$ method are shown. Furthermore, reference values obtained by van der El et al. (2020) and theoretical estimates from the theory of McRuer and Jex (1967) are shown. . . . 54

8.1 Mean absolute error (MAE) of the different methods for the evaluated noise ratios. Results are shown for single- (a) and double integrator (b) dynamics. . . . 55
8.2 Parameter estimates of the feedback subsystem $H_{o_{cmp}}$ for SI (a, c, e, g & i) and DI (b, d, f, h & j) CE dynamics. Errorbars indicate the 95% confindence interval. . . . 56
8.3 Parameter estimates of the feedforward subsystem H_{o_p} for SI (a, c & e) and DI (b, d & f) CE dynamics. Errorbars indicate the 95% confindence interval. . . . 57
8.4 Parameter estimates of the experiment obtained with the evaluated methods for both SI- (a) and DI (b) dynamics. . . . 58
8.5 Box plots showing the VAFs of the experiment results for both types of dynamics and all methods. . . . 58
8.6 Frequency domain error of the estimated and the true subsystems H_{o_t} and H_{o_x}. Results are shown for single- (a) and double integrator (b) dynamics, with errorbars indicating the mean and 95% confidence intervals. . . . 59

2.1 Schematic representations of the three stages of skill-based manual control behaviour, according to the Successive Organisation of Perception (McRuer et al., 1968): compensatory- (a), pursuit- (b) and precognitive (c) behaviour. The figures are reproduced largely based on the adaptions according to Mulder et al. (2018). . . . 66
2.2 Schematic representations of compensatory (a) and pursuit/preview displays (b). A pursuit display is basically a preview display with 0 preview ($\tau_p = 0$). Figures taken from van der El (2018). . . . 67

2.3 A three-channel block diagram of a human in a pursuit tracking task (a) and two reduced two-channel models in either the XT- (b) or the ET (c) configuration. Figure taken from van der El (2018). . . . 68
2.4 Block diagrams of the pursuit- and preview model proposed by van der El et al. (2016). Although the structure of the two models is different, the dynamics are identical. The figure is taken from van der El et al. (2016). . . . 69

3.1 Block diagram of the ET system for which H_{ot} and H_{oe} are to be identified. . . . 74
3.2 Block diagram of the ET system with sign conventions according to Zhang and Hoagg (2016a). . . . 75
3.3 Simplified block diagram of the MIMO system with two controlled element outputs according to Zhang and Hoagg (2016b). . . . 78
3.4 Block diagram of the model with multiple feedforward subsystems with nonlinear inputs according to Koushkbaghi et al. (2018). . . . 79
3.5 Block diagram of the model for which feedback- and feedforward subsystems as well as a feedback delay are identified by Zhang et al. (2018) . . . 80
3.6 Block diagram of the model for which feedback- and feedforward subsystems and time delays are identified by Sheffler et al. (2019) . . . 80
3.7 Schematic explaining the working of the subsystem identification method. . . . 84

4.1 Block diagram of the ET system with sign conventions according to Zhang and Hoagg (2016a). . . . 85
4.2 PSDs of $S_{uu_{\sigma_r}}$ and $S_{\sigma_r\sigma_r}$ for different values of the noise intensity index i and the PSD of S_{uu_r}. . . . 90
4.3 Difference between the identified parameters ϕ^+ and β^+ and the true parameters ϕ_* and β_* for different candidate pool densities, for noiseless frequency response data. The figure shows the original plots from (Zhang and Hoagg, 2016a) (a) and reproduced plots (b). . . . 90
4.4 Bode plots of the true- and of the identified feedforward transfer functions for j = 1,2 and 12. . . . 91
4.5 Difference between the identified parameters ϕ^+ and β^+ and the true parameters ϕ_* and β_* for different noise intensities, using the candidate pool Φ_0. The figure shows the original plots from Zhang and Hoagg (2016a) (a) and reproduced plots (b). For the reproduced values, the noise is derived in two ways: both from the analytical noise and from a separate noise realizations. . . . 91
4.6 Difference between the identified parameters ϕ^+ and β^+ and the true parameters ϕ_* and β_* for different candidate pool densities, for the four noise intensities $i = 5, 10, 12\&20$. The figure shows the original plots from Zhang and Hoagg (2016a) (a) and reproduced plots (b). The vertical lines in (b) indicate the highlighted conditions that are discussed in more detail. . . . 92
4.7 Difference between the identified parameters α^+, ϕ^+ and ψ^+, and the true parameters α_*, ϕ_* and ψ_* for different noise intensities, using the defined candidate pools Φ and Ψ. The noise index is proportional to the inverse of the noise intensity. . . . 93

5.1 Schematic overview of the difference between subsystem identification and the multi-loop Fourier Coefficients method. . . . 97

6.1 Basic TX model structure . . . 100
6.2 Single loop model structure for pursuit- and preview based on van der El et al. (2018b). 103

7.1 Diagram explaining the time shift of the target signal to deal with the negative time delays in the Simulink model . . . 108
7.2 Power spectra of the target (f_t) and disturbance (f_d) function by van der El (2018) (a) and reproduced from the data (b). . . . 109
7.3 Comparison of the simulated CE outputs to the target function for the SI pursuit- and preview models between van der El (2018) (a) and the reproduced data (b). . . . 110

7.4 Comparison of the simulated CE outputs to the target function for the DI pursuit- and preview models between van der El (2018) (a) and the reproduced data (b). 110
7.5 Comparison of the control output power spectra for the SIPR simulation between van der El (2018) (a) and the reproduced data (b). 111
7.6 Comparison of the experimentally estimated FRFs and the resulting model fits of van der El (2018) (a) and the reproduced FRFs estimated from the simulated models (b) for the target response H_{o_t} for the DIPR experiment. 111
7.7 Comparison of the experimentally estimated FRFs and the resulting model fits of van der El (2018) (a) and the reproduced FRFs estimated from the simulated models (b) for the output response H_{o_x} for the DIPR experiment. 112
7.8 Absolute values of the Fourier transform of the control output $U(j\omega)$ at both: the original target frequencies ω_t interpolated at the disturbance frequencies ω_d (a) and vice versa (b). 113
7.9 Phase angle between $-180°$ and $+180°$ for the control output $U(j\omega)$ at the original target frequencies ω_t interpolated at the disturbance frequencies ω_d (i.e., corresponding to Figure 7.8a). 113
7.10 Schematic drawing to indicate the error caused by linearly interpolating complex numbers.113

List of Tables

1.1 Maximum- and minimum values of the different parameters as obtained from van der El (2018) for DI CE dynamics . . . 34
1.2 Maximum- and minimum values of the different parameters as obtained from van der El (2018) for SI CE dynamics . . . 34
1.3 Maximum- and minimum values of the different parameters as obtained from van der El (2018) for both SI and DI CE dynamics and the selected ranges. . . . 35

2.1 Overview of different candidate pools used in Paper I . . . 37
2.2 Overview of candidate pool sizes used in Paper I . . . 38

4.1 Overview of the identified parameters for both dynamics for subject 2 using the $U2_{td}$ method. . . . 43

5.1 Overview of the model parameters used in the simulations. . . . 46
5.2 Examples of the different values for $\theta_i \cdot f_\theta$ and $\theta_i \cdot \frac{1}{f_\theta}$ for the different parameters for a value of $f_\theta = 2$ and DI CE dynamics. . . . 46

6.1 Balanced Latin Square of the experimental conditions. . . . 50

8.1 Overview of the methods evaluated in this research. . . . 55

3.1 Overview of sign conventions throughout the research on SSID . . . 74
3.2 Overview of the different versions of the SSID algorithm. . . . 80

4.1 Values of the noise-to signal ratio R_i given by Zhang and Hoagg (2016a) and from a reproduction, and the relative contribution of σ_r to the control output u, $\frac{\sigma_u^2 \sigma_r}{\sigma_u^2}$, for different noise intensities ($var(\sigma_r) = 4^{-i}$). . . . 89
4.2 Values of the identified parameters β^+ and ϕ^+ for different noise intensities and two candidate pool densities. . . . 92
4.3 Values of the identified parameters for several noise intensities and the true parameters of the performed example. . . . 94

6.1 The numerator- and denominator terms $N_f(j\omega)$ and $N_e(j\omega)$ for gain (G), single integrator (SI) and double integrator (DI) dynamics. . . . 104

7.1 Forcing function parameters as retrieved from van der El et al. (2016). . . . 106
7.2 Test conditions and model parameters as obtained from van der El et al. (2016). . . . 107

8.1 Different options for the tracking tasks to select as a reference condition. For each task, both the general settings and the evaluation with respect to specific trade-off criteria are displayed. . . . 116

List of Abbreviations

ARMAX	Autoregressive Moving-Average Exogenous
ARX	Autoregressive Exogenous
CE	Controlled Element
DFT	Discrete Fourier Transform
ET	Error-target
EX	Error-output
FB	Feedback
FC	Fourier Coefficients
FF	Feedforward
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FRF	Frequency Response Function
HC	Human Controller
HO	Human Operator
IMH	Internal Model Hypothesis
LPV	Linear Parameter Varying
LTI	Linear Time-Invariant
MIMO	Multiple-Input Multiple-Output
MLE	Maximum Likelihood Estimation
NMS	Neuromuscular System
OLS	Ordinary Least Squares
PBSID	Predictor-Based Subsapce Identification
PSD	Power Spectral Density
RLS	Rcursive Least Squares
SISO	Single-Input Single-Output
SOP	Successive Organisation of Perception
SSID	Subsystem Identification
TX	Target-output
VAF	Variance-Accounted-For

List of Symbols

This report contains an extensive description of a subsystem identification technique, of which several different versions of the algorithm are discussed. As the notations used in the different versions, which are inconsistent with respect to each other, are followed, a large amount of symbols occurs in this report. To provide a better overview of all symbols, a division is made between symbols that are generally used throughout the report and symbols that are only used in correspondence with the explanation of subsystem identification in Chapters 3 and 4. First the general symbols will be provided, followed by the subsystem identification-specific symbols.

General symbols

Symbol	Description	Unit
α	Feedforward numerator coefficients for the subsystem identification method including feedback and feedforward time delays	[-]
β	Feedforward numerator coefficients in the basic subsystem identification algorithm and feedback subsystem coefficients in the extended version	[-]
$\Gamma_j(s)$	$[s^j \quad s^{j-1} \quad \cdots \quad s \quad 1]^T$	[-]
γ	Discrete feedback time delay	[-]
ζ_{nms}	Damping ratio of the neuromuscular system	[-]
λ	Eigenvalue	[-]
σ_u^2	Variance of the control output u	[inch^2]
$\sigma_{u_n}^2$	Variance of the control output u due to the human remnant n	[inch^2]
τ_f	Far-viewpoint delay term	[s]
τ_n	Near-viewpoint delay term	[s]
τ_p	Preview time	[s]
τ_v	Output response delay term	[s]
ϕ_k	Phase of a sine function	[rad]
ψ	Discrete feedforward time delay	[-]
ω_{base}	Base frequency of a time signal	[rad/s]
ω_d	Disturbance signal frequencies	[rad/s]
$\omega_{e^\star}$	Compensatory cutoff frequency	[rad/s]
ω_f	Pre-filter cutoff frequency	[rad/s]
ω_k	Frequency of a multi-sine signal	[rad/s]
ω_{nms}	Natural frequency of the neuromuscular system	[rad/s]
ω_t	Target signal frequencies	[rad/s]
A_k	Amplitude of a sine function	[inch]
$\mathbb{C}$	Set of complex numbers	[-]
c	Complex number	[-]
$\tilde{\mathcal{D}}(s,\phi)$	Parameterized denominator of the closed loop system without time delays	[-]
d_f	Discrete far-viewpoint feedforward time delay	[-]

Symbol	Description	Unit
d_v	Discrete feedback time delay	[-]
$D_{CE,d}(z)$	Discrete-time denominator of the controlled element dynamics	[-]
$D_{CE}(s)$	Denominator of the controlled element dynamics	[-]
$D_{o_p}(s)$ or D_p	Denominator of the prefilter dynamics (or abbreviation)	[-]
$D_{o_t,d}(z)$ or $D_{t,d}$	Discrete-time denominator of the target signal response dynamics (or abbreviation)	[-]
$D_{o_t}(s)$ or D_t	Denominator of the target signal response dynamics (or abbreviation)	[-]
$\mathcal{D}_{o_t,d}(z,\beta)$	Discrete-time parameterized denominator of the target signal response dynamics	[-]
$\mathcal{D}_{o_t}(s,\phi)$	Parameterized denominator of the target signal response dynamics	[-]
$D_{o_x}(s)$ or D_x	Denominator of the output signal response dynamics	[-]
$D_{o_x,d}(z)$ or $D_{x,d}$	Discrete-time denominator of the output signal response dynamics	[-]
$\mathcal{D}_{o_x}(s,\phi)$	Parameterized denominator of the output signal response dynamics	[-]
$\mathcal{D}_{o_x,d}(s,\phi)$	Discrete-time parameterized denominator of the output signal response dynamics	[-]
$e(t)$	Error signal	[inch]
$e^{\star}(t)$	Modified error/future error	[inch]
$f_t(t)$	Target signal	[inch]
$F_t\ (j\omega)$	Fourier Transform of the target signal	[inch]
$f_d(t)$	Disturbance signal	[inch]
$H_{CE}(s)$	Controlled element dynamics	[-]
$H_{cl}^{TX}(s)$	Closed-loop transfer function for a TX model structure	[-]
$H_{cl,d}^{TX}(z)$	Discrete time closed loop transfer function for a TX model structure	[-]
$\tilde{H}_{cl}^{TX}(s,\phi,\beta)$	Parameterized closed loop transfer function for a TX model structure without time delays	[-]
$\tilde{H}_{cl,d}^{TX}(z,\alpha,\psi,\beta,\gamma)$	Discrete time parameterized closed loop transfer function for a TX model structure with time delays	[-]
$H_{o_e}(j\omega)$ or H_e	Human operator error signal response dynamics (or abbreviation)	[-]
$H_{o_f}(j\omega)$ or H_f	Human operator far-viewpoint subsystem dynamics (or abbreviation)	[-]
$H_{o_n}(j\omega)$ or H_n	Human operator near-viewpoint subsystem dynamics (or abbreviation)	[-]
$H_{o_p}(j\omega)$	Human operator pre-filter dynamics	[-]
$H_{o_t}(j\omega)$ or H_t	Human operator target signal subsystem dynamics (or abbreviation)	[-]
$H_{o_x}(j\omega)$ or H_x	Human operator system output dynamics (or abbreviation)	[-]
$H_r(j\omega)$	Remnant dynamics	[-]
I	Identity matrix	[-]
K_{CE}	Controlled element dynamics gain	[-]
K_f	Far-viewpoint gain	[-]
K_n	Near-viewpoint gain	[-]

Symbol	Description	Unit
$K_{e^\star}$	Future error gain	[-]
K_r	Remnant gain	[-]
N	Number of samples of the time signal used to obtain the DFT	[-]
$\mathbb{N}$	Set of natural numbers	[-]
$n(t)$	Remnant signal	[inch]
$N_{CE,d}(z)$	Discrete-time numerator of the controlled element dynamics	[-]
$N_{CE}(s)$	Numerator of the controlled element dynamics	[-]
$N_{o_p}(s)$ or N_p	Numerator of the pre-filter subsystem dynamics (or abbreviation)	[-]
$N_{o_t}(s)$ or N_t	Numerator of the target signal subsystem dynamics (or abbreviation)	[-]
$N_{o_t,d}(z)$ or $N_{t,d}$	Discrete-time numerator of the target signal subsystem dynamics (or abbreviation)	[-]
$\mathcal{N}_{o_t}(s,\beta)$	Parameterized numerator of the target signal subsystem dynamics	[-]
$\mathcal{N}_{o_t,d}(z,\alpha)$	Discrete-time parameterized numerator of the target signal subsystem dynamics	[-]
$\tilde{\mathcal{N}}_{o_t}(s,\phi)$	Parameterized numerator of the closed loop system for the algorithm without time delays	[-]
$N_{o_x}(s)$ or N_x	Numerator of the output signal response subsystem dynamics	[-]
$N_{o_x,d}(z)$ or $N_{x,d}$	Discrete-time numerator of the output signal response subsystem dynamics	[-]
$\mathcal{N}_{o_x,d}(s,\phi)$	Discrete-time parameterized numerator of the output signal response subsystem dynamics	[-]
$\mathcal{N}_{o_x}(s,\phi)$	Parameterized numerator of the output signal response subsystem dynamics	[-]
n_{o_t}	Degree of the numerator of the target signal subsystem	[-]
$\mathbb{R}$	Set of real numbers	[-]
$S_{uu}(j\omega)$	Power spectral density of the control output signal	[inch^2/Hz]
$S_{uu_d}(j\omega)$	Power spectral density of the control output signal due to the disturbance function f_d	[inch^2/Hz]
$S_{uu_n}(j\omega)$	Power spectral density of the control output signal due to the remnant n	[inch^2/Hz]
$S_{uu_t}(j\omega)$	Power spectral density of the control output signal due to the target function f_t	[inch^2/Hz]
T	Measurement period of the time signal used for the DFT	[s]
$T_{L,e^\star}$	Future error lead-time constant	[s]
$T_{l,e^\star}$	Future error lag-time constant	[s]
$T_{l,f}$	Far-viewpoint lag-time constant	[s]
$T_{l,n}$	Near-viewpoint lag-time constant	[s]
$T_{l,r}$	Remnant lag-time constant	[s]
T_s	Sampling time	[s]
t	Time	[s]
$u(t)$	Human control output signal	[inch]
$U\,(j\omega)$	Fourier Transform of the control output	[inch]
$x(t)$	System output signal	[inch]
$X\,(j\omega)$	Fourier Transform of the system output signal	[inch]
$\mathbb{Z}$	Set of integers	[-]

SSID specific

This report provides an extensive description of different versions of the newly developed subsystem identification technique and these different versions use different sign conventions. In the chapters describing/explaining subsystem identification (Chapters 3 and 4), the notations from the original papers have been adopted. However, as these different papers used different notations, the number of symbols became large. Therefore, a separate list of symbols has been made for the symbols occurring only in Chapters 3 and 4, the chapters elaborating on subsystem identification.
Within this list of symbols dedicated to subsystem identification, a division is made between parameters that are generally used in the context of subsystem identification and the parameters that solely involve the two versions of the technique (the basic and the extended algorithm) that are treated extensively.

General parameters

Symbol	Description	Unit
γ_r	Noise input at r, according to Zhang and Hoagg (2016b)	[inch]
γ_e	Noise input at e, according to Zhang and Hoagg (2016b)	[inch]
γ_u	Noise input at u, according to Zhang and Hoagg (2016b)	[inch]
γ_y	Noise input at y, according to Zhang and Hoagg (2016b)	[inch]
$\eta(z)$	Nonlinear function in Koushkbaghi et al. (2018)	[-]
ω_k	Evaluated frequencies	[rad/s]
Π	All possible options of the (feedback) candidate pool entries provided the ranges per coefficient	[-]
ρ	Stability margin for the stability domain $\mathcal{S}$	[-]
$\sigma^2_{u_{\sigma_r}}$	Contribution of the noise σ_r inserted at the input to the control output u	[inch^2]
Φ	(Feedback) candidate pool Φ $(= \Pi \cap \mathcal{S}/\mathcal{S}_\rho)$	[-]
ϕ_i	Entry of the candidate pool	[-]
d	Discrete feedback delay term (as defined by Zhang et al. (2018))	[-]
$G(z)$	Controlled element dynamics according to Zhang et al. (2018); Koushkbaghi et al. (2018)	[-]
$\tilde{G}(z)$	Closed loop transfer function (Zhang and Hoagg, 2016b)	[-]
$G_d(z)$	Controlled element dynamics Sheffler et al. (2019)	[-]
$G_{ff}(z)$	Feedforward subsystem dynamics	[-]
$G_{ff,p}(z)$	Feedforward subsystem dynamics of order p according to Koushkbaghi et al. (2018)	[-]
$G_{fb}(z)$	Feedback subsystem dynamics	[-]
$G_v(z)$	Controlled element output fed back into the system (Zhang and Hoagg, 2016b)	[-]
G_{ur}	Transfer function relating the input r to the control output u	[-]
$G_y(z)$	Controlled element output (Zhang and Hoagg, 2016b)	[-]
$\tilde{G}_{yr}(z)$	Closed loop transfer function according to Zhang et al. (2018); Sheffler et al. (2019)	[-]
$\tilde{H}(H_{o_t}, H_{o_e}, \omega_k)$	Closed loop transfer function of the human in the loop system	[-]
$H(\omega_k)$	Closed-loop frequency response data	[-]
J	Cost function	[-]
k	Evaluated frequency index ($\in [1, 2, \dots, N]$)	[-]

Symbol	Description	Unit
M	Total number of (feedback) candidate pool entries	[-]
$\mathcal{M}$	$\{1, 2, \dots, M\}$	[-]
N	The number of evaluated frequencies (not related to the DFT)	[-]
$\mathcal{N}$	$\{1, 2, \dots, N\}$	[-]
$N_{samples}$	Number of samples of the time signal used for the DFT	[-]
$\mathcal{R}$	Candidate pool coefficient range	[-]
R_i	Noise to signal ratio corresponding to noise intensity 4^{-i}	[-]
$r(j\omega)$	Target signal	[inch]
$\mathcal{S}$	Domain for which the candidate pool entries ϕ_i lead to a stable closed-loop system	[-]
$\mathcal{S}_\rho$	Domain for which the candidate pool entries ϕ_i lead to a stable closed-loop system with margin ρ	[-]
$S_{rr}(j\omega)$	Power spectral density of the input signal	[inch^2/Hz]
$S_{uu_{\sigma_r}}(j\omega)$	Power spectral density of the control output signal due to the noise at the input signal σ_r	[inch^2/Hz]
$S_{uu_r}(j\omega)$	Power spectral density of the control output signal due to the input signal r	[inch^2/Hz]
$y(j\omega)$	System output	[inch]

Basic algorithm

The symbols treated in this part use the sign convention corresponding to the basic subsystem identification technique as used by Zhang and Hoagg (2016a) and correspond to Sections 3.2 and 4.3.

Symbol	Description	Unit
β	Feedforward numerator coefficients	[-]
β_*	True coefficients of the feedforward numerator	[-]
β^+	Identified feedforward numerator coefficients	[-]
$\beta_i(\phi_i)$	Global minimizer of the cost function for ϕ_i	[-]
$\sigma_r(j\omega)$	Fourier Transform of the noise at the input	[inch]
$\sigma_y(j\omega)$	Fourier Transform of the noise at the output	[inch]
ω_k	Evaluated frequencies	[rad/s]
ϕ	Feedforward denominator and feedback subsystem coefficients	[-]
ϕ^+	Identified feedforward denominator and feedback subsystem coefficients	[-]
ϕ_*	True coefficients of the feedforward denominator and the feedback subsystem	[-]
$\Omega_0(\phi)$	Term of the cost function that corresponds to β to the 0th order	[-]
$\Omega_1(\phi)$	Term of the cost function that corresponds to β to the first order	[-]
$\Omega_2(\phi)$	Term of the cost function that corresponds to β to the second order	[-]
$\tilde{\mathcal{D}}(s,\phi)$	Parameterized closed-loop denominator	[-]
d	Number of coefficients in a candidate pool entry.	[-]
$D_0(s)$	Denominator of the controlled element dynamics subsystem	[-]
$D_1(s)$	Denominator of the feedforward subsystem	[-]
$\mathcal{D}_1(s,\phi)$	Parameterized feedforward denominator	[-]
$D_2(s)$	Denominator of the feedback subsystem	[-]
$\mathcal{D}_2(s,\phi)$	Parameterized feedback denominator	[-]
d_0	Degree of controlled element subsystem denominator	[-]
d_1	Degree of the feedforward subsystem denominator	[-]
d_2	Degree of feedback subsystem denominator	[-]
E_1	Matrix to convert the candidate pool entries to $\mathcal{D}_1$	[-]
E_2	Matrix to convert the candidate pool entries to $\mathcal{N}_2$	[-]
E_3	Matrix to convert the candidate pool entries to $\mathcal{D}_2$	[-]
$\tilde{G}(s)$	Closed-loop transfer function	[-]
$\tilde{\mathcal{G}}(s,\beta,\phi)$	Parameterized closed loop transfer function	[-]
$G_0(s)$	Controlled element subsystem dynamics	[-]
$G_1(s)$	Feedforward subsystem dynamics	[-]
$\mathcal{G}_1(s,\beta,\phi)$	Parameterized feedforward subsystem	[-]
$G_1^+(s)$	Identified feedforward subsystem	[-]
$G_2(s)$	Feedback subsystem dynamics	[-]
$\mathcal{G}_2(s,\phi)$	Parameterized feedback subsystem	[-]
$G_2^+(s)$	Identified feedback subsystem	[-]
$\mathcal{J}_i(\beta)$	Cost function corresponding to the candidate pool entry ϕ_i	[-]
$J(\beta,\phi)$	Cost function in terms of the coefficients of β and ϕ	[-]
$\mathcal{J}_\ell$	Minimum value of the cost function over all entries of the candidate pool Φ	[-]

Symbol	Description	Unit
l	Index corresponding to the minimum value of the cost function for all options in both candidate pools.	[-]
$N_0(s)$	Numerator of the controlled element dynamics subsystem	[-]
$N_1(s)$	Numerator of the feedforward subsystem	[-]
$\mathcal{N}_1(s,\phi)$	Parameterized numerator of the feedforward subsystem	[-]
$\tilde{\mathcal{N}}_1(s,\phi)$	Combined numerator part of the closed-loop transfer function multiplied by β	[-]
$N_2(s)$	Numerator of the feedback subsystem	[-]
$\mathcal{N}_2(s,\phi)$	Parameterized feedback numerator	[-]
$\tilde{\mathcal{N}}_2(s,\phi)$	Combined numerator part of the closed-loop transfer function not multiplied by β	[-]
n_0	Degree of the controlled element subsystem numerator	[-]
n_1	Degree of the feedforward numerator	[-]
n_2	Degree of the feedback numerator	[-]

Extended algorithm

The symbols treated in this section use the sign convention corresponding to the basic subsystem identification technique as used by Sheffler et al. (2019) and correspond to Sections 3.4 and 4.4.

Symbol	Description	Unit
α	Feedforward numerator coefficients	[-]
$\alpha_i(\psi, phi_i)$	Global minimizer of the cost function for an entry of the feedback candidate pool ϕ_i	[-]
α^+	Identified feedforward numerator coefficients	[-]
β	Feedback subsystem coefficients	[-]
β^+	Identified feedback subsystem coefficients	[-]
$\Gamma_j(z)$	$[z^j \quad z^{j-1} \quad \cdots \quad z \quad 1]^T$	[-]
γ	Discrete feedback delay	[-]
γ^+	Identified discrete feedback time delay	[-]
$\mathrm{Z}(\psi)$	Vector containing the delay terms of all evaluated frequencies	[-]
σ_k	Z-transformed evaluated frequencies ($\sigma_k = e^{j\omega_k T_s}$)	[-]
τ_{fb}	Discrete feedback time delay	[s]
τ_{ff}	Discrete feedforward time delay	[s]
$\Upsilon_1(\beta, \gamma)$	Vector containing the elements of $\Omega_1(\beta, \gamma)$ not corresponding to α	[-]
$\Upsilon_2(\beta, \gamma)$	Vector containing the elements of $\Omega_1(\beta, \gamma)$ corresponding to α	[-]
Ψ	Feedforward time delay candidate pool	[-]
ψ	Discrete feedforward time delay	[-]
ψ^+	Identified feedforward discrete time delay	[-]
$\Omega_0(\beta, \gamma)$	Term of the cost function that corresponds to α to the 0th order	[-]
$\Omega_1(\beta, \gamma)$	Term of the cost function that corresponds to α to the first order excluding $\mathrm{Z}(\psi)$	[-]
$\hat{\Omega}_1(\psi, \beta, \gamma)$	Term of the cost function that corresponds to α to the first order including $\mathrm{Z}(\psi)$	[-]
$\Omega_2(\beta, \gamma)$	Term of the cost function that corresponds to α to the second order	[-]
$D_d(z)$	Denominator of the controlled element dynamics subsystem	[-]
$\mathcal{D}_{fb}(z, \beta)$	Parameterized feedback denominator	[-]
$\tilde{\mathcal{D}}(z, \beta, \gamma)$	Closed-loop system denominator	
d	Number of coefficients in a feedback candidate pool entry.	[-]
d_{fb}	Degree of feedback denominator	[-]
d_{ff}	Degree of feedforward denominator	[-]
E_β	Matrix to convert the feedback candidate pool entries ϕ_i to β	[-]
E_γ	Matrix to convert the feedback candidate pool entries ϕ_i to γ	[-]
$\mathcal{F}_i(\phi_i)$	Simplifying term used for finding the optimal value in the feedforward candidate pool	[-]
$\tilde{\mathcal{G}}(z, \alpha, \psi, \beta, \gamma)$	Parameterized closed-loop transfer function	[-]
$\mathcal{G}_{fb}^+(z, \beta^+, \gamma^+)$	Identified feedback subsystem	[-]
$\mathcal{G}_{ff}(z, \alpha, \beta)$	Parameterized feedforward subsystem	[-]
$\mathcal{J}(\alpha, \psi, \beta, \gamma)$	Parameterized cost function	[-]
$\mathcal{J}_i(\alpha, \psi)$	Cost function for one specific entry of the feedback candidate pool ϕ_i	[-]
ℓ	Index corresponding to the minimum value of the cost function for all options in both candidate pools.	[-]

Symbol	Description	Unit
$N_d(z)$	Numerator of the controlled element dynamics subsystem	[-]
$\mathcal{N}_{fb}(z,\beta)$	Parameterized feedback subsystem numerator	[-]
$\tilde{\mathcal{N}}_{fb}(z,\beta)$	Combined numerator part of the closed-loop transfer function not multiplied by α	[-]
$\mathcal{N}_{ff}(z,\alpha)$	Parameterized feedforward subsystem numerator	[-]
$\tilde{\mathcal{N}}_{ff}(z,\beta,\gamma)$	Combined numerator part of the closed-loop transfer function multiplied by α	[-]
n_{fb}	Degree of feedback numerator	[-]
n_{ff}	Degree of feedforward numerator	[-]
P	$\{1,2,\dots,p\}$	[-]
p	Number of elements in the feedforward time delay candidate pool	[-]
$\mathcal{Q}_i(\psi)$	Modified cost function, minimized for α, for each entry of the feedback candidate pool in terms of the feedforward delay	[-]
$\mathrm{Q}_i\left(\psi_{q_i}\right)$	Minimum value of the modified cost function of all options of the feedforward time delay in $\mathcal{P}$ for an entry of the feedback candidate pool ϕ_i	[-]
$\mathrm{Q}_\ell\left(\psi_{q_\ell}\right)$	Minimum value of the modified cost function for all options of the feedback candidate pool.	[-]
ψ_{q_i}	The value of the feedforward time delay for which $\mathcal{Q}_i(\psi)$ is minimal	[-]

I

Paper I: On the Application of a Subsystem Identification Technique to a Manual Control Pursuit- and Preview Model

On the Application of a Subsystem Identification Technique to a Manual Control Pursuit- and Preview Model

Authour: Pieter-Bas J. C. Bentinck
Supervisors: Daan M. Pool, Kasper van der El, Max Mulder
Control & Simulation Section, Faculty of Aerospace Engineering,
Delft University of Technology, Delft, The Netherlands

In the process of creating a unifying understanding of human control behaviour, an important area of improvement involves anticipative feedforward behaviour. This paper combines two recent contributions in this area — a versatile model for pursuit- and preview tracking tasks and a newly developed subsystem identification (SSID) technique that uses a candidate pool approach to identify manual control behaviour — to evaluate the performance of the latter. Through simulations of a preview task with two types of controlled element dynamics, the performance of the technique with respect to different noise levels and candidate pool densities is addressed. Although, the paper demonstrates how the method is successfully applied to the preview model, it is observed that the performance of SSID starts deteriorating increasingly over the range of noise levels found in real human-in-the-loop experiments. Furthermore, for some experiment conditions, problems appear regarding the identifiability of the system.

I. Introduction

An essential part of human control behavior is the human's ability to look ahead and to anticipate future developments. This type of *feedforward* behavior is seen in curve driving or tracking tasks with predictable signals. Within manual control cybernetics — the research field that focuses on understanding the interaction between humans and machines — the theory regarding this type of control behavior is unfortunately insufficiently developed [1]. Within the field, the more constrained compensatory control task is well understood through the crossover model [2], accelerating the design of human-machine systems in earlier days [3], [4]. On the other hand, the lack of such knowledge in more complex tracking tasks featuring feedforward control behavior is limiting the optimization of control interfaces [1].

To bridge this gap between theory and technology, and to create a unified understanding of human control behavior, efforts have been made in various areas. Regarding feedforward control behavior, [5] developed a linear time-invariant model that is well suited to explain and predict behavior in pursuit- and preview tracking tasks. The model is based on the crossover model, works well under a wide range of conditions and uses intuitive and identifiable parameters such as the human look-ahead time. Through this, it provides a good understanding of how humans use pursuit- and preview displays.

Another effort regarding feedforward human control behavior is observed in the set of identification techniques developed to establish a proper understanding, with examples provided in [6]–[8]. Here, [9] (and later extended to include estimation of time delays in [10], [11]) developed a new subsystem identification (SSID) technique, which is a frequency-domain technique that allows for estimation of feedback- and feedforward subsystems. The technique uses a candidate pool approach and identifies the systems from experiments involving only a single forcing function.

It is interesting to see if, and if so, how, the SSID technique can be used to estimate the model parameters of the pursuit- and preview model of [5]. Furthermore, despite previous demonstrations of how the technique works with increasing noise levels and candidate pool densities (in e.g., [11]), an unanswered question is how those noise levels relate to the non-linearities observed in real human-in-the-loop experiments, as described by [12], [13].

The aim of this research is to apply the SSID technique from [11], [14] to the pursuit- and preview model as proposed by [5], [15] and to evaluate the performance of the model parameter estimation for different noise levels and candidate pool densities. Offline simulations with previously identified human operator (HO) models will be used. For the simulations, a preview tracking task is evaluated, as it allows for a stronger feedforward component compared to pursuit tracking tasks. Furthermore, two types of controlled element (CE) dynamics will be evaluated, as these affect both the amount of preview and the level of non-linearities (or noise levels) that humans behavior shows..

The paper is structured as follows. First, Section II elaborates on the common model structures for pursuit- and preview tracking tasks and the selected preview model. Then, Section III explains how the SSID technique is applied to this model. Consequently, Section IV elaborates on the setup of the simulations after which Sections V and VI present and discuss the results, respectively. Finally, the conclusions are presented in Section VII.

II. Feedforward Human Control Models

To identify and model human control behavior in pursuit- and preview tracking tasks, different approaches have been taken [4], [16]–[19]. Fig. 1 shows a typical preview display and model structures used to describe the human control behavior. In the tracking task, the human operator (HO) generates control outputs (u) to the controlled element. The goal is to keep the CE (white circle in Fig. 1(a) as close as possible to the target f_t (black circle), and thus minimizing the error e.

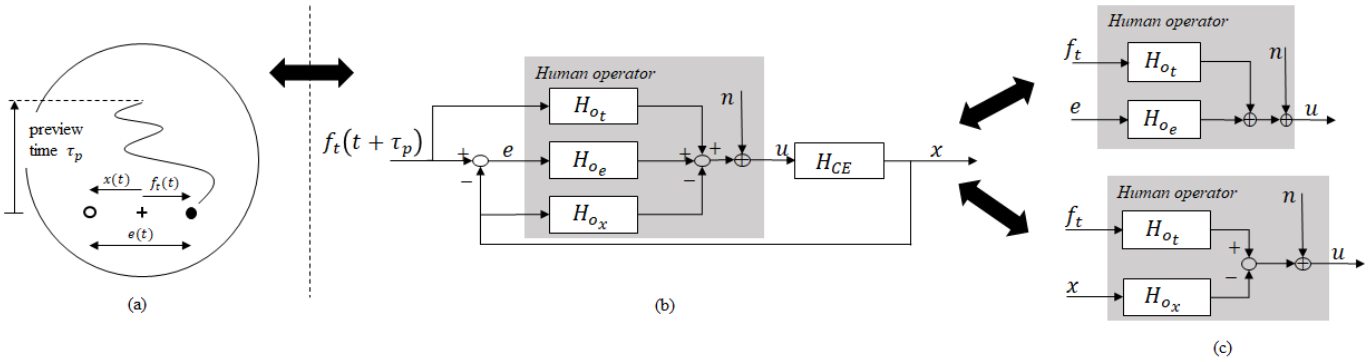


Fig. 1. Overview of a pursuit/preview display (a) and the three-channel block diagram that represents the human in a pursuit/preview tracking task (b). Furthermore, (c) shows the two-channel simplifications used for identification in practice.

The display is said to be a preview display when part of the future trajectory of the target signal is also visible. When this is not the case, (i.e., when $\tau_p = 0$), the task becomes a pursuit tracking task.

Fig. 1(b) shows the most complete model structure to model the HO in such a tracking task, where the HO can respond to either the target (f_t), the error (e) and the CE output (x). However, there are two problems with this setup, when trying to measure the different HO responses. First, due to the linear relation between f_t, e and x, the three-channel response system is overdetermined. Second, for each response that is to be measured, a separate forcing function is required. A realistic tracking task, however, only allows for a maximum of two relevant forcing functions (see [20], [6]).

Due to the issues with a three-channel model structure, the model structure is often reduced to a two-channel model structure, where either the error- or the CE output responses are omitted. These model structures, shown in Fig. 1(c), are then referred to as ET (responding to the error and the target) and TX (target and CE output) models, both of which are equally able to describe HO behavior [21]. In both models, the target response captures any feedforward behavior. In literature, the most commonly used model structure is the ET model structure [18], [19], [21], which allows for a straightforward extension of the widely accepted crossover model for compensatory tracking tasks [2]. Until now, the subsystem identification (SSID) technique investigated in this paper has only been applied to this ET model structure. Recently a model has been developed using the TX model structure at its foundation [5].

A. HO model

This pursuit- and preview model can be rewritten to the form as shown in Fig. 2 (see e.g. [15]). This quasi-linear model is an extension of McRuer's crossover model used for compensatory tracking tasks [2], where instead of the true error, the HO is now controlling an 'internally calculated' (by the HO) error $e^\star$. This internal error is derived from the current CE output and the target a certain τ_f seconds ahead (see also Fig. 1 (a).

The preview filter dynamics, H_{o_p}, consist of the feedforward look-ahead time τ_f (expressed as a 'negative time delay') and a low-pass filter H_{o_f}. Generally, this low-pass filter is expressed

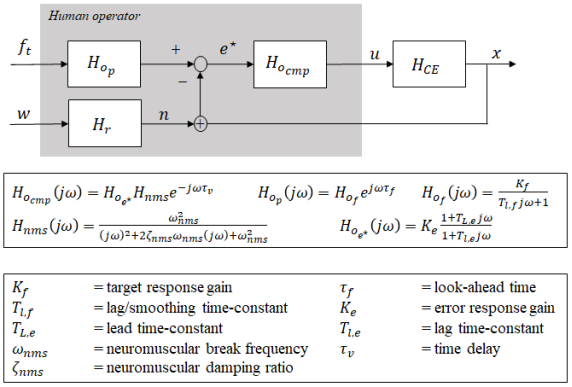


Fig. 2. Overview of the pursuit- and preview model as proposed by [22].

in terms of a feedforward gain K_f and a time constant $T_{l,f}$, while this is being rewritten in terms of a cut-off frequency $\omega_{l,f}$ $(= \frac{1}{T_{l,f}})$ for the SSID application in this paper. Note that rewriting to this form is only possible when considering preview, as for pursuit tracking both τ_f and $T_{l,f}$ equal 0. Then, H_{o_p} only consists of the gain K_f.

The other linear part of the pursuit- and preview model consists of the compensatory dynamics $H_{o_{cmp}}$, following the crossover model. These compensatory dynamics consist of the equalization dynamics $H_{o_{e^*}}$, the neuromuscular dynamics H_{nms} and a time delay τ_v. The equalization dynamics vary with the type of CE dynamics, as the lead-time constant $T_{L,e}$ is only non-zero for double integrator (DI) dynamics, while the lag-time constant is non-zero for gain dynamics only [23].

Besides the linear part of the model, all non-linearities of the HO are modelled by the remnant $n(t)$. As specified in [13] (based on earlier work in [12]), this model inserts the remnant at the system output x, where it enters the closed-loop as low-pass filtered Gaussian white noise. The remnant filter H_r, highlighted in (1), consists of a remnant gain K_r and a remnant lag-time constant $T_{l,r}$. The remnant gain is, together with the white noise intensity, determined by the desired level remnant level in the control output $u(t)$, whereas the time constant varies based on the type of CE dynamics.

$$H_r(j\omega) = \frac{K_r}{1 + T_{l,r}j\omega} \tag{1}$$

III. Subsystem Identification

The parameters of the pursuit- and preview model can be estimated using a Subsystem Identification (SSID) technique. This technique is based on the method described by [14], which in turn built upon earlier efforts described in [9], [10], [24]. The essence of SSID is that it estimates the dynamics of multiple subsystems (i.e., blocks) from a single input-output relation. The technique operates in the frequency domain and uses a candidate pool approach to ensure stability of the estimated system.

The technique in this paper is modified with respect to the version of [14] to allow for estimation of the preview model parameters. The modifications involve the following:

- The model structure. Instead of an ET model structure, a prefiltered compensatory model structure is used.
- The domain of the technique. This is changed from the discrete-time z-domain to the continuous-time $j\omega$-domain.
- The feedforward subsystem modelling. This is modelled as an infinite impulse response (IIR), whilst being a finite impulse response (FIR) in earlier versions. This allows the estimation of $\omega_{l,f}$.
- The coefficients of the closed-loop stability candidate pool. This paper uses the model coefficients to define the candidate pool, whereas this was previously done using the transfer function coefficients.

Besides these modifications, the SSID technique is identical to the one described in [14]. In this section, the technique, including aforementioned modifications, is explained. The method is first summarized in the next subsection.

A. Method Summary

Fig. 3 provides an overview of the SSID algorithm, including the application to the preview model structure. The figure shows how the model parameters described in Section II are translated into two candidate pools. From there, the optimal value for the feedforward time delay τ_f and the optimization parameters in α are being found for each entry of the closed-loop stability candidate pool. This is done by minimizing the cost function $\mathcal{Q}$. Finally, the set of optimal parameters is found after evaluating all entries in the closed-loop stability candidate pool. The optimal parameters can then form the system dynamics H_{o_p} and $H_{o_{cmp}}$.

It is noted that although the SSID algorithm described in this section is applied to a preview tracking task, it is equally capable of estimating the model parameters of a pursuit tracking task. As for the pursuit tracking task the feedforward look-ahead time τ_f and the lag-time constant $T_{l,f}(=1/\omega_{l,f})$ are 0, the model reduces and the pre-filter system H_{o_p} becomes equal to just the gain K_f. In the algorithm overview in Fig. 3, this means that Step 2 simplifies to solving only a single quadratic optimization to solve for α_i (instead of solving one for every entry ψ_j in Ψ). The algorithm then becomes equivalent to the one explained in [10].

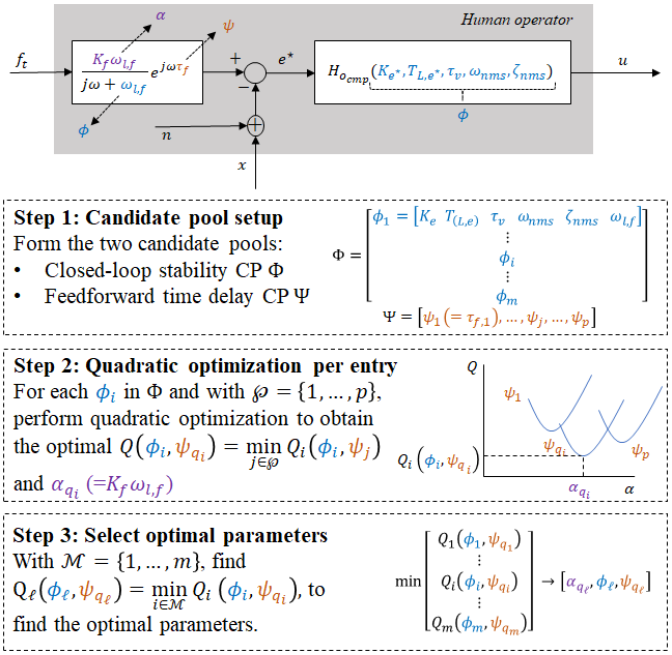


Fig. 3. Schematic overview of the SSID technique.

B. Problem Formulation

As mentioned, SSID estimates multiple subsystems from a single input-output relation. This is the set of closed-loop frequency data defined as $H_{cl}(j\omega) = X(j\omega)/F_t(j\omega)$, where $X(j\omega)$ and $F_t(j\omega)$ are the discrete Fourier Transforms of the target function $f_t(t)$ and the system output $x(t)$, respectively. Looking at Fig. 2, the subsystems that are to be identified are the pre-filter dynamics H_{o_p} and the compensatory dynamics $H_{o_{cmp}}$. Combined with the CE dynamics, these subsystems form the closed-loop dynamics $H_{cl}(j\omega)$ as described:

$$H_{cl}(j\omega) = \frac{H_{o_p}(j\omega) H_{o_{cmp}}(j\omega) H_{CE}(j\omega)}{1 + H_{CE}(j\omega) H_{o_{cmp}}(j\omega)} \tag{2}$$

The goal of the method is to find the parameter set Θ such that the modelled frequency response $H_{cl}^{mod}(j\omega_k)$ approximates the measured frequency response $H_{cl}^{DF}(j\omega_k)$ (where DF represents the describing function). This is achieved by minimizing the cost function given by (3), which has the dependence on $j\omega_k$ omitted in the second part for brevity.

$$\begin{aligned} J(\Theta) &= \sum_{k=1}^{N_f} \left| H_{cl}^{mod}(j\omega_k|\Theta) - H_{cl}^{DF}(j\omega_k) \right|^2 \\ &= \sum_{k=1}^{N_f} \left| \frac{H_{o_p}(\Theta) H_{o_{cmp}}(\Theta) H_{CE}}{1 + H_{CE} H_{o_{cmp}}(\Theta)} - H_{cl}^{DF} \right|^2 \end{aligned} \tag{3}$$

In (3), $H_{cl}^{DF}(j\omega_k) = X(j\omega_k)/F_t(j\omega_k)$, which can be obtained using the discrete Fourier Transforms of the input f_t and the output x. The frequencies ω_k are the frequencies for which the (multi-sine) target signal has power, where $k \in \{1, 2, \dots, N_f\}$. This ensures a high signal-to-noise ratio

(SNR) at the estimation frequencies. The exact definition of the target signal is treated more extensively in Section IV.

The closed-loop dynamics in (2) can also be divided into the numerator and denominator terms, with $H_{o_p}(j\omega) = \frac{N_{o_p}(j\omega)}{D_{o_p}(j\omega)}e^{j\omega\tau_f}$ and $H_{o_{cmp}}(j\omega) = \frac{N_{o_{cmp}}(j\omega)}{D_{o_{cmp}}(j\omega)}e^{-j\omega\tau_v}$ to write the closed-loop equation as (4) (where the dependencies of $(j\omega)$ are omitted for brevity). When comparing those numerators and denominators to the ones described in Fig. 2, it can be seen that $N_{o_p} = N_{o_f}$ and $D_{o_p} = D_{o_f}$, while $N_{o_{cmp}} = N_{o_e}N_{nms}$ and $D_{o_{cmp}} = D_{o_e}D_{nms}$.

$$H_{cl}(j\omega) = \frac{N_{CE}N_{o_{cmp}}N_{o_p}e^{j\omega(\tau_f-\tau_v)}}{D_{o_p}\left(D_{CE}D_{o_{cmp}} + N_{CE}N_{o_{cmp}}e^{-j\omega\tau_v}\right)} \tag{4}$$

To perform SSID, the two subsystems are parameterized by their numerator and denominator coefficients. Let n_p, d_p, n_{cmp} and d_{cmp} be the degrees of the respective numerators and denominators N_{o_p}, D_{o_p}, $N_{o_{cmp}}$ and $D_{o_{cmp}}$ and define $a = n_p + 1$ and $d = d_p + n_{cmp} + d_{cmp} + 1$. Furthermore, both D_{o_p} and $D_{o_{cmp}}$ are monic. Then, define the vectors α, containing the coefficients of N_{o_p}, and β, containing those of D_{o_p}, $N_{o_{cmp}}$ and $D_{o_{cmp}}$, and define the functions $\mathcal{N}_p$, $\mathcal{D}_p$, $\mathcal{N}_{cmp}$ and $\mathcal{D}_{cmp}$, as described in (5), where $\Gamma_j(s) = [s^j \quad s^{j-1} \quad \cdots \quad s \quad 1]$.

$$\begin{aligned}
\mathcal{N}_p(s,\alpha) &= \Gamma_{n_p}\alpha \\
\mathcal{D}_p(s,\beta) &= s^{d_p} + \left[\Gamma_{d_p-1}(s)0_{1\times(1+n_{cmp}+d_{cmp})}\right]\beta, \\
\mathcal{N}_{cmp}(s,\beta) &= \left[0_{1\times d_p} \quad \Gamma_{n_{cmp}}(s) \quad 0_{1\times d_{cmp}}\right]\beta, \\
\mathcal{D}_{cmp}(s,\beta) &= s^{d_{cmp}} + \left[0_{1\times(1+d_p+n_{cmp})} \quad \Gamma_{d_{cmp}-1}(s)\right]\beta
\end{aligned} \tag{5}$$

To apply this technique to the preview model of [5], the coefficients of α and β are expressed in terms of the HO model parameters. This also means that, as the equalization dynamics $H_{o_{e^\star}}$ (see Fig. 2) vary with CE, β will do too. Tables I and II show the different orders of the subsystem transfer functions and the resulting α and β for gain- (G, position control), single integrator- (SI, rate control) and double integrator (DI, acceleration control) CE dynamics and a preview display, respectively. As mentioned, for pursuit control, the time constant $T_{l,f} = 0s$ and the equations simplify upon processing this into the coefficients.

TABLE I
OVERVIEW OF THE ORDERS OF THE SUBSYSTEM NUMERATORS AND DENOMINATORS FOR DIFFERENT CE DYNAMICS.

	n_{o_p}	d_{o_p}	$n_{o_{cmp}}$	$d_{o_{cmp}}$
G	0	1	0	3
SI	0	1	0	2
DI	0	1	1	2

Analogous to α and β, let ψ and γ represent the feedforward look-ahead time τ_f and the feedback time delay τ_v, respectively. Then, the cost function of (3) can be written in terms of α, β, γ and ψ, see (6), where $\tilde{\mathcal{N}}$ is described in (7) and $\tilde{\mathcal{D}}$ by (8).

$$\mathcal{J}(\alpha,\beta,\gamma,\psi) = \sum_{k=1}^{N_f}\left|\frac{e^{j\omega_k\psi}\tilde{\mathcal{N}}(j\omega_k,\beta,\gamma)\alpha}{\tilde{\mathcal{D}}(j\omega_k,\beta,\gamma)} - H_{cl}^{DF}(j\omega_k)\right|^2 \tag{6}$$

$$\tilde{\mathcal{N}}(j\omega_k,\beta,\gamma) = N_{ce}(j\omega_k)\mathcal{N}_{cmp}(j\omega_k,\beta)e^{-j\omega_k\gamma}\Gamma_{n_p}(j\omega_k) \tag{7}$$

$$\begin{aligned}
\tilde{\mathcal{D}}(j\omega_k,\beta,\gamma) = \mathcal{D}_p(j\omega_k,\beta)(D_{ce}(j\omega_k)\mathcal{D}_{cmp}(j\omega_k,\beta)+ \\
N_{ce}(j\omega_k)\mathcal{N}_{cmp}(j\omega_k,\beta)e^{-j\omega_k\gamma})
\end{aligned} \tag{8}$$

C. Candidate Pool Setup

To perform SSID, two candidate pools are required. First, the *closed-loop stability candidate pool* Φ, which consists of the model parameters from the preview model that occur in β and γ (i.e., the parameters that affect $\tilde{\mathcal{D}}$). Table II shows the preview model parameters that form β for different types of CE dynamics. Furthermore, the feedback time delay γ is always represented by τ_v.

This closed-loop stability candidate pool assures stability by considering the set of $(\beta,\gamma) \in \mathbb{R}^d_{>0}$ that lies within $\mathcal{S} = \left\{(\beta,\gamma) \in \mathbb{R}^d_{>0} : \text{if } \widetilde{\mathcal{D}}(\lambda,\beta,\gamma) = 0, \text{ then } \operatorname{Re}\lambda < 0\right\}$ (i.e., the set of parameters (β,γ) for which the closed-loop system is stable). The candidate pool Φ has M elements ($\Phi = \{\phi_1,\ldots,\phi_M\}$), which are all unique. Furthermore, let $E_\beta = [I_d \quad 0_{d\times 1}]$ and $E_\gamma = [0_{1\times d} \quad 1]$ so that $\beta_i = E_\beta\phi_i$ and $\gamma_i = E_\gamma\phi_i$.

Note that when applying SSID to the preview model described in Section II, β and γ are functions of the model parameters ($K_{e^\star}$ etc.) and in practice, the closed-loop stability candidate pool is formed based on the model parameters that compose β and γ. Besides a translating step, however, this does not add any complexity to the algorithm.

Besides the closed-loop stability candidate pool Φ, SSID requires a second candidate pool: the *feedforward-delay candidate pool* $\Psi \in \mathbb{R}_{>0}$, which consists of the parameterized look-ahead time ψ. It consists of p elements ($\ominus = \psi_1, \psi_2, \ldots, \psi_p$) and all elements ψ_j are unique.

D. Algorithm Operation

With the candidate pools defined and the cost function rewritten in terms of (9), the parameter estimation is performed in the following steps. Working out the squared term in (6) results in (9), with Eqs. (10) to (12) describing the terms Ω_0, Ω_1 and Ω_2 and Eqs. (13) to (15) explaining Υ_1, Υ_2 and Z, respectively. It is noted how, compared to the ET algorithms used in e.g., [11], [14], the Ω_1 term is simplified, as it is independent of β and γ.

$$\begin{aligned}
\mathcal{J}(\alpha,\beta,\gamma,\psi) = \alpha^T\Omega_2(\beta,\gamma)\alpha + \Omega_0(\beta,\gamma)+ \\
2\operatorname{Re} Z^T(\psi)\Omega_1^T(\beta,\gamma)\alpha
\end{aligned} \tag{9}$$

TABLE II
OVERVIEW OF THE PARAMETER VECTORS α AND β FOR DIFFERENT CE DYNAMICS.

	α	β
G	$\omega_{l,f}K_f$	$[\omega_{l,f} \quad \omega_{l,e}\omega_{nms}^2 K_e \quad \omega_{l,e}+2\zeta_{nms}\omega_{nms} \quad 2\zeta_{nms}\omega_{nms}\omega_{l,e}+\omega_{nms}^2 \quad \omega_{nms}^2]$
SI	$\omega_{l,f}K_f$	$[\omega_{l,f} \quad \omega_{nms}^2 K_e \quad 2\zeta_{nms}\omega_{nms} \quad \omega_{nms}^2]$
DI	$\omega_{l,f}K_f$	$[\omega_{l,f} \quad \omega_{nms}^2 K_e T_{L,e} \quad \omega_{nms}^2 K_e \omega_{nms} \quad 2\zeta_{nms}\omega_{nms} \quad \omega_{nms}^2]$

$$\Omega_0 = \sum_{k=1}^{N_f} \left| -H\left(\omega_k\right)\right|^2 \in \mathbb{R} \tag{10}$$

$$\Omega_1(\beta,\gamma) = \Upsilon_2^{\mathrm{T}}(\beta,\gamma)\left(\operatorname{diag}\Upsilon_1\right)^* \in \mathbb{C}^{(a)\times N_f} \tag{11}$$

$$\Omega_2(\beta,\gamma) = \operatorname{Re}\sum_{k=1}^{N}\left[\frac{\tilde{\mathcal{N}}\left(j\omega_k,\beta,\gamma\right)}{\tilde{\mathcal{D}}\left(j\omega_k,\beta,\gamma\right)}\right]^* \left[\frac{\tilde{\mathcal{N}}\left(j\omega_k,\beta,\gamma\right)}{\tilde{\mathcal{D}}(j\omega_k,\beta,\gamma)}\right] \in \mathbb{R}^{a\times a} \tag{12}$$

$$\Upsilon_1 = \left[-H_{cl}^{DF}\left(\omega_1\right) \quad \cdots \quad -H_{cl}^{DF}\left(\omega_{N_f}\right)\right]^{\mathrm{T}} \in \mathbb{C}^{N_f} \tag{13}$$

$$\Upsilon_2(\beta,\gamma) = \left[\frac{\tilde{\mathcal{N}}\left(j\omega_1,\beta,\gamma\right)}{\tilde{\mathcal{D}}\left(j\omega_1,\beta,\gamma\right)} \quad \cdots \quad \frac{\tilde{\mathcal{N}}\left(j\omega_{N_f},\beta,\gamma\right)}{\tilde{\mathcal{D}}\left(j\omega_{N_f},\beta,\gamma\right)}^{\mathrm{T}}\right] \in \mathbb{C}^{N_f\times(a)} \tag{14}$$

$$Z(\psi) = \left[e^{j\omega_1\psi}\cdots e^{j\omega_{N_f}\psi}\right]^{\mathrm{T}} \in \mathbb{C}^{N_f} \tag{15}$$

After, substituting β_i and γ_i (converted from the entry ϕ_i of the closed-loop stability candidate pool) into the cost function, it can be seen that the cost function $\mathcal{J}_i$ is convex in α. Furthermore, when using sufficient frequency response data points (i.e., N_f is sufficiently large), it can be shown that Ω_2 is positive definite (see also [9], [24]). From there, a unique global minimizer for α can be found by setting $\frac{\partial \mathcal{J}}{\partial \alpha} = 0$ and solving for α to get (16). This is the optimal value for α given a specific ϕ_i, and is a function of ψ.

$$\alpha_i(\psi) \equiv -\frac{1}{2}\Omega_2^{-1}\left(\beta_i,\gamma_i\right)\operatorname{Re}\Omega_1\left(\beta_i,\gamma_i\right)\Gamma(\psi) \tag{16}$$

Next, substituting (16) in (9) results in the auxiliary cost function $\mathcal{Q}_i$, as defined in (17) with (18) describing $\mathcal{F}_i$.

$$\begin{aligned}\mathcal{Q}_i(\psi) &= \mathcal{J}_i\left(\alpha_i(\psi),\psi\right)\\ &= \Omega_0\left(\beta_i,\gamma_i\right) - \begin{bmatrix}\operatorname{Re}\Gamma(\psi)\\ -\operatorname{Im}\Gamma(\psi)\end{bmatrix}^{\mathrm{T}}\mathcal{F}_i\begin{bmatrix}\operatorname{Re}\Gamma(\psi)\\ -\operatorname{Im}\Gamma(\psi)\end{bmatrix}\end{aligned} \tag{17}$$

$$\mathcal{F}_i = \frac{1}{4}\begin{bmatrix}\operatorname{Re}\Omega_1^{\mathrm{T}}\left(\beta_i,\gamma_i\right)\\ \operatorname{Im}\Omega_1^{\mathrm{T}}\left(\beta_i,\gamma_i\right)\end{bmatrix}\Omega_2^{-1}\left(\beta_i,\gamma_i\right)\begin{bmatrix}\operatorname{Re}\Omega_1^{\mathrm{T}}\left(\beta_i,\gamma_i\right)\\ \operatorname{Im}\Omega_1^{\mathrm{T}}\left(\beta_i,\gamma_i\right)\end{bmatrix}^{\mathrm{T}} \tag{18}$$

Then, to identify the parameters, the feedforward-delay candidate pool $\Psi(=\psi_1,\ldots,\psi_p)$ is used. For each ϕ_i, calculate the minimum value of $\mathcal{Q}_i$, i.e. find ψ_{q_i} with $\mathcal{Q}_i\left(\psi_{q_i}\right) = \min_{j\in\mathcal{P}}\mathcal{Q}_i\left(\psi_j\right)$. Next, find the minimum value of $\mathcal{Q}_i$ for all possible ϕ_i with $\mathcal{Q}_\ell\left(\psi_{q_\ell}\right) = \min_{i\in\mathcal{M}}\mathcal{Q}_i\left(\psi_{q_i}\right)$. Then the identified parameters are $\alpha^+ = \alpha_\ell(\psi_{q_\ell})$, $\psi^+ = \psi_{q_\ell}$, $\beta+ = \beta_\ell$ and $\gamma^+ = \gamma_\ell$. These are easily converted to the model parameters of the preview model.

It is noted that, compared to the original algorithm, an additional form of filtering can be applied on the results. As all model parameters of the model described in Section II are positive, the identified value for α has to be positive too. Then, with respect to the previous paragraph, ψ_{q_i} is found using $\mathcal{Q}_i\left(\psi_{q_i}\right) = \min_{j\in\mathcal{P}}\mathcal{Q}_i\left(\psi_j \| \alpha_j > 0\right)$. For this research, however, this step is not yet implemented.

A last remark is made regarding the parameter α, which is found by performing a quadratic minimization (as seen in the result in (16)). As the parameter $\omega_{l,f}$ occurs in both α and β (see Table II), it might seem odd that it is both set in advance (when forming the CPs) and solved for in the quadratic optimization. However, as α is the product of K_f and $\omega_{l,f}$, this problem is resolved as in fact only K_f is quadratically solved (during a final conversion step).

IV. METHOD: OFFLINE SIMULATIONS

To illustrate the properties of the SSID algorithm, a set of offline simulations of a preview tracking task is performed. The simulations show how the algorithm is able to estimate the (known) HO model parameters using different candidate pool densities and for different noise levels. For the simulations, a preview tracking task with a preview time of 2s is evaluated, matching the experiment of [22]. This is done for both SI and DI CE dynamics. In this section, the setup of the simulations is explained. This consists of the forcing functions, the HO model parameters, the noise model, the candidate pool setup and the general approach for the simulations.

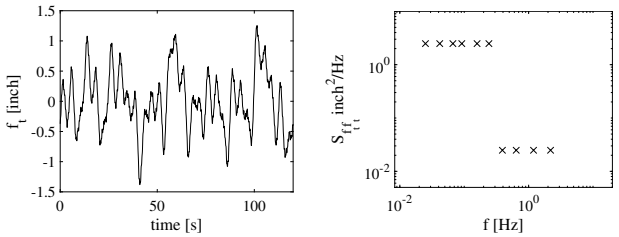
A. Forcing Function Definition

The target signal used for the simulations is a quasi-random appearing multi-sine as described by (19). In the equation, N_f represents the number of frequencies at which f_t has power. Furthermore, A_k, ω_k and ϕ_k represent the amplitude, frequency and phase of the kth sinusoid, respectively.

$$f_t(t) = \sum_{k=1}^{N_f} A_k \sin(\omega_k t + \phi_k) \tag{19}$$

For the simulations, signals from a previous preview experiment were used [22], which in turn were based on [23].

For these signals, $N_f = 10$, the signal bandwidth ω_i, the frequency until which the signal has significant power, is 1.5 rad/s and the standard deviation $\sigma_{f_t} = 0.5$ inch. Fig. 4 shows the time trace and the spectra of the target signal. Note that the amplitude is constant up until the bandwidth frequency, after which it is attenuated by a factor of 10. Furthermore, the frequencies used for the target signal were integer multiples of the base frequency $\omega_b = 0.0524$ rad/s (corresponding to a measurement time T of 120s). For the simulations, different phase realizations were used, selected to ensure a randomly appearing signal and to avoid high- and low crest factors [25].



(a) (b)

Fig. 4. Part of the time trace (a) and spectrum (b) of one of the forcing functions used for the simulations (other realizations had different phase sets).

B. Human Operator Model Parameters

For the simulations, HO models from preview tracking tasks with single- and double integrator CE dynamics and a preview time $\tau_p = 2s$ are used. Table III shows the HO model parameters, which are selected from earlier experiments in [15], [22], which used forcing functions with the same bandwidth and variance as used for these simulations.

TABLE III
OVERVIEW OF THE MODEL PARAMETERS USED IN THE SIMULATIONS.

CE	H_{ce} [−]	$K_{e^\star}$ [-]	$T_{L,e^\star}$ [s]	τ_v [s]	ω_{nms} [rad/s]	ζ_{nms} [-]	K_f [-]	τ_f [s]	$T_{l,f}$ [s]
SI	$\frac{1.5}{(j\omega)}$	1.25	–	0.23	13.0	0.13	1.0	0.60	0.15
DI	$\frac{5}{(j\omega)^2}$	0.33	1.27	0.33	7.4	0.13	0.95	0.95	1.15

C. Remnant Modelling

To simulate the non-linear contributions of the HO as well as time variations and perception- and motor noise, the remnant is simulated as low-pass filtered Gaussian white noise, inserted at the system output (see Section II). When assessing the parameters of the remnant model in (1), the remnant time constant $T_{l,r}$ for DI dynamics is set to 12 s and for SI dynamics to 0.2 s. This is based on the observations in [13], which states that this break frequency varies with CE dynamics and is typically between $1 - 10$ rad/s for SI- and below 0.1 rad/s for DI CE dynamics.

The remnant gain K_f is determined iteratively based on the relative remnant level present in the control output $\frac{\sigma^2_{u_n}}{\sigma^2_u}$. The ratio of this value in real experiments varies most dominantly with CE dynamics and forcing function bandwidth and is observed to range between 0.1 and 0.9, with (for a bandwidth of 1.5 rad/s) an average value of about 0.35 and 0.6 for tasks with SI and DI CE dynamics, respectively.

In this paper, the performance of SSID will be evaluated for a set of noise ratios (NRs), equal to $\frac{\sigma^2_{u_n}}{\sigma^2_u} = 0.0, 0.25, 0.5$ and 0.75.

D. SSID Candidate Pool Setup

To evaluate the performance of SSID and to select a suitable configuration, several candidate pools with varying densities are constructed. The candidate pools are constructed based on the parameters of the HO preview model and converted to be used in SSID. The limits of the parameters are selected based on the results of [15], [22], [26]. Table IV shows the selected limits, which are combined for both SI and DI dynamics. Then, for each model parameter θ_i (i.e., K_e, $T_{L,e}$ etc.), a vector $\vec{\theta}_{i,d_{CP}}$ is constructed based on the limits of the specific parameter, the density number d_{CP} and the initial amount of steps $n_{\theta,1}$. Eqs. (20) and (21) illustrate this mathematically. The initial amount of steps n_{θ_i} is shown in the lower part of Table IV and is selected based on a desired step size for $d_{CP} = 5$. The decision to use one set of parameter limits for both SI- and DI tracking tasks, causes the fact that some parameters (e.g. K_e and ω_{nms}) have a larger value for $n_{\theta,1}$.

$$n_{\theta_i,d_{CP}} = n_{\theta,1} d_{CP} \tag{20}$$

$$\vec{\theta}_{i,d_{CP}} = \theta_{i,min} + \frac{\theta_{i,max} - \theta_{i,min}}{n_{\theta_i,d_{CP}} - 1} \begin{bmatrix} 0 \\ 1 \\ \vdots \\ n_{\theta_i,d_{CP}} - 1 \end{bmatrix} \tag{21}$$

TABLE IV
OVERVIEW OF THE MODEL PARAMETER LIMITS AND THE INITIAL AMOUNT OF STEPS USED IN THE CANDIDATE POOLS.

	K_e [-]	$T_{L,e}$ [s]	τ_v [s]	ω_{nms} [rad/s]	ζ_{nms} [-]	K_f [-]	τ_f [s]	$T_{l,f}$ [s]
max	2	3	0.4	16	1	-	2	1.6
min	0.1	0.6	0.1	4	0.05	-	0.3	0
$n_{\theta,1}$	6	3	3	4	2	-	3	2

The vectors then form the unfiltered candidate pool by combining all possibilities of the θ_i, i.e., K_e, $T_{L,e}$, τ_v, ω_{nms}, ζ_{nms} and $T_{l,f}$. Note that τ_f is not added here, as its vector does not appear in the closed-loop stability candidate pool Φ. This candidate pool Φ is then finally constructed by filtering the unstable entries, as was explained in Section III-C. For this research, this is done by looking at the phase margins of the open-loop function.

E. Approach

To illustrate the working of SSID under different conditions, a number of simulations is performed. For these simulations, noise levels and candidate pool densities are varied to evaluate the settings for which SSID performs well. This is done for both SI and DI CE dynamics. For the candidate pools,

candidate pools with $d_{CP} = 1-5, 7, 10$ and 12 are evaluated. For each combination of candidate pool density and noise level, 40 realizations are performed (except for noiseless data, where 1 realization is possible). One realization consists of the frequency-averaged results from 5 different phase realizations of a 120s simulation, analogous to experiments performed in [22].

V. Results

This section elaborates on the results obtained through the described simulations. Sequentially, the optimal cost function value, the parameter fits, the identified frequency-domain dynamics and the computational cost of the algorithm will be discussed.

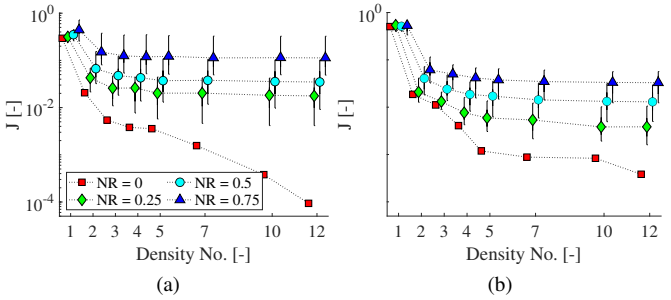


Fig. 5. Cost function value for SI (a) and DI (b) for different noise levels and candidate pool densities. Markers indicate the sample median and whiskers indicate the 2.5 and 97.5 percentiles.

A. Cost Function

Fig. 5 shows how the cost function develops for the two CE dynamics, for different candidate pool densities and noise ratios (NRs). The decrease in cost function value with candidate pool density levels off after a certain density, especially for noisy data. There, the cost function remains at approximately the same level after $d_{CP} = 5$, for both controlled element (CE) dynamics. For noiseless data, the cost function keeps decreasing, especially for single integrator (SI) dynamics. For double integrator (DI) dynamics, an additional drop is also visible at $d_{CP} = 12$.

Another interesting feature is observed when comparing the cost function values between the two types of dynamics. For SI dynamics the cost function values are more affected by the noise levels, as the difference of the noiseless and the noisy cost function values is larger.

B. Parameter Estimates

When looking at the parameter estimates, first the general trend of the fits is observed, after which focus is put on the individual parameters. Therefore, Fig. 6 shows the mean absolute error (MAE) as a percentage of the true parameter values, see (22). Here, n_Θ is the number of parameters estimated.

$$MAE = \frac{1}{n_\Theta} \sum_{i=1}^{n_\Theta} \frac{|\theta_{i,est} - \theta_{i,ref}|}{\theta_{i,ref}} \cdot 100\% \qquad (22)$$

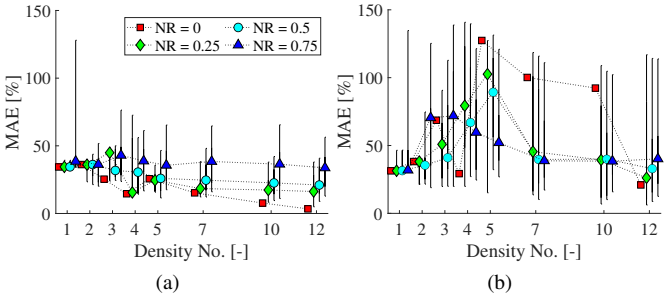


Fig. 6. Mean absolute error (MAE) of the parameter fits for SI (a) and DI (b) for different noise levels and candidate pool densities (horizontal axis). Markers indicate the sample median and whiskers indicate the 2.5 and 97.5 percentiles.

The figure shows a median and the 2.5 and 97.5 percentiles. A clear difference observed in the plots concerns the levels of the MAE for the two types of controlled elements. While for SI dynamics, the median MAE is always below 50 %, values of over 100 % are found for DI dynamics. This indicates that better estimates are obtained for SI dynamics.

Furthermore, the lowest MAEs for SI dynamics are generally found for noiseless data, where the values decrease steadily until $d_{CP} = 12$. For noisy data, the MAE is fairly flat after about $d_{CP} = 5$. This corroborates the observations of the cost function analysis.

For DI dynamics it occurs that the MAEs are larger for noiseless data at a number of densities. This is highly unexpected, as low noise should result in lower estimation errors. Only at $d_{CP} = 12$, the noiseless estimates become better than the medians of the noisy fits. It is especially interesting to see how the low cost function values in Fig. 5b correspond to large MAEs in Fig. 6b. This indicates that some parameters are difficult to estimate, despite low noise levels. This is investigated in more detail when looking at the specific parameters.

Finally, for larger noise levels — i.e., NR = 0.75 — the MAE is fairly constant (around 30%) after $d_{CP} = 5$ for DI dynamics.

Figs. 7 and 8 show the parameter fits of the parameters of $H_{o_{cmp}}$ and H_{o_p}, respectively. The plots show the mean and error bars indicating the 95% confidence interval for the different noise levels, for both CE dynamics. Furthermore, dashed lines indicate the true parameter value (black) and the parameter limits set by the candidate pool setup. For DI dynamics, it is seen that the limits for K_e and τ_f are adjusted after $d_{CP} = 5$. This is done to reduce the computational time of the algorithm for denser candidate pools and was deemed possible as optimal fits from lower density CPs did not occur in those ranges, even with added noise. Finally, note that the red parameter limits are missing in the plot of K_f, as this parameter is solved through quadratic optimization and is thus not limited.

First, the plots in Figs. 7 and 8 show how some parameters are estimated more accurately than others. For example, when comparing the fits of the target response gain K_f (Fig. 8b) and the damping ratio of the neuromuscular system (NMS)

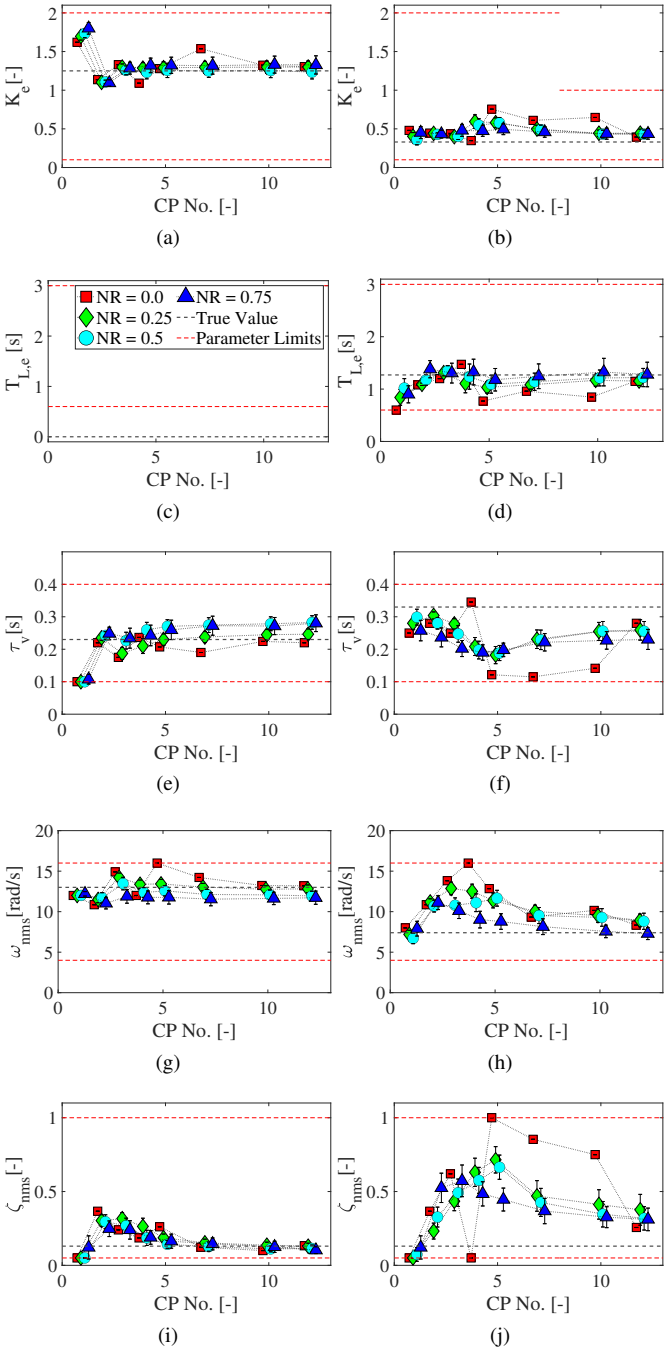


Fig. 7. Parameter estimates of the feedback subsystem $H_{o_{cmp}}$ for SI (a, c, e, g & i) and DI (b, d, f, h & j) CE dynamics. Errorbars indicate the 95% confindence interval.

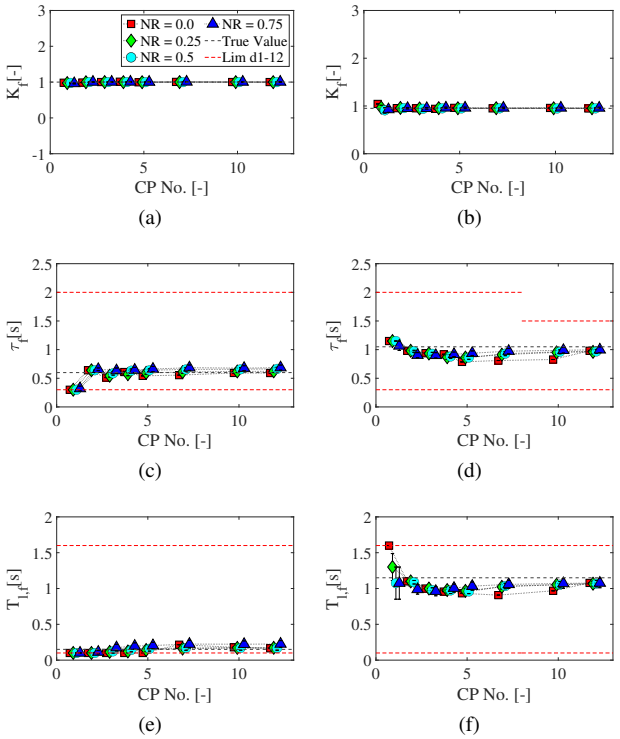


Fig. 8. Parameter estimates of the feedforward subsystem H_{o_p} for SI (a, c, e, g & i) and DI (b, d, f, h & j) CE dynamics. Errorbars indicate the 95% confindence interval.

ζ_{nms} (Fig. 7j) for DI dynamics, it is seen that the former has a higher quality fit than the latter. The same can be said when comparing τ_v (Fig. 7e) and $T_{l,f}$ (Fig. 8e) for SI dynamics. These results provide an indication of the sensitivity of the different parameters, which depends on the type of dynamics, but also on the exact values of the selected set of reference (or true) parameters.

Second, when comparing the two types of dynamics, the individual parameters confirm the observations in the MAE plots. For DI dynamics, both the estimation error and the spread in the estimates for different realizations are generally larger. The exception here is found at $K_{e^\star}$, which is more accurately estimated for DI dynamics.

Besides confirming the differences observed when comparing SI and DI estimates, the plots explain the odd results of the MAEs in Fig. 6b. It is interesting to see that indeed for almost every parameter in the DI simulations, the noiseless simulation estimates have a larger offset from the reference value than the estimates obtained from simulations with noise. Furthermore, it is seen that the estimation errors of the high-frequency NMS parameters (ω_{nms} and ζ_{nms}) and the feedback time-delay τ_v have the largest influence on the MAE. For example, the estimated value of $\zeta_{nms} \approx 0.75$ for NR = 0 and $d_{CP} = 10$ is about 500% larger than the original value.

As a final observation, it is interesting to see that, especially for DI dynamics, the pre-filter dynamics H_{o_p} are estimated more accurately than those of the compensatory dynamics $H_{o_{cmp}}$. This observation is further elaborated on in the next paragraph.

C. Frequency Domain Fits

The frequency response functions (FRFs) of the closed- and open-loop dynamics can be formed from the parameter fits. Fig. 9 shows these dynamics for noiseless data and a number of different candidate pool densities. The figures show how the parameter fits for coarser candidate pools result in nicely fitting closed-loop FRFs (for H_{cl}), while a significantly denser candidate pool is needed to approximate the subsystem dynamics (H_{o_p} and $H_{o_{cmp}}$). The difference between the fits in

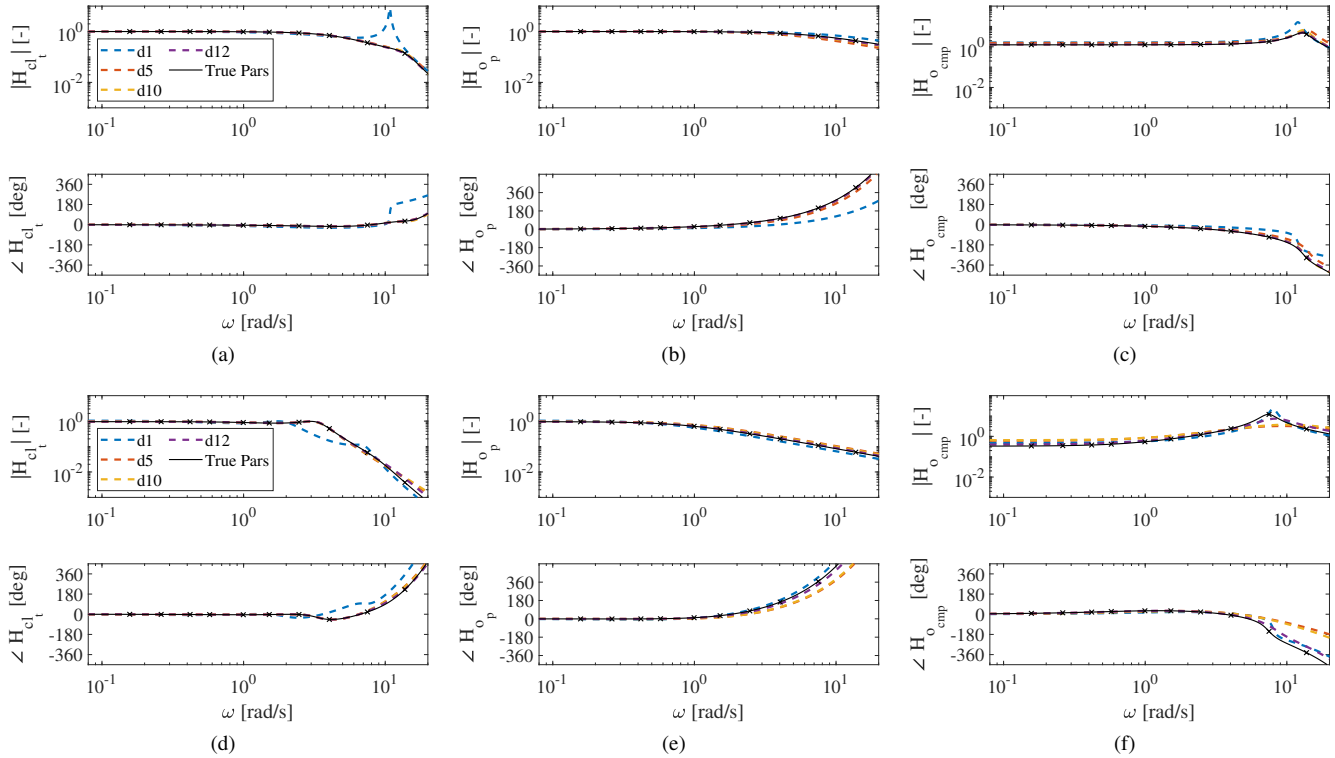


Fig. 9. Frequency domain functions of the closed-loop (H_{cl}) and open-loop (H_{o_p} and $H_{o_{cmp}}$) of the estimated- and reference parameters for noiseless data for SI- (a, b, c) and DI (d, e, f) dynamics. Different lines indicate different CP densities.

the closed-loop and open-loop responses is especially apparent for DI dynamics and for $H_{o_{cmp}}$, see Fig. 9f. There, the fit only starts to approximate the reference at high frequency at d_{12}. It is noted that the neuromuscular system (NMS) mainly affects high frequencies. Considering that the parameters of the NMS in Fig. 7 for noiseless data showed a large offset, this is in accordance with those observations. An explanation for these results could be found in the magnitude of the closed-loop function at high frequencies, which is about 10 to 100 times smaller at frequencies above 4 rad/s for DI dynamics (see Fig. 9d).

To illustrate the observed difference in the frequency domain estimates in more detail, Fig. 10 shows the development of the original and an alternative cost function for both dynamics. The alternative cost function evaluates the models based on the subsystem (or open-loop system) dynamics, see (23). The figure shows how, for both SI and DI dynamics, the original cost function decreases by about 1,000 over the first 5 densities, indicating an accurate estimate of the closed-loop system. However, the alternative cost function does, especially for DI dynamics, not decrease much (less than a factor of 10). This confirms that the subsystems are not as well estimated as the closed-loop frequency response itself.

$$J_{alt} = \sum_{k=1}^{N_f} \left| H_{o_p,meas}(j\omega_k) - H_{o_p,mod}(j\omega_k) \right|^2 + \left| H_{o_{cmp},meas}(j\omega_k) - H_{o_{cmp},mod}(j\omega_k) \right|^2 \quad (23)$$

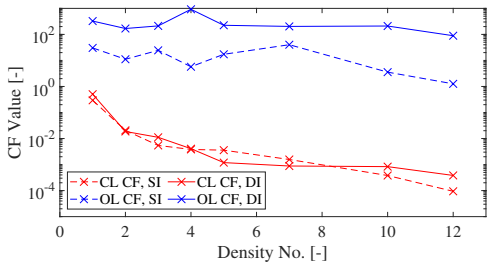


Fig. 10. Comparison of the original- (closed-loop) and the alternative (open-loop) cost functions for both types of dynamics.

D. Computational Time

Finally, Fig. 11 shows how the computational time of the algorithm develops with increasing candidate pool density. In the figure, trend lines are added to indicate exponential growth. The computational times for SI dynamics are shorter than those for the DI dynamics, as the candidate pool is smaller for this case due to an omitted parameter ($T_{L,e}$). Fig. 11 shows that the difference is between a factor of 5 and 20. The calculations were made on four Intel Xeon E5-1620 @3.5 GHz processors.

VI. Discussion

The simulations in this paper produced three main insights. First, it was demonstrated how the SSID algorithm can be implemented to identify the parameters of the human operator (HO) model. This is done in such a way that the candidate pools consist of the model parameters, which can all be converted to and from the original model coefficients (α, β,

TABLE V
OVERVIEW OF TWO DIFFERENT SSID ALGORITHMS AND THE MOST RELEVANT DATA.

	# Pars Φ	# Pars α	N_f	Size Φ	Domain	Program	t_{run} [h]	# Cores	# calcs/core/s
DI $d_{CP} = 12$	6	1	10	1.25E+09	$s/j\omega$	MATLAB	12	4 Intel E5-1620 @3.5 GHz	7.23E+03
SSID [11]	3 (G_{ff} = FIR)	3	30	1.00E+09	z	C++	3	16 Intel E5-2670 @ 2.6 GHz	5.94E+03

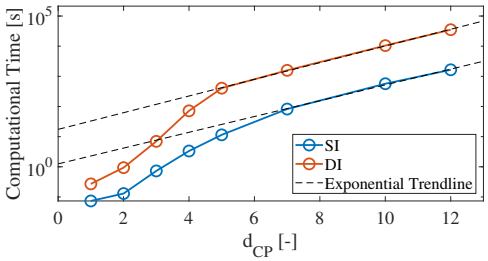


Fig. 11. Computational times of a single run of the SSID algorithm for different candidate pool densities and both types of CE dynamics. Exponential trends are added for clarity.

γ and ψ). Compared to the target-error (ET, see Section II) response, the algorithm is actually slightly simplified (see the explanation of (10)). It is thus possible to perform SSID to both an ET and a TX structure. It is, however, doubtful whether the method can also be applied to an EX structure, as this structure has all responses in the closed loop. This results in the absence of α, the parameter which is solved with the convex optimization. Then, this part disappears and the algorithm basically becomes a grid search over all parameters. However, this should be investigated in more detail.

Second, in the results, similar to [9], [11], the algorithm showed how the parameter estimates improve when the candidate pool density increases and noise levels decrease. However, despite the trend of improving parameter estimates with increased candidate pool density, the SSID algorithm did show problems estimating specific parameters. Although for single integrator (SI) dynamics, the parameter estimates are quite accurate with a predicted MAE of less than 20 % for realistic noise ratios (i.e., between $\sigma^2_{u_n}/\sigma^2_u$ of 0.25 and 0.5), this was different for double integrator (DI) dynamics. There, estimation results were worse regarding both the estimation error and the spread in the estimates over different realizations. It is remarkable to see that, even for noiseless data, large estimation errors (MAEs of over 100 %) were found up to a density of $d_{CP} = 10$. Particularly, the feedback time delay τ_v and the neuromuscular system (NMS) parameters (ω_{nms} and ζ_{nms}) are badly estimated. These parameters have a strong influence on the human control behaviour at high frequencies.

An explanation for the lower identifiability of these parameters is found in the bode plots the identified systems of the closed- and open-loop dynamics in Fig. 9, which clearly show how these parameters (τ_v, ω_{nms} and ζ_{nms}) have little impact on the closed-loop response, while being more clearly present in the open loop-response. This observation might indicate that the data used for the SSID technique, namely the closed-loop FRF, are not suitable for estimating the model parameters of both subsystems. In other words, the system may not be identifiable [27] when only the closed-loop system data are used. Other signals, such as for example the control output $U(j\omega)$ or the subsystem dynamics themselves, might allow for improved identification results for similar conditions. However, this might require addition of a second forcing function, which would enable to distinguish between the two responses. It is recommended to investigate this in more detail in future research.

It could be argued that, in order to improve the fits, modifications to the set of candidate pool models are required. This could be done by changing the initial step sizes $n_{\theta,1}$ (see Table IV) or by further increasing the candidate pool densities. The first option would involve forming the candidate pools based on the sensitivities of the cost function for the DI reference parameters. However, it should be noted that by doing this, the candidate pools are being biased towards the reference parameters, as the sensitivities might be influenced by the selected reference points. Care should therefore be taken, as this is obviously not possible when applying this technique to real-life situations, as the true parameters are then unknown.

The second option of increasing the candidate pool density also encounters a problem: the computational time required for the algorithm. This leads to the third insight of the results. As the candidate pool size (and thus the computational time) increases exponentially with candidate pool density (see Fig. 11), there is a limit after which the method becomes too computationally complex. The current SSID algorithm for this research is programmed in MATLAB and requires about 12 hours for $d_{CP} = 12$ (2.6 billion elements) on the setup as described in Section V-D (four Intel Xeon E5-1620 @3.5 GHz workers). Considering the 31 times the algorithm is run per density for the simulations in this paper, candidate pool densities with much larger numbers for d_{CP} will start causing problems regarding computational power.

Table V provides an overview of the SSID algorithm in this paper for $d_{CP} = 12$ for DI dynamics, and the reference algorithm applied in [11]. The table summarizes the differences between the different algorithms and types of models that were estimated. The last column shows the number of entries of the closed-loop stability candidate pool that are evaluated per second. The numbers indicate that both algorithms evaluate approximately the same number of entries per core per second. This does, however, not mean both algorithms are equally fast, as a couple of aspects are not being reflected in this number. For example, the algorithm in [11] estimates 3 parameters for α. This makes quite an impact on the computational power of the elements, as this number relates to the size of a matrix that is to be inverted for every entry in the closed-loop stability candidate pool, which is a relatively computationally

expensive step, see [24]. Furthermore, the number of frequency points N_f is larger (30 vs. 10), meaning more calculation per candidate pool entry. These aspects are countered by the fact that the algorithm is implemented in the z-domain and is written in C++.

Considering the comparison between the two implementations of the SSID algorithm, there are various remaining options to increase computational efficiency of the SSID algorithm. However, it should be noted that despite these improvements, difficulties regarding computational performance will remain. It is therefore interesting to see how the performance of this technique relates to other techniques used for similar parameter estimation problems (as treated in e.g. [5], [8], [22]). In fact, this interest extends past the area of computational power, as it is also interesting to compare the accuracy of fit for different noise levels and the previously mentioned identifiability. Such a general comparison between the different methods can provide valuable information regarding the suitability of different techniques for different applications.

VII. Conclusion

This article investigated the application of a subsystem identification (SSID) technique to a quasi linear HO model for pursuit- and preview tracking tasks. With some modifications regarding earlier versions of this technique, it is shown how the technique can be successfully applied to estimate the human operator (HO) model parameters for both pursuit- and preview tasks for two different controlled element dynamics. Furthermore, simulations showed how, similar to earlier work, the technique performs for different noise levels and candidate pool densities. A valuable addition in this paper concerns the fact that noise levels used are comparable to those encountered in real human-in-the-loop experiments. For realistic noise levels, mean absolute errors of 15 % and 35 % were predicted for single- and double integrator dynamics, respectively. However, the method also showed that, under specific conditions, the high frequency components (i.e., the neuromuscular dynamics) and the feedback time delay are more difficult to estimate using the closed-loop response function.

References

[1] M. Mulder, D. M. Pool, D. A. Abbink, E. R. Boer, P. M. T. Zaal, F. M. Drop, K. van der El, and M. M. van Paassen, "Manual Control Cybernetics: State-of-the-Art and Current Trends," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 468–485, Oct. 2018.

[2] D. T. McRuer and H. R. Jex, "A Review of Quasi-Linear Pilot Models," *IEEE Transactions on Human Factors in Electronics*, vol. HFE-8, no. 3, pp. 231–249, Sep. 1967. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1698271

[3] T. P. Neal and R. E. Smith, "A Flying Qualities Criterion for the Design of Fighter Flight-Control Systems," *Journal of Aircraft*, vol. 8, no. 10, pp. 803–809, Oct. 1971.

[4] R. A. Hess and K. K. Chan, "Preview Control Pilot Model for Near-Earth Maneuvering Helicopter Flight," *Journal of Guidance, Control, and Dynamics*, vol. 11, no. 2, pp. 146–152, 1988.

[5] K. van der El, D. M. Pool, H. J. Damveld, M. M. van Paassen, and M. Mulder, "An Empirical Human Controller Model for Preview Tracking Tasks," *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2609–2621, Nov. 2016.

[6] M. M. van Paassen and M. Mulder, "Identification of Human Operator Control Behaviour in Multiple-Loop Tracking Tasks," in *Proceedings of the Seventh IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, Kyoto Japan*. Pergamon, 1998, pp. 515–520.

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

[8] P. M. T. Zaal, D. M. Pool, Q. P. Chu, M. M. van Paassen, M. Mulder, and J. A. Mulder, "Modeling Human Multimodal Perception and Control Using Genetic Maximum Likelihood Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 4, pp. 1089–1099, 2009.

[9] X. Zhang and J. B. Hoagg, "Frequency-Domain Subsystem Identification with Application to Modeling Human Control Behavior," *Systems & Control Letters*, vol. 87, Jan. 2016.

[10] X. Zhang, S. Wang, J. B. Hoagg, and T. M. Seigler, "The Roles of Feedback and Feedforward as Humans Learn to Control Unknown Dynamic Systems," *IEEE Transactions on Cybernetics*, vol. 48, no. 2, pp. 543–555, Feb. 2018.

[11] S. A. S. Mousavi, X. Zhang, T. Seigler, and J. B. Hoagg, "Subsystem identification of feedback and feedforward systems with time delay," *Results in Control and Optimization*, vol. 1, p. 100002, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2666720720300023

[12] W. H. Levison, S. Baron, and D. L. Kleinman, "A Model for Human Controller Remnant," *IEEE Transactions on Man-Machine Systems*, vol. 10, no. 4, pp. 101–108, Dec. 1969.

[13] K. van der El, D. M. Pool, and M. Mulder, "Analysis of Human Remnant in Pursuit and Preview Tracking Tasks," in *Proceedings of the 14th IFAC Symposium on Analysis Design and Evaluation of Human Machine Systems Tallinn, Estonia*, 2019, pp. 145–150.

[14] A. J. S. Sheffler, S. A. S. Mousavi, E. Hellström, M. Jankovic, M. A. Santillo, T. M. Seigler, and J. B. Hoagg, "Effects of Reference-Command Preview as Humans Learn to Control Dynamic Systems," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Oct. 2019, pp. 4199–4204.

[15] K. van der El, S. Padmos, D. M. Pool, M. M. van Paassen, and M. Mulder, "Effects of Preview Time in Manual Tracking Tasks," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 486–495, Oct. 2018.

[16] T. B. Sheridan, "Three Models of Preview Control," *IEEE Transactions on Human Factors in Electronics*, vol. 7, no. 2, pp. 91–102, Jun. 1966.

[17] K. Ito and M. Ito, "Tracking Behavior of Human Operators in Preview Control Systems," *Electrical Engineering in Japan (English translation of Denki Gakkai Ronbunshi)*, vol. 95, no. 1, pp. 120–127, 1975.

[18] D. T. McRuer, R. W. Allen, D. H. Weir, and R. H. Klein, "New Results in Driver Steering Control Models," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 19, no. 4, pp. 381–397, Aug. 1977.

[19] B. Yu, B. R. Gillespie, J. S. Freudenberg, and J. A. Cook, "Human Control Strategies in Pursuit Tracking with a Disturbance Input," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 3795–3800.

[20] R. L. Stapleford, D. T. McRuer, and R. E. Magdaleno, "Pilot Describing Function Measurements in a Multiloop Task," *IEEE Transactions on Human Factors in Electronics*, vol. 8, no. 2, pp. 113–125, Jun. 1967.

[21] R. J. Wasicko, D. T. McRuer, and R. E. Magdaleno, "Human Pilot Dynamic Response in Single-loop Systems with Compensatory and Pursuit Displays," Air Force Flight Dynamics Laboratory, Tech. Rep. AFFDL-TR-66-137, Dec. 1966.

[22] K. van der El, D. M. Pool, M. M. van Paassen, and M. Mulder, "Effects of Target Trajectory Bandwidth on Manual Control Behavior in Pursuit and Preview Tracking," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 1, 2020.

[23] D. T. McRuer, D. Graham, E. S. Krendel, and W. J. Reisener, "Human Pilot Dynamics in Compensatory Systems: Theory, Models, and Experiments with Controlled Element and Forcing Function Variations," Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base (OH), Tech. Rep. AFFDL-TR-65-15, 1965.

[24] X. Zhang and J. B. Hoagg, "Subsystem Identification of Multivariable Feedback and Feedforward Systems"," *Automatica*, vol. 72, pp. 131 – 137, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109816302229

[25] H. J. Damveld, G. C. Beerens, M. M. van Paassen, and M. Mulder, "Design of Forcing Functions for the Identification of Human Control Behavior," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1064–1081, Jul. 2010.

[26] K. van der El, D. M. Pool, M. M. van Paassen, and M. Mulder, "Effects of Preview on Human Control Behavior in Tracking Tasks With Various Controlled Elements," *IEEE Transactions on Cybernetics*, vol. 48, no. 4, pp. 1242–1252, Apr. 2018.

[27] L. Ljung, *System Identification Theory for the User*, 2nd ed. Prentice Hall, Inc., 1999.

II

Paper II: Comparison of Identification Techniques used to Identify Feedback- and Feedforward Control Behaviour in Preview Tracking Tasks

Comparison of Identification Techniques used to Identify Feedback- and Feedforward Control Behaviour in Preview Tracking Tasks

Author: Pieter-Bas J. C. Bentinck,
Supervisors: Kasper van der El, Daan M. Pool, Max Mulder
Control & Simulation Section, Faculty of Aerospace Engineering,
Delft University of Technology, Delft, The Netherlands

In recent efforts to improve the theory of feedforward human control behaviour, different system identification techniques have been developed to estimate human feedback- and feedforward control behavior. This paper compares the performance of six system identification techniques used to identify linear time-invariant control behaviour in preview tracking tasks. The methods differ in the number of forcing functions required, the form in which the experiment reference data is used in the cost function and the optimization algorithm. The six methods are compared through offline simulations and a human-in-the-loop experiment for a preview tracking task performed with two controlled element dynamics. Results show that the best — a mean absolute error of the parameter estimates of less than 10% and a VAF of over 80% — parameter fits are obtained when using two forcing functions and when using the Fourier Transform of the control output signal for the cost function data. Furthermore, use of a non-linear Nelder-Mead optimization algorithm is computationally favourable compared to a candidate pool-based subsystem identification algorithm. This implies that, despite the additional efforts required in the experiment setup, two forcing functions are required to identify feedback- and feedforward control behavior, especially when looking at higher (5 to 15 rad/s) frequencies.

I. Introduction

In the quest of forming a unifying theory of human control behaviour, Mulder et al. (2018) argue that an improved understanding of the human's ability to apply feedforward control is required [1]. This feedforward behaviour, seen in for example car driving, is often applied in daily life. However, contrary to the highly-constrained and well-understood compensatory control task [2], a widely accepted theory on this behaviour is missing. Advancements in this area can lead to an acceleration in the design of human-machine systems, similar to the acceleration seen following the crossover model for compensatory tracking [1], [3], [4].

Recently, several efforts have been made to better understand feedforward control behaviour, specifically the behaviour in pursuit- and preview tracking tasks [5]–[8]. For example, the human operator (HO) model developed by van der El et al. (2016) offers a good understanding of how humans use preview in manual control, while building on the existing theory of the cross-over model [7].

However, besides developments regarding the models and theory itself, the system identification techniques used to develop this theory have also advanced. When focusing on the identification of linear time-invariant (LTI) models of feedback- and feedforward control behavior, several methods have been used in literature [6], [7], [9], [10]. One of the recent methods is a subsystem identification (SSID) technique that allows for estimation of multiple subsystems from a measured closed-loop frequency response estimate. The fact that the method only requires data from a single forcing function, sets it apart from other methods [6] and can remove current obstacles in the experiment setup. In [11], it was shown how the technique can be used to identify the model parameters of the preview model from [7]. As a next step, it is essential to evaluate how this method performs compared to other methods frequently used to identify preview control behaviour.

Therefore, the goal of this paper is to discuss the different identification techniques and provide a direct quantitative comparison of their performance. To achieve this, the methods are evaluated using offline simulations and a human-in-the-loop experiment. The performance is evaluated for a preview tracking task with two different controlled element (CE) dynamics and different noise levels. For the methods, the spread in the describing function data, the identification results — in terms of the mean absolute error of the parameter fits, the variance-accounted-for and the error of the frequency-domain subsystems — and the computational time are evaluated.

The structure of the paper is as follows. First, Section III gives an overview of the control task and the different methods evaluated. Then, Section IV elaborates on the setup of the simulations and the experiment in the methods section. This is followed by a presentation- and a discussion of the results in Sections V and VI, respectively. Finally, conclusions are listed in Section VII.

II. Control Task

There are several types of tracking tasks that allow for both feedback- and feedforward control behaviour, such as pursuit- and preview tracking tasks [7], [12] or tracking tasks with predictable target signals [13], [14]. For this research, a preview tracking task is selected, as this type of tracking task allows for a clear feedforward response and has often been used in combination with frequency-domain identification techniques (which form the focus point of this research) [6], [9]–[11], [15]. Furthermore, well-fitting and intuitive HO models are found for such tracking tasks [7].

Fig. 1 provides an example of a typical preview display. In the tracking tasks, the human operator (HO) is to minimize the distance between the controlled element (CE) output $x(t)$ and the target signal $f_t(t)$, which is defined as the error $e(t)$. Feedforward can be enabled as the future trajectory of the

target signal is shown, up to a certain preview time τ_p. When $\tau_p = 0s$, the tracking task reduces to a pursuit tracking task and when only the error is $e(t)$ is displayed, the task becomes a compensatory tracking task [16].

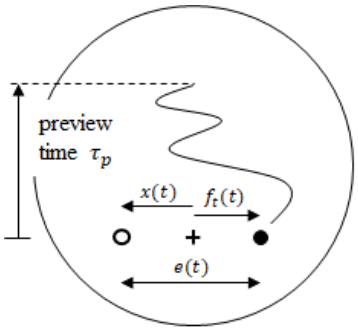
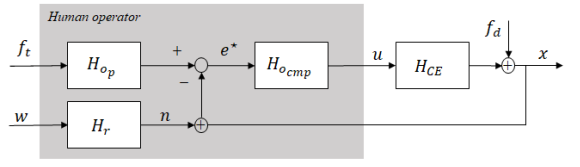


Fig. 1. Schematic representation of a preview tracking task.

The modelling of the human operator dynamic response in a preview tracking task starts with the input signals. When looking at Fig. 1, there are three signals to which the human can respond: the target signal f_t, the output x and the error e. Problems regarding the identification of all three signal responses involve an overdetermined response structure due to the linear relation between the input signals and the inability to add three meaningful perturbations to measure the three responses [17]–[19]. Therefore, two-channel structures are generally selected for modelling human control behaviour in preview tracking tasks [7], [12]. Van der El et al. [7] showed how an intuitive model can be developed based on a structure using the target and controlled element output responses (TX) and [11] showed how the subsystem identification (SSID) technique developed in [6], [11], [12] can be applied to this model structure. This TX model structure, that combines pure feedforward with a controlled element output feedback response, is therefore also used in this paper.

Fig. 2 shows this preview model as used in [10]. The model is an equivalent version of the original TX model, although it is written in the form of a compensatory structure combined with a target pre-filter. The quasi-linear model is based on McRuer's crossover model [16], but instead of the actual error, the human controls an imaginary error τ_f seconds ahead.



$$H_{o_{cmp}}(j\omega) = H_{o_{e^*}} H_{nms} e^{-j\omega\tau_v}, \quad H_{o_{e^*}}(j\omega) = K_e \frac{1+T_{L,e}j\omega}{1+T_{l,e}j\omega}, \quad H_{o_p}(j\omega) = H_{o_f} e^{j\omega\tau_f}$$

$$H_{o_f}(j\omega) = \frac{K_f}{T_{l,f}j\omega+1}, \quad H_{nms}(j\omega) = \frac{\omega_{nms}^2}{(j\omega)^2+2\zeta_{nms}\omega_{nms}(j\omega)+\omega_{nms}^2}, \quad H_r(j\omega) = \frac{K_r}{T_{l,r}j\omega+1}$$

K_f	= target response gain	τ_f	= look-ahead time
$T_{l,f}$	= lag/smoothing time-constant	K_e	= error response gain
$T_{L,e}$	= lead time-constant	$T_{l,e}$	= lag time-constant
ω_{nms}	= neuromuscular break frequency	τ_v	= time delay
ζ_{nms}	= neuromuscular damping ratio		

Fig. 2. Overview of the pursuit- and preview model as proposed by [10].

The model consists of a linear part and a non-linear part. The linear part consists of the pre-filter dynamics H_{o_p} and the compensatory dynamics $H_{o_{cmp}}$. The pre-filter dynamics are formed by a low-pass filter, containing a gain K_f and a lag-time constant $T_{l,f}$, and a feedforward look-ahead time τ_f, expressed as a negative time delay. The compensatory dynamics are formed by the equalization dynamics $H_{o_{e^\star}}$, the neuromuscular system (NMS) dynamics H_{nms} and a feedback time delay term, parametetrized by τ_v. The equalization dynamics consist of a gain $K_{e^\star}$ and a lead-lag filter characterized by the lead- and lag-time constants $T_{L,e^\star}$ and $T_{l,e^\star}$, respectively. These time constants vary depending on the CE dynamics [16], as $T_{L,e^\star}$ is non-zero for double integrator dynamics and $T_{l,e^\star}$ is non-zero for gain dynamics. The neuromuscular dynamics are represented by a second-order low-pass filter, described by ω_{nms} and ζ_{nms}.

The non-linear part of the human response is approximated by injecting filtered white noise at the system output x [20]. This part of the model is only used for simulation purposes, as the remnant contribution is assumed to be zero during the actual identification, as will be discussed next. The filter H_r is a low-pass filter [20], [21], where the time constant $T_{l,r}$ is selected based on the CE dynamics. The gain K_r is tuned to reach a specific noise level at the control output u, which depends on the CE dynamics and the signal bandwidth [20].

III. Identification Methods

A number of frequency-domain system identification techniques, used to identify linear-time-invariant human control behavior, will be compared. Table I lists of the methods that are evaluated in this paper. It shows how the methods can be distinguished based on three aspects: the number of forcing functions used, the cost function (CF) data and the optimization algorithm. These three methods will be discussed in the coming subsections.

Finally, the table shows in which previous research the different methods have been applied. Note that some of the methods have not been used in previous work and are developed specifically for this paper. This is done to illustrate the effect of the three evaluated variables: the number of forcing functions, the cost-function data and the optimization algorithm. For example, while using the same cost function data, the SSID and CLDF_t methods use a different optimization algorithm. Similar results between these two methods can then indicate that this optimization algorithm has little effect on the estimation. Similarly, comparing the CLDF_t and CLDF_td (methods 2 and 3) will address the effect of adding a second forcing function, while comparing the CLDF_td and OLDF methods will provide information regarding the effects of open- or closed-loop data used for identification. Finally, methods 5 and 6 (U1_t and U2_td) provide another comparison of the effects of the number of forcing functions and the frequencies at which the model is estimated.

A. Number of Forcing Functions

The first distinguishing feature to be discussed is the number of forcing functions used to generate the data. This is either

TABLE I
OVERVIEW OF THE METHODS EVALUATED IN THIS PAPER.

Method #	Abbreviation	Forcing functions	CF data	Optimization Algorithm	Reference
1	SSID	f_t	CL DF	SSID	[6], [11], [12]
2	CLDF_t	f_t	CL DF	NM simplex	-
3	CLDF_td	f_t, f_d	CL DF	NM simplex	-
4	OLDF	f_t, f_d	OL DF	NM simplex	[7]
5	U1_t	f_t	$U(j\omega)$	NM simplex	-
6	U2_td	f_t, f_d	$U(j\omega)$	NM simplex	[9], [10]

one (only f_t) or two (both f_t and f_d). Fig. 2 shows how both input functions are used to excite the system. Previous experiments [7], [9], [10] have deliberately used two forcing functions, to separately excite the two HO responses. This allows, for example, for measurement of the two open-loop responses by estimating the describing functions (DFs). On the other hand, when applying solely the target function in an experiment (see e.g., [12], [22], [23]), the data only provides information on the closed-loop system response. As mentioned, comparison of the CLDF_t and CLDF_td and the U1_t and U2_td methods will provide an indication of the performance difference caused by the added forcing function.

B. Cost Function Data

The cost function (CF) data is a second aspect by which the methods differ, with methods using the closed-loop- (CL) or open-loop (OL) describing function (DF), or the control output signal $U(j\omega)$. Although the cost function data of the different methods may vary, the setup of the cost function is the same. All methods treated in this research use a least squares based approach, where a cost function value is minimized, according to (1).

$$\Theta = \underset{\Theta}{\operatorname{argmin}} \sum_{k=1}^{N_f} W_k \left|\epsilon\left(j\omega_k|\Theta\right)\right|^2 \tag{1}$$

The vector Θ contains all the model parameters to be estimated, while ω_k indicates the evaluated frequencies that range from 1 to N_f. ϵ then represents the error between the measured and the simulated data and W_k is a weighting factor that can be applied based on, e.g., the accuracy of the frequency response data [24].

As mentioned, while the least-squares approach is used by all methods, the cost function data is different. Fig. 3 shows the two-channel model structure (equivalent to the pre-filter model structure in Fig. 2) and the different data used for the cost function. In the figure, Fig. 3a shows the block diagram of the linear part of the HO model in the tracking task. This model represents the same model as described in Fig. 2, but posed in the TX configuration. The red-, blue- and purple boxes indicate the data used for the parameter estimations of the different methods. Fig. 3b shows the subsystem dynamics H_{o_t} and H_{o_x}, which can be estimated using a non-parametric identification method [18] when two forcing functions are used. Fig. 3c shows the closed-loop dynamics with respect to f_t ($H_{cl,t} = \frac{X(j\omega)}{F_t(j\omega)}$ and f_d ($H_{cl,d} = \frac{X(j\omega)}{F_d(j\omega)}$). Finally, Fig. 3d and Fig. 3e show spectrum of the control output $U(j\omega)$ for one and two forcing functions, respectively.

For the SSID technique as used in [6], [11], this is the closed-loop describing function (DF) data $H^{DF}_{cl,t}(= \frac{X(j\omega)}{F_t(j\omega)})$, see (2) and the green crosses in Fig. 3 (c). In (2), the frequencies ω_k correspond to the frequencies at which the target function has power, as the SSID method only uses data from a single forcing function. This is also why for this method, the blue triangles in Fig. 3 (c) are not considered. This cost function data is used for both SSID and CLDF_t.

$$\epsilon_{cl,t}(j\omega_k) = H^{mod}_{cl,t}(j\omega_k|\Theta) - H^{DF}_{cl,t}(j\omega_k) \tag{2}$$

When in addition to just the f_t, also a disturbance function is added, (1) is extended with a second term to become (3), where $\epsilon_{cl,d}$ is given by (4). Analogous as for $H_{cl,t}$, $H_{cl,d}$ is formed by $\frac{X(j\omega)}{F_d(j\omega)}$). In this case, both the green crosses and the blue triangles in Fig. 3 (c) are evaluated. This data is used for the CLDF_td method.

$$\begin{aligned}\Theta = \underset{\Theta}{\operatorname{argmin}} \sum_{k=1}^{N_f} & W_k \left|\epsilon_{cl,t}\left(j\omega_k|\Theta\right)\right|^2 \\ & + W_k \left|\epsilon_{cl,d}\left(j\omega_k|\Theta\right)\right|^2\end{aligned} \tag{3}$$

$$\epsilon_{cl,d}(j\omega_k) = H^{mod}_{cl,d}(j\omega_k|\Theta) - H^{DF}_{cl,d}(j\omega_k) \tag{4}$$

Besides looking at the closed-loop cost function, another set of data used to estimate the HO model parameters involves the open-loop subsystem dynamics $H_{o_t}(j\omega)$ and $H_{o_x}(j\omega)$ (e.g. in [7]). However, as the estimation is then performed on two DFs, (1) becomes slightly different, see (5). Eqs. (6) and (7) then represent $\epsilon_{OL,1}$ and $\epsilon_{OL,2}$, respectively. In these equations, the responses of $H^{DF}_{o_t}(j\omega_k)$ and $H^{DF}_{o_x}(j\omega_k)$ are obtained using a non-parametric system identification method as described in [17], [18]. Through interpolation, the method estimates the DFs at both the target- and disturbance frequencies. This explains why (5) uses $2N_f$ for the summation, if N_f represents the number of multi-sine frequencies of a single forcing function (e.g., f_t). This difference can also be observed in Fig. 3 (b), where the open-loop dynamics are shown graphically.

$$\begin{aligned}\Theta = \underset{\Theta}{\operatorname{argmin}} \sum_{k=1}^{2N_f} & W_k \left|\epsilon_{OL,1}\left(j\omega_k|\Theta\right)\right|^2 \\ & + W_k \left|\epsilon_{OL,2}\left(j\omega_k|\Theta\right)\right|^2\end{aligned} \tag{5}$$

$$\epsilon_{OL,1}(j\omega_k) = H^{mod}_{o_t}(j\omega_k|\Theta) - H^{DF}_{o_t}(j\omega_k) \tag{6}$$

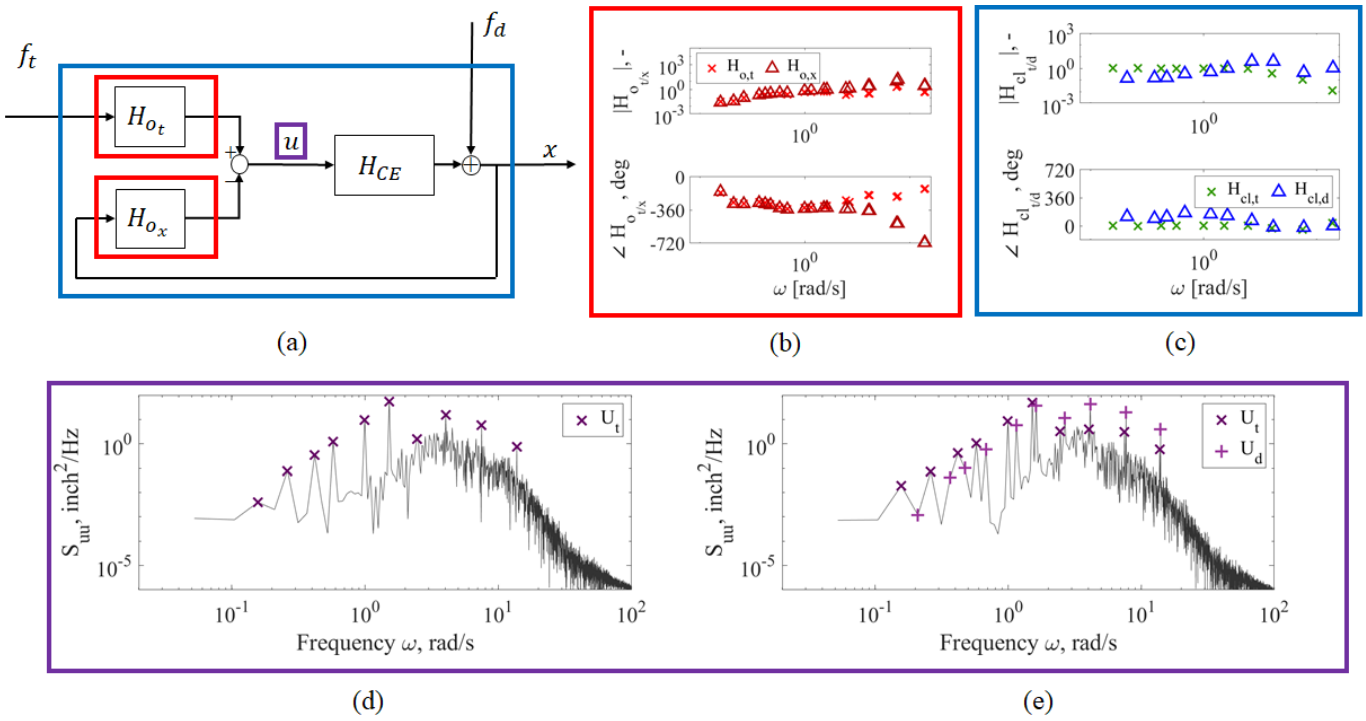


Fig. 3. Two-channel model structure of the preview model (a) and examples of the different data sets used to estimate the model parameters. These are the open-loop dynamics H_{o_t} and H_{o_x} (b), the closed-loop dynamics H_{cl_t} and H_{cl_d} (c) and the control output signal $U(j\omega)$ with one (d) and two (e) forcing functions.

$$\epsilon_{OL,2}(j\omega_k) = H_{o_x}^{mod}(j\omega_k|\Theta) - H_{o_x}^{DF}(j\omega_k) \tag{7}$$

Next to the describing functions, an alternative cost function is formed by minimizing the error of the control output $U(j\omega)$, as done in [9], [10], [15]. Then, (1) can be used with the error ϵ given by (8). Here, $U^{mod}(j\omega|\Theta) = H_{o_t}(j\omega|\Theta)F_t - H_{o_x}(j\omega|\Theta)X$. In this format, N_f can be defined as the total amount of frequencies to be evaluated, either belonging to the target function, target- and disturbance function, or even all frequencies. This because, contrary to the DF cost functions, fitting on the frequency-domain signals can also be done on the remnant frequencies (i.e., all frequencies at which f_t and f_d do not have power). Fig. 3 (d) and (e) show representative examples of the power spectral densities (PSDs) of $U(j\omega)$ when using one and two forcing functions, respectively.

$$\epsilon_U(j\omega_k) = U^{mod}(j\omega_k|\Theta) - U^{DF}(j\omega_k) \tag{8}$$

Finally, although all methods use discrete Fourier Transforms of the input- and output signals, the pre-processing of this data can be different. For example, differences in averaging can be observed in the application of different identification techniques (see [10] and [12]). In this research, this pre-processing step is applied in the same way for all methods and is thus a control variable. After all, the effects of averaging frequency-domain data (to reduce noise levels) have been discussed in earlier work [16], [24]. Furthermore, this allows to put the focus of the comparison on the specific data type of the cost function and, to lesser extent, the optimization algorithm.

C. Optimization Algorithms

Besides the number of forcing functions and the various cost functions, differences in the optimization method are also present. The SSID methods used in [6], [11], [12] for example, use a grid search combined with a quadratic optimization to estimate the parameters. With this technique, candidate pool densities need to be selected. Care should be taken in this process, as the density is related to computational time.

The other methods used to identify human preview behaviour [7], [9], [10], [15] use a non-linear solver to minimize the cost function. Often, a Nelder-Mead (NM) simplex algorithm is used, repeated a certain amount of times (e.g., 100) from random initializations.

IV. Methods

To evaluate and compare the performance of the different system identification techniques, offline simulations and a human-in-the-loop experiment are performed. Overall, the focus of this paper lies on the simulations, while the experiment results are mainly used for validation of the simulations. Both the simulations and the human-in-the-loop experiment will evaluate a preview tracking task with a preview time of $\tau_p = 2s$ and two types of controlled element (CE) dynamics: a single integrator (SI) and a double integrator (DI). This section will elaborate on the methods used for both types of tests. First however, a set of hypotheses is posed.

A. Hypotheses

Regarding the tests described in the coming sections, a number of hypotheses are stated:

H1 When using closed-loop DF data to estimate the HO model parameters, the HO dynamics are not identifiable, even for noiseless data. As multiple combinations of parameters can fit the closed-loop data, the parameter set that matches the open-loop dynamics is not found.

H2 Parameter estimates improve when using a second forcing function, due to additional excitation of the HO dynamics.

H3 Fitting on the subsystem DFs (or OLDFs, H_{o_t} and H_{o_x}) leads to lower quality fits compared to fitting on $U(j\omega)$ with two forcing functions. This is both due to the larger noise in the DFs that arises due to interpolation and due to missing information of the relative strength of the forcing functions in the open-loop cost function data.

B. Offline Simulations

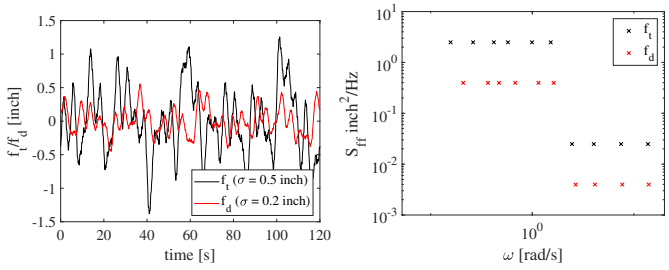
1) Forcing Functions

As mentioned in Section III, the forcing functions play an important role in the comparison of the different methods. After all, one of the unique capabilities of SSID is that it estimates multiple subsystems from data with a single forcing function. For this research, the forcing functions applied in the simulations and the experiment are identical and taken from [10]. Both the target- and disturbance function are quasi-random appearing multi-sines. The random appearance prevents the human operator (HO) from recognizing the signal and applying precognitive feedforward behaviour [25]. Furthermore, the use of multi-sines ensures high signal-to-noise-ratios (SNRs) at the measurement frequencies.

Then, (9) shows how the forcing functions are formed. Here, N_f equals the number of frequencies of the multi-sine (the same as in (1)), while A_k, ω_k and ϕ_k represent the amplitude, frequency and phase of each sine, respectively.

$$f(t) = \sum_{k=1}^{N_f} A_k \sin(\omega_k + \phi_k) \tag{9}$$

The signals used in this research are based on those used in [10], [16]. Fig. 4 shows both the target function f_t and the disturbance function f_d in the time- and frequency domain. For both signals, N_f is equal to 10 and after the bandwidth ω_i at 1.5 rad/s (for $k = 6$), the signal amplitudes are attenuated by 20 dB. The standard deviations for the target- and disturbance function are equal to 0.5 and 0.2 inch, respectively. Besides, the frequencies ω_k are integer multiples of the base-frequency $\omega_b = 0.0524$ rad/s, corresponding to a measurement time of 120 s. As the signals are also used for the human-in-the-loop experiments, 5 different phase realizations are made for the target function. This prevents the HO from memorizing the target signal and guarantees random appearance [25]. As the disturbance function is not visible during tracking and is thus unlikely to be memorized, a single realization is deemed sufficient.



(a) (b)

Fig. 4. Part of the time trace (a) and spectrum (b) of one of the forcing functions used for the simulations (other realizations had different phase sets).

2) HO Models

To generate the simulated time series for identification, reference values for the HO model parameters are selected, based on experiment results obtained in [9], [10]. Parameter fits for SI and DI experiments with a preview time $\tau_p = 2s$ are selected. Table II provides the CE dynamics and the HO model parameters for both cases.

TABLE II
OVERVIEW OF THE MODEL PARAMETERS USED IN THE SIMULATIONS.

CE	H_{ce} [−]	$K_{e^\star}$ [-]	$T_{L,e^\star}$ [s]	τ_v [s]	ω_{nms} [rad/s]	ζ_{nms} [-]	K_f [-]	τ_f [s]	$T_{l,f}$ [s]
SI	$\frac{1.5}{(j\omega)}$	1.25	–	0.23	13.0	0.13	1.0	0.60	0.15
DI	$\frac{5}{(j\omega)^2}$	0.33	1.27	0.33	7.4	0.13	0.95	1.05	1.15

3) Remnant Modelling

As described in Section II, the remnant inserted in the model accounts for non-linear behaviour, time varying behaviour and motor noise [16], [21]. Fig. 2 already showed how the remnant is inserted into the model: as low-pass filtered Gaussian white noise, inserted at the system output $x(t)$ input to the HO.

The low-pass remnant filter contains two parameters, the remnant gain K_r and the remnant lag-time constant $T_{l,r}$. As described in [20], K_r, together with the white noise intensity, defines the desired noise level at the control output u. This noise ratio (NR) is defined as $\frac{\sigma^2_{u_n}}{\sigma^2_u}$, or the part of the control signal that is caused by the remnant. Experiments have shown that this ratio depends on the controlled element dynamics and the target signal bandwidth. For a target function bandwidth of 1.5 rad/s, this ratio is around 0.3 for SI dynamics and 0.6-0.7 for DI dynamics [20].

Besides the noise level, the remnant lag-time constant also depends on the CE dynamics. In the simulations performed, the remnant lag-time constant is set to 12 s for DI dynamics and 0.2 for SI dynamics. For the noise levels, a number of ratios are selected for $\frac{\sigma^2_{u_n}}{\sigma^2_u}$. These are $0.0, 0.25, 0.5$ and 0.75, similar to the levels evaluated in [11]. This is done to compare the different methods to noise sensitivity.

4) Model Parameter Limits and Candidate Pools

The next point involves the parameter limits applied to the identification methods. For the system identification (SSID) technique, these are essential, as the method uses a candidate pool approach. The parameter limits in combination with the selected candidate pool density then determine the size of the

candidate pool, which directly relates to the computational cost of the algorithm [23]. For the analysis in this research, the parameter limits from earlier work by [11] are selected. Table III shows these limits on the top half, which were selected based on earlier results in [9], [10], [15]. Here, these limits are applied to all identification techniques, to keep the candidate models the same.

Although the parameter limits are the same for all methods, the candidate models will vary slightly when using SSID and the Nelder-Mead (NM) simplex optimizers. After all, SSID uses a candidate pool approach and thus offers a finite amount of candidate models, while the NM simplex methods are not constrained in this sense. The second half of Table III therefore shows the step sizes and the number of steps used for each parameter, n_θ. With these steps, an unfiltered candidate pool for DI dynamics of about 2.6 billion elements is formed, which is reduced to just over 1 billion after selecting only the stable candidates. For SI dynamics, this number becomes around 70 million, as one parameter ($T_{L,e^\star}$) is omitted.

TABLE III
PARAMETER LIMITS FOR THE DIFFERENT HO MODEL PARAMETERS (TOP) AND THE CANDIDATE POOL STEP SIZES FOR SSID (BOTTOM).

		$K_{e^\star}$	$T_{L,e^\star}$	τ_v	ω_{nms}	ζ_{nms}	K_f	τ_f	$T_{l,f}$
Limits	max	2	3	0.4	16	1	2	2	1.6
	min	0.1	0.6	0.1	4	0.05	0.1	0.3	0.1
SSID	n_θ	72	36	36	48	24	-	36	24
	step size	0.03	0.07	0.01	0.26	0.04	-	0.05	0.07

5) Simulation Approach

To compare the performance of the different identification methods, simulations with different noise levels are performed. 40 different realizations are performed for each type of CE dynamics and each noise level. Each realization consists of 5 runs of 120s, of which the frequency domain signals are averaged after correcting for the target phase. This procedure is based on experiments described in [20]. Then, both the spread in the cost function data and the quality of the parameter estimates is evaluated.

6) Dependent Measures

The spread in the cost function data is expressed in terms of the frequency-averaged standard deviation of the estimated describing functions (DF). The weighted standard deviation is calculated over the 5 different target realizations, see (10), where ρ_i is the signal-to-noise ratio (SNR) of the power spectra $1 - \frac{S_{uu_n}}{S_{uu}}$. This weighting factor is applied to correct for the estimated noise levels at the data points. Furthermore, N_{runs} is the number of runs performed for a single estimation, which is equal to five. Then, the frequency-averaged spread of the DFs is calculated using (11). Here, N_{DF} represents the number of estimated data points of the describing function.

$$\sigma_k(j\omega_k) = \sqrt{\frac{\sum_{i=1}^{N_{runs}} \rho_i \left(\hat{H}_i(j\omega_k) - \hat{\mu}_{H(j\omega_k)}\right)^2}{N_{runs}}} \tag{10}$$

$$\mu_{\sigma_{DF}} = \frac{1}{N_{DF}} \sum_{k=1}^{N_{DF}} \sigma_k(j\omega_k) \tag{11}$$

For the parameter estimates, both the individual parameters as well as a more generic metric, the mean absolute error (MAE), will be provided. The MAE is defined according to (12), where n_{pars} is the number of parameters that are estimated. Here, $\hat{\theta}$ is the estimated parameter, while θ is the true parameter.

$$\text{MAE} = \frac{1}{n_{pars}} \sum_{i=1}^{n_{pars}} \frac{\left|\hat{\theta}_i - \theta_i\right|}{\theta_i} \cdot 100\% \tag{12}$$

Finally, a last set of performance metric is formed by calculating the offset between the estimated- and the true dynamics of the open and the closed-loop. For the open-loop dynamics (or subsystems), these are $\hat{H}_{o_t}(j\omega_k|\theta)$ and $\hat{H}_{o_x}(j\omega_k|\theta)$ and the true subsystems $H_{o_t}(j\omega_k)$ and $H_{o_x}(j\omega_k)$. This open-loop frequency domain error ϵ_{OL} is calculated according to (13). For the closed-loop, the difference between $\hat{H}_{cl_t}(j\omega_k|\theta)$ and $H_{cl_t}(j\omega_k)$ is evaluated using (14).

$$\epsilon_{OL} = \sum_{k=1}^{2N_f} \left(\hat{H}_{o_t}(j\omega_k|\theta) - H_{o_t}(j\omega_k)\right)^2 + \left(\hat{H}_{o_x}(j\omega_k|\theta) - H_{o_x}(j\omega_k)\right)^2 \tag{13}$$

$$\epsilon_{CL} = \sum_{k=1}^{N_f} \left(\hat{H}_{cl_t}(j\omega_k|\theta) - H_{cl_t}(j\omega_k)\right)^2 \tag{14}$$

C. Human-in-the-Loop Experiment

To supplement- and validate the simulation results, a human-in-the-loop experiment was performed. The experiment was performed in the fixed-base simulator in the Human-Machine Interaction Laboratory at Delft University of Technology. Four subjects (all students or staff of TU Delft) were seated in front of a display and performed the preview tracking task using an electro-hydraulic side-stick. The exact setup is identical to the experiments described in [9], [10], [15].

1) Independent Variables

The independent variables in this experiment were the number of forcing functions (f_t only or f_t and f_d) and the controlled element dynamics (SI or DI), just as in the simulations. The forcing functions were the same as in the simulations, see Section IV-B1. For the f_t, 5 realizations were made. In the experiment, the forcing function was always present, while the disturbance function would be either switched on or off. The CE dynamics were the same as in Table II.

2) Experiment Procedure

During the experiment, participants were instructed to minimize the tracking error during the run, of which the RMS value was reported after each run. The experiment was setup to be a within-subjects experiment, and subjects performed the four different conditions in a randomized order as defined through a Latin Square setup. Per condition, tracking runs were performed until steady performance (in terms of error- and control output variance) was reached for 5 consecutive runs. All runs prior to the five runs with steady performance were considered to be training data. A single run takes 128 s, of which the first 8 seconds were considered to be run-in

time. The entire experiment took about 2 hours per subject and breaks were taken between conditions.

3) Dependent Variables

Just as with the simulations, a number of measures are used to quantify the performance of the different methods. The spread in the cost function data and the parameter estimates are compared again, although the mean absolute error cannot be evaluated as the 'true' values are unavailable for real HO data. Furthermore, the variance-accounted-for (VAF), a measure of similarity between two signals, is calculated. The VAF quantifies the difference between the measured signal $U(j\omega)$ and the modelled signal $U_{mod}(j\omega)$, which is determined from $U_{mod}(j\omega) = H_{o_t}(j\omega)F_t(j\omega) - H_{o_x}(j\omega)X(j\omega)$. (15) describes how the VAF is calculated, with $\epsilon_u = U(j\omega) - U_{mod}(j\omega)$. The signals $U(j\omega)$ and $U_{mod}(j\omega)$ are the frequency-domain signals averaged over the 5 runs.

$$\text{VAF} = (1 - \frac{\sigma^2_{\epsilon_u}}{\sigma^2_u}) \times 100\% \tag{15}$$

V. Results

The results of both the simulations and the experiments are discussed in this section. If similar features are evaluated for both the offline simulations and the experiment, these are presented and discussed together. For all methods, the accuracy of the cost function data, the identification results — in terms of parameter estimates (MAE and actual estimates), variance-accounted-for (VAF) and the frequency domain error ϵ_{OL} — and the computation times are evaluated.

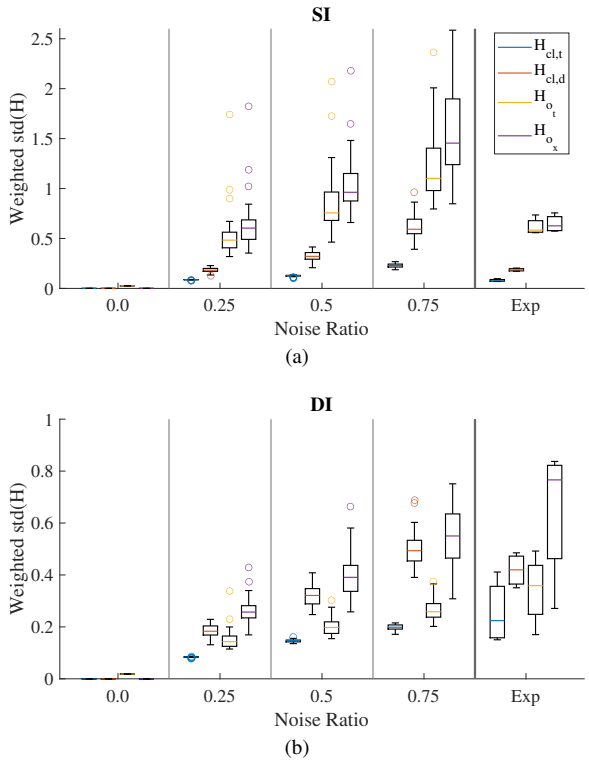


Fig. 5. Weighted spread in the describing function data of the closed- and open-loop for the different noise levels of the simulations and the experiment data. Results are shown for SI (a) and DI (b) dynamics.

A. Spread of DF Data

Fig. 5 shows the frequency-averaged, weighted standard deviation of the simulation- and the experiment data for both dynamics (see Section IV-B5). The two figures show how the spread of the closed-loop and the subsystem describing functions (DFs) increases with the remnant noise level, and how these relate to the results measured in the experiment. The data are taken from the runs with two forcing functions only, as these allow for estimation of all four DFs (instead of just $H_{cl,t}$). As the data of the control output $U(j\omega)$ is a different type then the describing functions, it is omitted from this comparison.

First, the general trend in the noise levels is evaluated. For SI dynamics, the levels measured in the experiment roughly corresponds to a noise ratio (NR) $\sigma^2_{u_n}/\sigma^2_u$ of 0.25. For the DI dynamics, the experimental results are similar to an NR of 0.75. This corresponds to the noise levels found when estimating the NRs from the experiment data, which are around 0.2 and 0.7 for SI and DI dynamics, respectively. These numbers are in range with the levels found in [20] (0.3 for SI and 0.65 for DI).

When comparing the spread of the different DFs, it is observed how, for SI dynamics, the subsystem dynamics H_{o_t} and H_{o_x} show an increased spread compared to the closed-loop DFs $H_{cl,t}$ and $H_{cl,d}$ by a factor of between 2 and 5. For DI dynamics, this difference is smaller, as especially the spread in $H_{cl,d}$ is larger than that found for H_{o_t}. Furthermore, the difference between the two subsystem dynamics H_{o_t} and H_{o_x} is larger for DI dynamics, as the spread in H_{o_t} is about half that of H_{o_x}, and is almost as low as the spread in $H_{cl,t}$. As the interpolation step is the main difference between the closed-loop and the open-loop DF, these results indicate that this operation increases the spread in the DFs.

B. Identification Results

The identification results consist of two types of results. First, the parameter estimates from the offline simulations and the human-in-the-loop experiment are shown, including the mean absolute error (MAE) from the offline simulations. Then, the variance-accounted-for (VAF) is evaluated for the experiment data.

1) Mean Absolute Error

Fig. 6 gives a general overview of the performance of the different identification techniques by showing the MAE for the different tested noise ratios. From the six different methods, the method $U2_{td}$ (with a cost function at $U(j\omega)$ and two forcing functions) shows the lowest mean error of around 10% for an NR = 0.75 for both SI and DI dynamics. Besides the low error, the spread for this method caused by the different noise realizations is also the smallest (all MAEs lie within 20%).

The other methods show mixed results depending on the types of dynamics. The OLDF and $U1_t$ methods show similar performance, which is slightly worse then $U2_{td}$ (less than 10% for SI dynamics at NR = 0.25 and less than 40% for DI at NR = 0.75). Compared to OLDF and $U1_t$, the $CLDF_{td}$ method

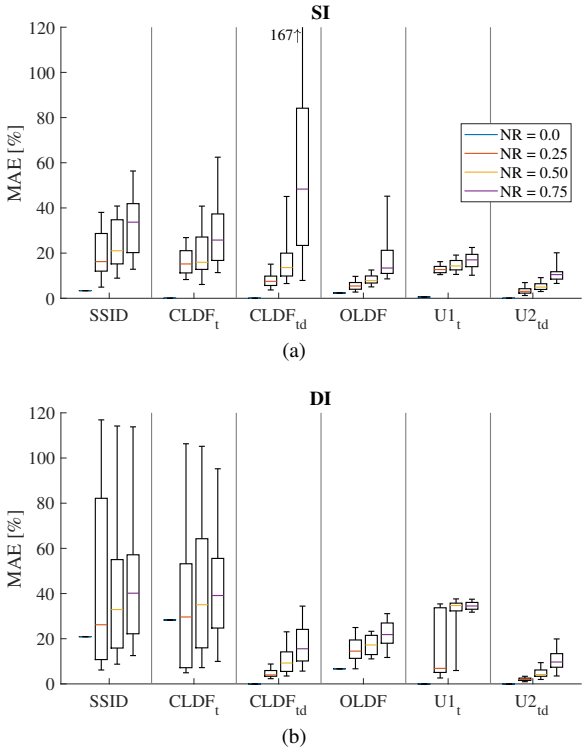


Fig. 6. Mean absolute error (MAE) of the different methods for the evaluated noise ratios. Results are shown for single- (a) and double integrator (b) dynamics.

only performs worse at the higher noise ratios in SI dynamics, where generally levels of up to 0.4 are found [20].

The SSID and CLDF$_t$ methods show the worst performance, especially for DI dynamics. There, MAEs of over 50% are obtained for all nonzero noise ratios. Even for noiseless data, the two methods cannot get the correct fit of the data, as MAEs of 20% and 30% are observed for SSID and CLDF$_t$, respectively. For SI dynamics, all methods have an MAE of below 40% at the noise levels found in real experiments in SI tasks (NR = 0.25).

2) Parameter Estimates

To look at the performance of the methods in more detail, individual parameter fits are evaluated. Figs. 7 and 8 show the estimates of the parameters of $H_{o_{cmp}}$ and H_{o_p} from the simulations and Fig. 9 shows the experimentally obtained estimates.

First, Fig. 7 shows how the methods using the closed-loop DF (SSID, CLDF$_t$ and CLDF$_{td}$) show considerable errors for the estimates of τ_v and the parameters of the neuromuscular system (NMS). For the DI data, these errors are more apparent than for SI dynamics. For DI, the methods that fit on the open-loop DFs and the control output signal show relatively larger errors in the estimates of $T_{L,e^\star}$. However, the CLDF methods (including SSID) also have a large spread for the estimates of $T_{L,e^\star}$, indicating difficulty to estimate this parameter.

For the feedforward parameters that form H_{o_p}, Fig. 8 shows that estimates appear to be accurate for all methods. Errors increase with noise levels for OLDF and U1$_t$ in the estimates of $T_{l,f}$, however.

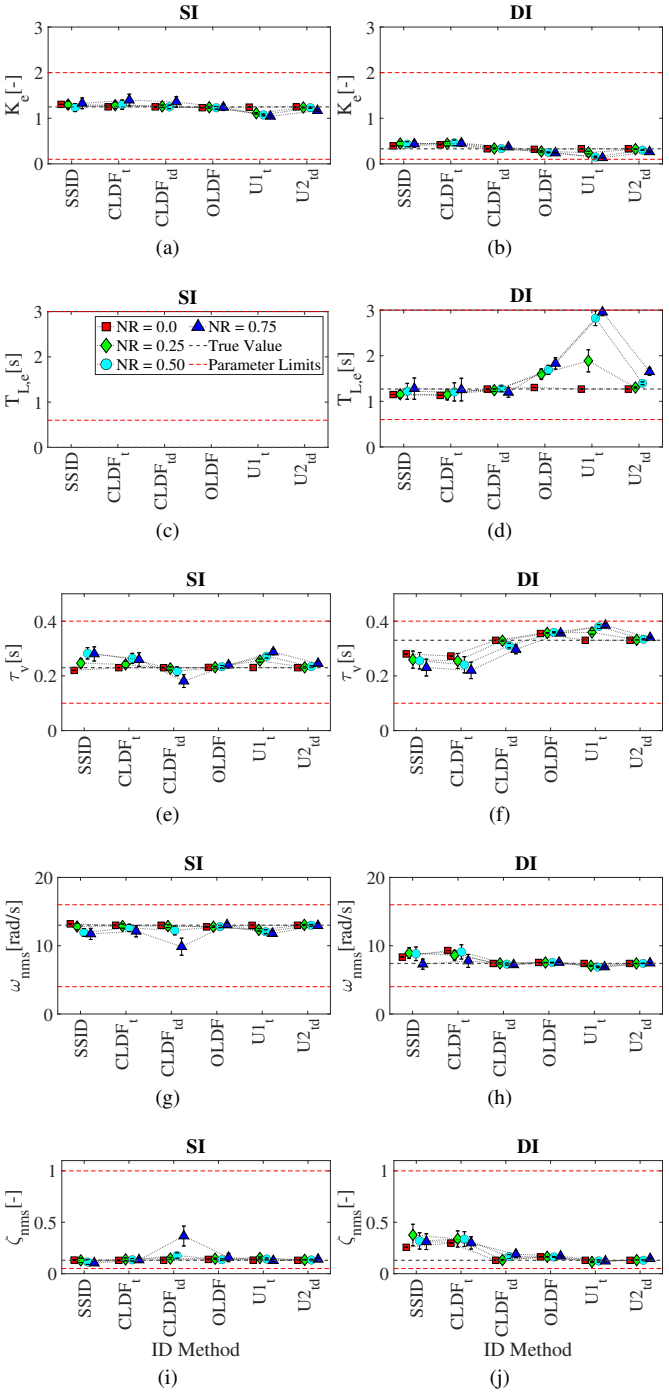


Fig. 7. Parameter estimates of the feedback subsystem $H_{o_{cmp}}$ for SI (a, c, e, g & i) and DI (b, d, f, h & j) CE dynamics. Errorbars indicate the 95% confindence interval.

Finally, when evaluating the experimental parameter estimates, Fig. 9 shows the fits, including the parameter limits and the values used for the simulations. When focusing on the SI dynamics, Fig. 9a shows how different fits are obtained when comparing the methods that use a single forcing function with those that use two forcing functions. Furthermore, the methods that use two forcing functions have a smaller spread of the estimates. The only exception here is found in the estimate of $T_{l,f}$, where for CLDF$_{td}$ and OLDF, the spread is larger than

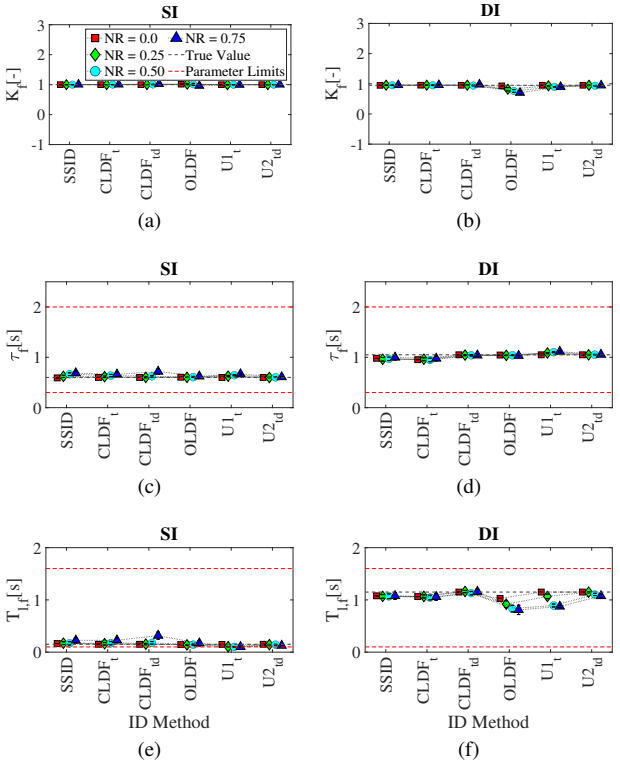


Fig. 8. Parameter estimates of the feedforward subsystem H_{o_p} for SI (a, c & e) and DI (b, d & f) CE dynamics. Errorbars indicate the 95% confindence interval.

that of the f_t only methods.

For the DI parameter estimates, the number of forcing functions does not appear to have a large effect on the estimates. Rather, the cost function data appears to influence the estimates, as the methods that estimate the parameters from the closed-loop describing functions show similar parameter estimates. This is especially visible in the parameters of $H_{o_{cmp}}$, i.e., $T_{L,e^\star}$, τ_v and to lesser extent ω_{nms}.

A last remark regarding the DI parameter estimates shown in Fig. 9b involves the estimates of the $U2_{td}$ method. As the experiment for the DI case was identical to the one performed in [10], it is interesting to see that the parameter estimates correspond well to the ones from that experiment (which are shown as the simulation reference in Fig. 9b). This result indicates successful experiment replication.

3) Variance-Accounted-For

Fig. 10 shows the VAFs obtained from the experiment results. The figure shows how for SI dynamics, the $CLDF_{td}$ and OLDF methods have a spread in the VAFs of 20% and 70%, respectively. On the other hand, the VAFs of the other methods are all around 95 %. For DI dynamics, the VAFs are all approximately between 60 and 80%, except for the $U2_{td}$ method, for which the VAFs are in the range of 80-90%. This result should not be surprising, as the $U2_{td}$ and the $U1_t$ methods are optimizing on the same signal as the VAF, namely the frequency domain control output $U(j\omega)$.

4) Frequency Domain Error

Fig. 11 shows the frequency-domain estimation errors ϵ_{OL} and ϵ_{CL}, the error between the true and the estimated open- and closed-loop dynamics, obtained with the simulations. Figs. 11a and 11b clearly show how, for both SI and DI dynamics, the OLDF and $U2_{td}$ are superior compared to the other methods when looking at ϵ_{OL}. For these methods, the mean error does not exceed 20 (for both dynamics). On the other hand, the methods that use the closed-loop dynamics and a f_t only have an average ϵ_{OL} of over 50 for SI- and over 300 for DI dynamics.

Looking at the closed loop dynamics in Figs. 11c and 11d, it is observed that the OLDF methods has the highest closed-loop error. As the scales that are used for ϵ_{OL} and ϵ_{CL} differ by a factor of around 1,000 (due to a different number of data points and different dynamics), the values of ϵ_{OL} and ϵ_{CL} cannot be compared directly.

It should be noted that the metrics used for ϵ_{OL} and ϵ_{CL} are biased towards OLDF and $CLDF_t$, respectively. Both methods use practically (weigthing is added) the same cost function as this metric. This is similar to the observation made regarding $U2_{td}$ when evaluating the VAF results. However, it is remarkable to see that, even with different metrics, the $U2_{td}$ method performs best.

To illustrate the difference in the errors observed between the OLDF and $U2_{td}$ methods on the one hand, and SSID and $CLDF_t$ on the other, Fig. 12 shows the estimated dynamics for one realization of the DI simulations with a noise ratio of 0.75. Besides the estimates, the FRFs and the true dynamics are included. The figure clearly shows how the open-loop dynamics show larger offsets for SSID and $CLDF_t$. This difference is most clearly visible at frequencies above 5 radians per second. Looking at the closed-loop in Fig. 12a, the OLDF appears to approximate the true dynamics accurately, although at frequencies below 1 rad/s the magnitude is underestimated. This difference is, however, not as clear as the described difference of SSID and $CLDF_t$.

C. Computation Times

Finally, Fig. 13 shows the box plots containing the computational times of the different algorithms. Here, all simulation results are combined (thus all noise levels are shown together), as the noise levels do not affect the computation time. The figure shows how all methods except SSID, which use a Nelder-Mead Simplex optimizer, take between 10 and 100 seconds to run. The SSID algorithm on the other hand, takes significantly longer: around 1,000 s for SI and up to 20,000 s for DI dynamics. Moreover, this number is dependent on the candidate pool density chosen for SSID.

VI. Discussion

This research aimed to compare the performance of different system identification methods used to estimate linear time-invariant (LTI) human feedback- and feedforward control behaviour in preview tracking tasks. For both single- and double integrator (SI- and DI) dynamics, offline simulations and a human-in-the-loop experiment provided data to compare 6 different frequency domain identification methods. Some of these methods had previously been used in other research,

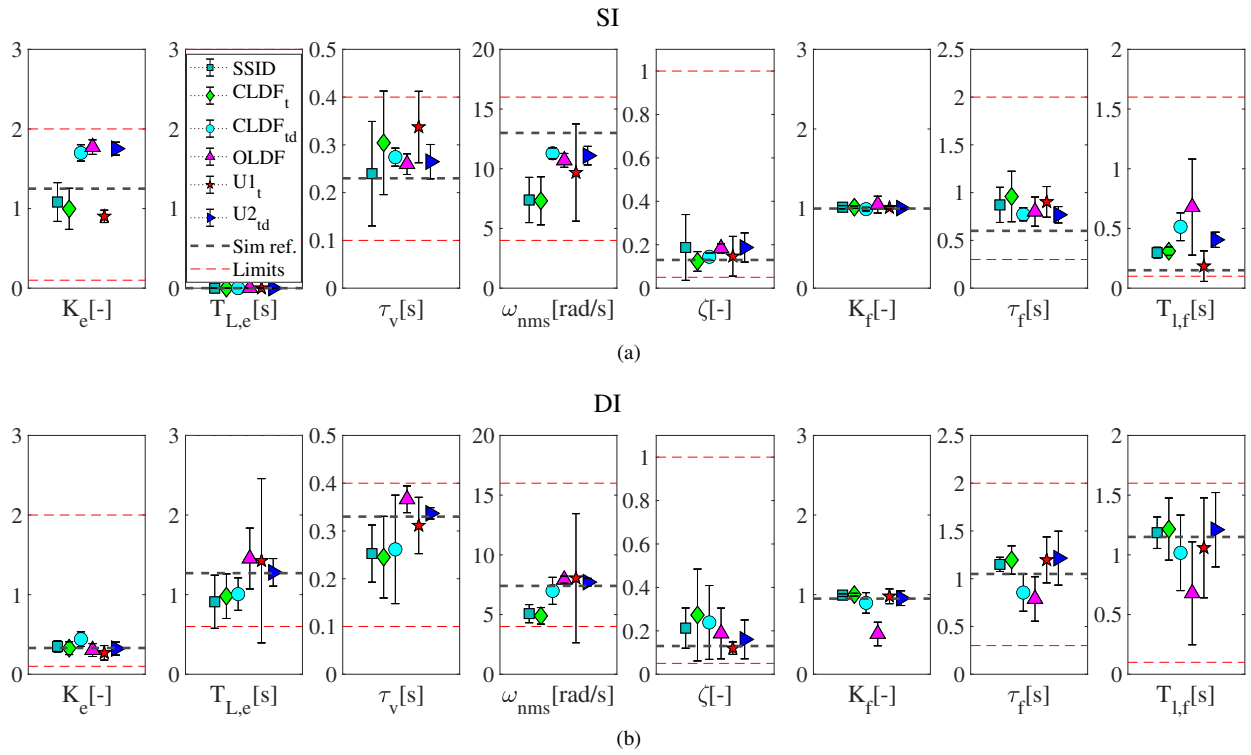


Fig. 9. Parameter estimates of the experiment obtained with the evaluated methods for both SI- (a) and DI (b) dynamics.

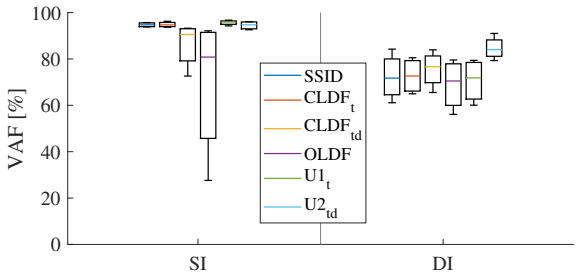


Fig. 10. Box plots showing the VAFs of the experiment results for both types of dynamics and all methods.

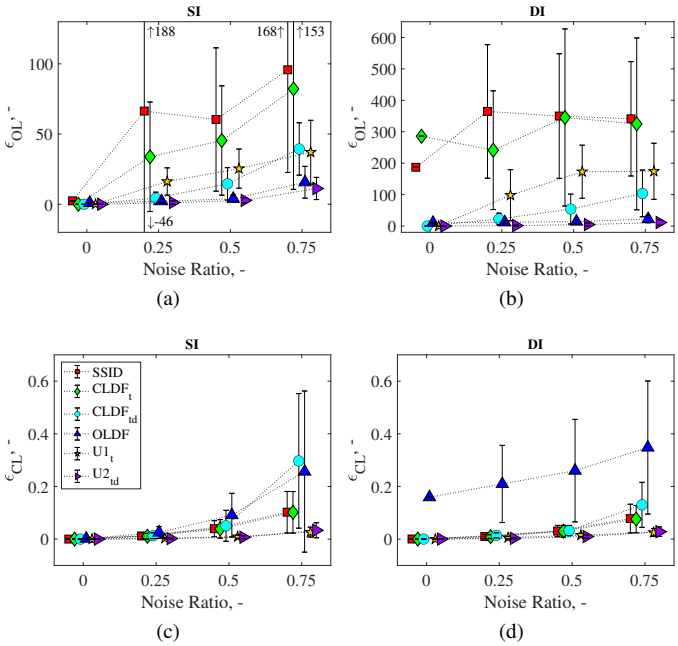


Fig. 11. Frequency domain error of the estimated and the true subsystems H_{o_t} and H_{o_x} (a and b) and closed-loop dynamics $H_{cl,t}$. Results are shown for single- (a and c) and double integrator (b and d) dynamics, with errorbars indicating the mean and 95% confidence intervals.

while others were first used in this research to function to facilitate a step-wise comparison.

Hypothesis H1 stated that the human operator (HO) model parameters may not be estimated accurately when closed-loop describing function (DF) data are used for the cost function. This hypothesis is confirmed, as both the SSID and the CLDF$_t$ methods are unable to estimate the parameters in DI dynamics, even for noiseless data. The CLDF$_{td}$ method, on the other hand, is able to estimate both subsystem dynamics (H_{o_p} and $H_{o_{cmp}}$) accurately for noiseless data, although its performance is worse than that of the U2$_{td}$ method (20% for the MAE and for the VAF, and about 10 times as bad for ϵ_{OL}). Estimation errors are largest in the estimation of the feedback time delay τ_v ($\pm$50%) and the parameters of the neuromuscular system (NMS) ζ_{nms} ($\pm$100%) and ω_{nms} ($\pm$10%). Especially the results of the frequency-domain errors and the corresponding bode plots, clearly illustrate that the high-frequency control behaviour cannot be estimated accurately when only using the closed-loop DF. When using forcing functions with power up to lower frequencies (e.g., 3.1 and 4.4 rad/s [12] and [23], respectively), the effects of the NMS are not as apparent and the problem becomes smaller.

For SI dynamics, SSID and CLDF$_t$ can estimate the param-

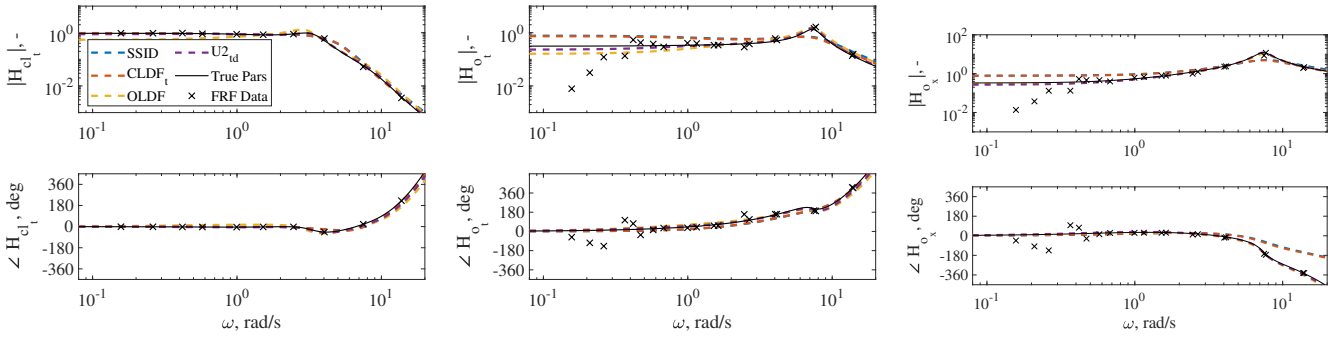


Fig. 12. Frequency domain functions of the closed-loop (H_{cl}) and open-loop (H_{o_t} and H_{o_x}) of a single realization with a noise ratio of 0.75 for DI dynamics and four of the evaluated methods. Dynamics of the true parameters and the describing function estimates are also shown.

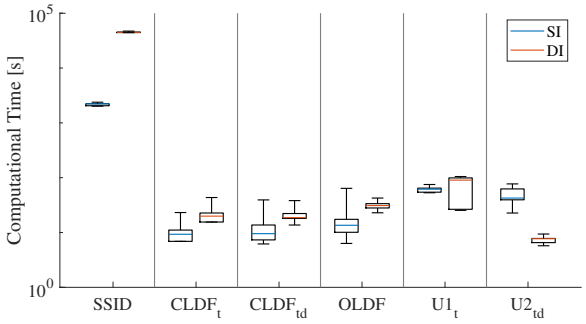


Fig. 13. Box plots illustrating the computation times of the different algorithms, obtained in the simulations.

eters for noiseless data, but the estimation error of τ_v also approaches 50%. The estimation errors of the CLDF_{td} method reach MAEs up to 200% for SI parameters at a noise ratio of 0.75, although it is noted that such a noise ratio generally does not occur in real data for SI tracking tasks.

Hypothesis H2, regarding the performance of the methods that use a second forcing function, is confirmed. The parameter estimates from the simulations clearly showed that the methods that use two forcing functions give smaller estimation errors, with MAEs being up to 400% larger for such methods (e.g., comparing the MAE of U1_{t} and U2_{td} for DI and NR = 0.75). Looking at the methods that use DFs for the cost function, both CLDF_{td} and OLDF give better results than SSID and CLDF_{t}, and U2_{td} performs better than U1_{t}. However, the results also clearly show that using a cost function at the control output signal leads to the best performance, as U2_{td} performs best for all conditions and all metrics (MAE of less than 10% for DI, less than 5% for SI for realistic noise levels). With respect to the CLDF_{td} method, this difference is explained by the information regarding the relative strength of f_t and f_d, that is present in $U(j\omega)$, but is not available in the CLDF methods. There, the lower power in $X(j\omega_d)$ is canceled out by the low power in $F_d(j\omega)$.

When looking at the OLDF method on the other hand, part of this information regarding the relative strength of the two forcing functions is transferred to the open-loop DFs, FRFs still do not contain information about the relative magnitudes of f_t and f_d. Furthermore, another aspect that might explain the performance difference with respect to U2_{td} — 5 to 10% for the MAE for SI and DI dynamics, respectively, about 20% for the VAFs, and an increased closed-loop error ϵ_{CL} of at least a factor 5 — is the fact that interpolation errors are present in the cost function data of OLDF. Both the simulation- and the experimental results show that the spread in the DFs of the subsystem dynamics is larger than that of the closed-loop dynamics. The combination of these two observations (the increased spread in the open-loop DFs and the lower performance of the OLDF method) indicates that hypothesis H3 can be confirmed. However, as the results in this paper do not directly show that the presence of the relative strength of the forcing functions in the cost function data lead to an increased performance of the U2_{td} method, this hypothesis can only be partly confirmed.

Besides the three hypotheses, a number of other insights was gained from the research. First, although the different optimization algorithms used by SSID and CLDF_{t} resulted in similar levels of the VAF (about 95% for SI and 75% for DI) and for the MAE, large differences were observed in the computation time required for both methods. Here, the SSID method takes between 100 to 1,000 times longer.

Although the focus of this research lies on the identification methods and not on human control behaviour itself, it should be noted that clear differences in the SI experiment results were observed when evaluating the methods that used a different number of forcing functions. This indicates that, especially for SI dynamics, the disturbance function does affect human control behaviour. Considering the fact that the estimation results of all methods were decent (maximum MAE of approximately 15%) for SI at a (realistic) noise ratio of 0.25, this offset does not appear to be caused by errors in the estimation methods. It could therefore be interesting to include the effect of the disturbance function on human control behaviour in the existing theory. This would require a more extensive experiment, however. For now, more information regarding the tracking performance (i.e., the cross-over frequencies and the phase margins) are found in the appendix of this thesis.

As mentioned earlier, this research compared 6 different methods, of which three (CLDF_{t}, CLDF_{td} and U1_{t}) had not been used in earlier work. In the end, an existing identification method, U2_{td}, performed best in all performance metrics

(MAE, VAF and ϵ_{OL}), meaning that the three new methods are not recommended to be used again in future work. However, the methods did allow for a step-wise comparison of the effects of the optimization algorithm, the number of forcing functions and the cost function data, thus proving their value for this research. In case only a single forcing function is available, on the other hand, the $U1_t$ method is recommended to be used, as it performed better compared to CL methods (about 50% to 100% better for ϵ_{OL} and about 10% to 100% better in terms of the MAE).

For future research into preview behaviour, it is nevertheless recommended to use the $U2_{td}$ method, as it offers the best performance in both estimation accuracy (MAEs of below 10%, VAFs of at least 80%) and computational times (up to 100 seconds per estimation). Another advantage of this method is that it does not require strict parameter limits, as computational time is independent of these limits and the respective step sizes. Although other artefacts in the parameter estimations might occur when loosening these limits, a possibility to solve this is by fitting on all frequencies (including the remnant frequencies) of the control output signal (as done in [10]). Performance of this method can be evaluated in the Appendix of this thesis.

A last recommendation for future work involves the analysis performed in this research applied to a different tracking tasks with feedforward behaviour, such as [5], [13], [14]. As this research focused on a preview tracking task, performance of the different methods might differ when applied to pursuit data. As the HO model requires fewer parameters for such a task, estimation performance of the single forcing function methods might improve. As such tasks have smaller feedforward components, it can be easier to estimate two responses from a single set of data. Looking even further, such an analysis can also be performed in other multi-loop tracking tasks, with motion feedback instead of feedforward behaviour, see for example [26]. Although this involves different model structures and thus potentially different techniques (see the explanation on SSID regarding this in [11]), it is interesting to see whether behaviour can be accurately using e.g., a single forcing function in such tracking tasks.

VII. Conclusion

This paper compared six different system identification techniques, used to identify linear time-invariant human feedback- and feedforward control behaviour. The results showed how, especially for double integrator dynamics, the identifiability of the system is compromised when using a single forcing function and only closed-loop describing function data for the cost function. On the other hand, the best estimation results — a mean absolute error (MAE) less than 10 % and a VAF above 80% — are obtained when using two forcing functions and the frequency domain control output signal as cost function data. When using a single forcing function, using the control output signal as the cost function data also gives the best results (maximum MAE of 35 % and a minimum VAF of 70%). Regarding the algorithm, the non-linear Nelder-Mead Simplex algorithm was both 100 to 1,000 times faster and less constraining then the hybrid grid search used by subsystem identification (SSID). A second forcing function is thus still required when accurately identifying multi-loop behaviour in preview tracking tasks.

References

[1] M. Mulder, D. M. Pool, D. A. Abbink, E. R. Boer, P. M. T. Zaal, F. M. Drop, K. van der El, and M. M. van Paassen, "Manual Control Cybernetics: State-of-the-Art and Current Trends," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 468–485, Oct. 2018.

[2] D. T. McRuer and H. R. Jex, "A Review of Quasi-Linear Pilot Models," *IEEE Transactions on Human Factors in Electronics*, vol. HFE-8, no. 3, pp. 231–249, Sep. 1967.

[3] T. P. Neal and R. E. Smith, "A Flying Qualities Criterion for the Design of Fighter Flight-Control Systems," *Journal of Aircraft*, vol. 8, no. 10, pp. 803–809, Oct. 1971.

[4] R. A. Hess and K. K. Chan, "Preview Control Pilot Model for Near-Earth Maneuvering Helicopter Flight," *Journal of Guidance, Control, and Dynamics*, vol. 11, no. 2, pp. 146–152, 1988.

[5] X. Zhang and J. B. Hoagg, "Frequency-Domain Subsystem Identification with Application to Modeling Human Control Behavior," *Systems & Control Letters*, vol. 87, Jan. 2016.

[6] S. A. S. Mousavi, X. Zhang, T. Seigler, and J. B. Hoagg, "Subsystem identification of feedback and feedforward systems with time delay," *Results in Control and Optimization*, vol. 1, p. 100002, 2020.

[7] K. van der El, D. M. Pool, H. J. Damveld, M. M. van Paassen, and M. Mulder, "An Empirical Human Controller Model for Preview Tracking Tasks," *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2609–2621, Nov. 2016.

[8] M. Yamagami, D. Howell, E. Roth, and S. A. Burden, "Contributions of Feedforward and Feedback Control in a Manual Trajectory-Tracking task," in *Proceedings of the 2nd IFAC Conference on Cyber-Physical & Human-Systems, Miami (FL)*, ser. IFAC-PapersOnLine, vol. 51, no. 34, 2019, pp. 61–66.

[9] K. van der El, S. Padmos, D. M. Pool, M. M. van Paassen, and M. Mulder, "Effects of Preview Time in Manual Tracking Tasks," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 486–495, Oct. 2018.

[10] K. van der El, D. M. Pool, M. M. van Paassen, and M. Mulder, "Effects of Target Trajectory Bandwidth on Manual Control Behavior in Pursuit and Preview Tracking," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 1, 2020.

[11] P. J. C. Bentinck, "On the Application of a Subsystem Identification Technique to a Manual Control Pursuit- and Preview Model," Unpublished paper, Delft University of Technology, 2021.

[12] A. J. S. Sheffler, S. A. S. Mousavi, E. Hellström, M. Jankovic, M. A. Santillo, T. M. Seigler, and J. B. Hoagg, "Effects of Reference-Command Preview as Humans Learn to Control Dynamic Systems," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Oct. 2019, pp. 4199–4204.

[13] F. M. Drop, R. J. De Vries, M. Mulder, and H. H. Bülthoff, "The Predictability of a Target Signal Affects Manual Feedforward Control," in *Proceedings of the 13th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, Kyoto, Japan*, 2016.

[14] R. W. Pew, J. C. Duffendack, and L. K. Fensch, "Sine-Wave Tracking Revisited," *IEEE Transactions on Human Factors in Electronics*, vol. HFE-8, no. 2, pp. 130–134, Jun. 1967.

[15] K. van der El, D. M. Pool, M. M. van Paassen, and M. Mulder, "Effects of Preview on Human Control Behavior in Tracking Tasks With Various Controlled Elements," *IEEE Transactions on Cybernetics*, vol. 48, no. 4, pp. 1242–1252, Apr. 2018.

[16] D. T. McRuer, D. Graham, E. S. Krendel, and W. J. Reisener, "Human Pilot Dynamics in Compensatory Systems: Theory, Models, and Experiments with Controlled Element and Forcing Function Variations," Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base (OH), Tech. Rep. AFFDL-TR-65-15, 1965.

[17] R. L. Stapleford, D. T. McRuer, and R. E. Magdaleno, "Pilot Describing Function Measurements in a Multiloop Task," *IEEE Transactions on Human Factors in Electronics*, vol. 8, no. 2, pp. 113–125, Jun. 1967.

[18] M. M. van Paassen and M. Mulder, "Identification of Human Operator Control Behaviour in Multiple-Loop Tracking Tasks," in *Proceedings of the Seventh IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, Kyoto Japan*. Pergamon, 1998, pp. 515–520.

[19] K. van der El, "How Humans Use Preview Information in Manual Control," Ph.D. dissertation, Aerospace Engineering, Delft University of Technology, Dec. 2018.

[20] K. van der El, D. M. Pool, and M. Mulder, "Analysis of Human Remnant in Pursuit and Preview Tracking Tasks," in *Proceedings of the 14th IFAC Symposium on Analysis Design and Evaluation of Human Machine Systems Tallinn, Estonia*, 2019, pp. 145–150.

[21] W. H. Levison, S. Baron, and D. L. Kleinman, "A Model for Human Controller Remnant," *IEEE Transactions on Man-Machine Systems*, vol. 10, no. 4, pp. 101–108, Dec. 1969.

[22] X. Zhang and J. B. Hoagg, "Subsystem Identification of Multivariable Feedback and Feedforward Systems"," *Automatica*, vol. 72, pp. 131 – 137, 2016.

[23] X. Zhang, S. Wang, J. B. Hoagg, and T. M. Seigler, "The Roles of Feedback and Feedforward as Humans Learn to Control Unknown Dynamic Systems," *IEEE Transactions on Cybernetics*, vol. 48, no. 2, pp. 543–555, Feb. 2018.

[24] L. Ljung, *System Identification Theory for the User*, 2nd ed. Prentice Hall, Inc., 1999.

[25] H. J. Damveld, G. C. Beerens, M. M. van Paassen, and M. Mulder, "Design of Forcing Functions for the Identification of Human Control Behavior," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1064–1081, Jul. 2010.

[26] D. M. Pool, P. M. T. Zaal, M. M. van Paassen, and M. Mulder, "Identification of Multimodal Pilot Models Using Ramp Target and Multisine Disturbance Signals," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 1, pp. 86–97, 2011.

III

Appendices

1

Candidate Pool Limit Selection

For selecting the limits of the HO model parameters for the SSID candidate pools, previous parameter fits obtained by van der El et al. (2018b,a, 2020) (or identically van der El (2018), chapters 4, 5 and 6, respectively) were used. tables 1.1 and 1.2 show a summary containing the maximum and minimum values estimated and the combined limits for SI and DI CE dynamics. table 1.3 then shows the combined limits for these two types of dynamics and the consequently selected limits for the candidate pool.

There are a number of aspects that can be clarified based on these tables. First, some of the fits in the previously performed experiments seem to exceed the limits eventually set for the SSID candidate pools. This shows for example in the fit of ζ_{nms} for Ch. 6 in table 1.1 and for ω_{nms} in table 1.2. For the first case, this limit is increased to 0.1 to prevent the algorithm from selecting bad fits with a low damping ratio. For the second example, the limits are decreased as the results of the fits for Ch.4 were obtained with 1s of preview. As this research focuses on preview tasks with 2s of preview, the limit was decreased to 16, corresponding well to high preview times.

Another example is found for $K_{e^\star}$ in table 1.1, where for Ch. 6 the limit is found to be 2.80. The reason for this is that these results have been obtained from predictive simulations, rather than from experimental results. As such values are not found in the experimental results, the limit is set at 2. Lastly, some limits have been decreased as the parameters were clear outliers.

Table 1.1: Maximum- and minimum values of the different parameters as obtained from van der El (2018) for DI CE dynamics

		Ch. 4	Ch. 5	Ch. 6	Combined
$K_{e^\star}$	max	0.31	0.51	0.42	0.51
	min	0.12	0.12	0.19	0.12
$T_{L,e^\star}$	max	2.97	2.66	1.84	2.97
	min	1.20	0.76	0.90	0.76
τ_v	max	0.35	0.33	0.42	0.42
	min	0.24	0.24	0.28	0.24
ω_{nms}	max	7.68	10.13	10.40	10.40
	min	4.81	4.98	5.74	4.81
ζ_{nms}	max	0.78	0.80	0.24	0.80
	min	0.33	0.16	0.05	0.05
K_f	max	0.83	1.23	1.07	1.23
	min	0.17	0.65	0.77	0.17
τ_f	max	1.08	1.93	1.41	1.93
	min	0.81	0.62	0.37	0.37
$T_{l,f}$	max	1.92	1.49	1.40	1.92
	min	0.10	0.24	0.80	0.10

Table 1.2: Maximum- and minimum values of the different parameters as obtained from van der El (2018) for SI CE dynamics

		Ch. 4	Ch. 5	Ch. 6*	Combined
$K_{e^\star}$	max	1.39	1.62	2.80	2.80
	min	0.84	0.82	1.40	0.82
$T_{L,e^\star}$	max	-	-	-	-
	min	-	-	-	-
τ_v	max	0.30	0.39	0.20	0.48
	min	0.18	0.14	0.20	0.14
ω_{nms}	max	18.00	14.20	12.00	14.20*
	min	7.47	7.41	12.00	7.41
ζ_{nms}	max	0.70	0.55	0.20	0.70
	min	0.24	0.06	0.20	0.06
K_f	max	1.12	1.05	1.00	1.12
	min	0.85	0.94	1.00	0.85
τ_f	max	0.99	0.87	0.70	0.99
	min	0.49	0.38	0.40	0.38
$T_{l,f}$	max	0.98	0.15	0.80	0.98
	min	0.11	0.00	0.70	0.00

Table 1.3: Maximum- and minimum values of the different parameters as obtained from van der El (2018) for both SI and DI CE dynamics and the selected ranges.

		Combined SI	Combined DI	Selected
$K_{e^\star}$	max	2.80	0.51	2*
	min	0.82	0.12	0.1
$T_{L,e^\star}$	max	-	2.97	3
	min	-	0.76	0.6
τ_v	max	0.39	0.37	0.4
	min	0.14	0.24	0.1
ω_{nms}	max	14.71	10.40	16
	min	7.41	4.81	4
ζ_{nms}	max	0.70	0.80	1
	min	0.06	0.05	0.1
K_f	max	1.12	1.23	2
	min	0.85	0.17	0.1
τ_f	max	0.99	1.93	2
	min	0.38	0.37	0.3
$T_{l,f}$	max	0.98	1.49	1.6
	min	0.00	0.10	0.1

2

Candidate Pool Densities Paper I

This appendix provides an overview of the sizes of the different candidate pools used for paper 1, see table 2.1. The two upper rows indicate the parameter limits and after that, rows indicate the number of steps between the parameters and the resulting step size. Furthermore, table 2.2 shows how the sizes of the resulting candidate pools in terms of entries for both single and double integrator dynamics.

Table 2.1: Overview of different candidate pools used in Paper I

		K_e	$T_{L,e}$	τ_v	ω_{nms}	ζ_{nms}	K_f	τ_f	$T_{l,f}$
	max	2	3	0.4	16	1	2	2	1.6
	min	0.1	0.6	0.1	4	0.05	0.1	0.3	0.1
CP1	$n_{\theta_i,d_{CP}}$	6	3	3	4	2	-	3	2
	step size	0.380	1.2	0.15	4	0.95	-	0.850	1.5
CP2	$n_{\theta_i,d_{CP}}$	12	6	6	8	4	-	6	4
	step size	0.173	0.480	0.060	1.714	0.317	-	0.340	0.500
CP3	$n_{\theta_i,d_{CP}}$	18	9	9	12	6	-	9	6
	step size	0.112	0.300	0.038	1.091	0.190	-	0.213	0.300
CP4	$n_{\theta_i,d_{CP}}$	24	12	12	16	8	-	12	8
	step size	0.083	0.218	0.027	0.800	0.136	-	0.155	0.214
CP5	$n_{\theta_i,d_{CP}}$	30	15	15	20	10	-	15	10
	step size	0.066	0.171	0.021	0.632	0.106	-	0.121	0.167
CP6	$n_{\theta_i,d_{CP}}$	36	18	18	24	12	-	18	12
	step size	0.054	0.141	0.018	0.522	0.086	-	0.100	0.136
CP7	$n_{\theta_i,d_{CP}}$	42	21	21	28	14		21	14
	step size	0.046	0.120	0.015	0.444	0.073	-	0.085	0.115
CP10	$n_{\theta_i,d_{CP}}$	60	30	30	40	20		30	20
	step size	0.032	0.083	0.010	0.308	0.050	-	0.059	0.079
CP12	$n_{\theta_i,d_{CP}}$	72	36	36	48	24		36	24
	step size	0.027	0.069	0.009	0.255	0.041	-	0.049	0.065

Table 2.2: Overview of candidate pool sizes used in Paper I

CP No.	SI	DI
CP1	8.6E+02	2.9E+02
CP2	5.5E+04	9.2E+03
CP3	6.3E+05	7.0E+04
CP4	3.5E+06	2.9E+05
CP5	1.4E+07	9.0E+05
CP7	1.0E+08	4.8E+06
CP10	8.6E+08	2.9E+07
CP12	2.6E+09	7.2E+07

3

Forcing Function Comparison

This appendix compares the different multi-sines encountered in literature were used in combination with the feedback- and feedforward system identification techniques. Specifically, the multi-sines from van der El et al. (2020) and Sheffler et al. (2019) will be compared. First, a description of the forcing function setup is provided, followed by a comparison.

3.1. Forcing Function Description

3.1.1. Sheffler et al.

Sheffler et al. (2019) use a multi-sine defined according to eq. (3.1). Here $i \in \{1, \dots, 40\}$ and $j \in \{1, \dots, 30\}$. $\phi_{i,j}$ is a randomly selected phase angle, which is selected such that the peak magnitude is less than 2.6 (i.e., $\max|f_t| < 2.6$ hash marks (hm). This unit relates to the experimental setup and corresponds to 5.45 cm. The multi-sine then has 30 evenly spaced frequencies between 0 and 0.5 Hz.

$$f_{t,i}(t) = \frac{1}{3}\sum_{j=1}^{30} \cos\left(\frac{2\pi j t}{60} + \phi_{i,j}\right), t \in [0, 60] \tag{3.1}$$

3.1.2. Van der El et al.

van der El et al. (2020) use the same forcing functions as described in both papers. They are formed using eq. (3.2), where $N_f = 10$ is the number of sines used to compose the function, and A_k, ω_k and ϕ_k are the amplitude, frequency and phase, respectively. Contrary to Sheffler et al. (2019), van der El et al. (2020) attenuate the frequency with a factor 10 after the signal bandwidth $\omega_i = 1.5$ rad/s. These forcing functions are based on those used by McRuer et al. (1965).

$$f(t) = \sum_{k=1}^{N_f} A_k \sin(\omega_k + \phi_k) \tag{3.2}$$

3.2. Comparison

figs. 3.1 and 3.2 show the forcing functions used by Sheffler et al. (2019); van der El et al. (2020) in the time-domain and the frequency domain, respectively. fig. 3.1b shows how, despite scaling to obtain a comparable amplitude, the forcing function of Sheffler et al. (2019) appears to have a larger variance and is thus more difficult to control.

Looking at the frequency-domain, the most important differences are the attenuation and the spacing. Sheffler et al. (2019) use a flat power spectrum and divide the sines evenly over the selected range.

To get a better understanding of how the forcing functions in Sheffler et al. (2019) can be controlled, fig. 3.3 shows the different controlled element dynamics used in Sheffler et al. (2019); van der El et al. (2020). The dynamics labelled as 'Hoagg' are used in Sheffler et al. (2019) and shift from gain dynamics

to double integrator dynamics. However, due to the many differences in experimental settings in the two experiments, it is recommended to perform a comparing experiment to make a proper comparison.

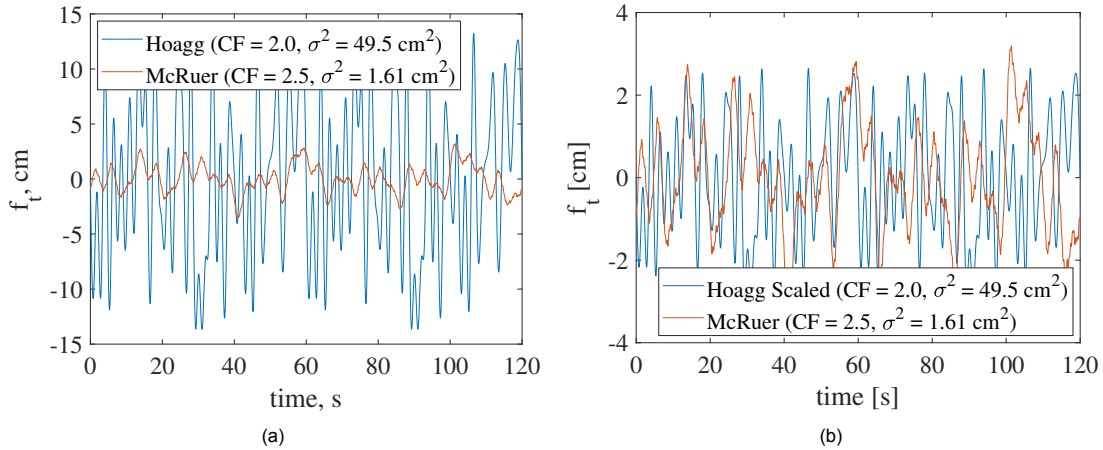


Figure 3.1: Original (a) and scaled time-domain forcing functions used by Sheffler et al. (2019) and van der El et al. (2020). In both figures, the forcing function of van der El et al. (2020) is kept constant.

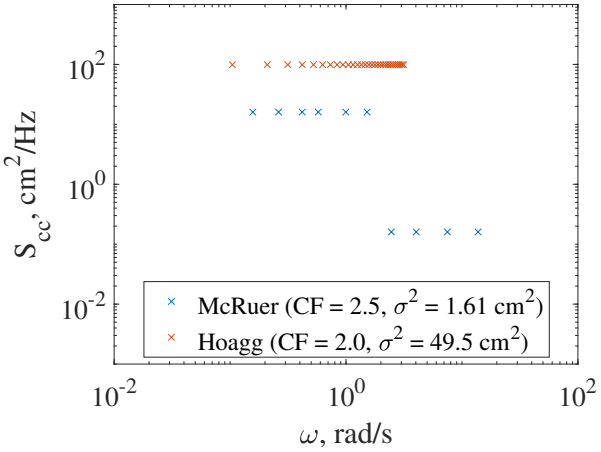


Figure 3.2: Frequency domain comparison of the forcing functions used by van der El et al. (2020) and McRuer et al. (1965).

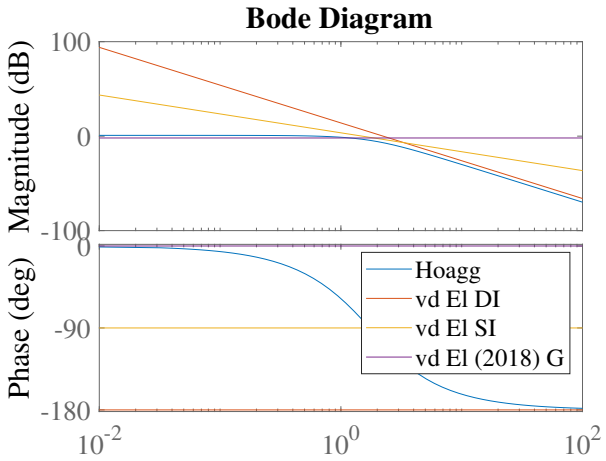


Figure 3.3: Comparison of the controlled element dynamics used by van der El et al. (2020) and McRuer et al. (1965).

4

Control Output Signal Simulation

This appendix illustrates how the simulation results from the human operator (HO) control model approximate the measured results from a human-in-the-loop experiment. This is first done to illustrate how the noise model resembles the true data and consequently, to show how the open- and closed-loop simulations of the control output signal $U(j\omega)$.

4.1. Remnant Model Validation

In order to show how the simulations match the experimental results, fig. 4.1 shows the spectra of the control output signal $U(j\omega)$ for both dynamics. For both plots, noise levels close to those found in the experiment (see chapter 7) are selected. The remnant time constant depends on the controlled element dynamics and is equal to the values described in both Paper 1 and Paper 2. table 4.1 shows the parameter estimates used for the simulations.

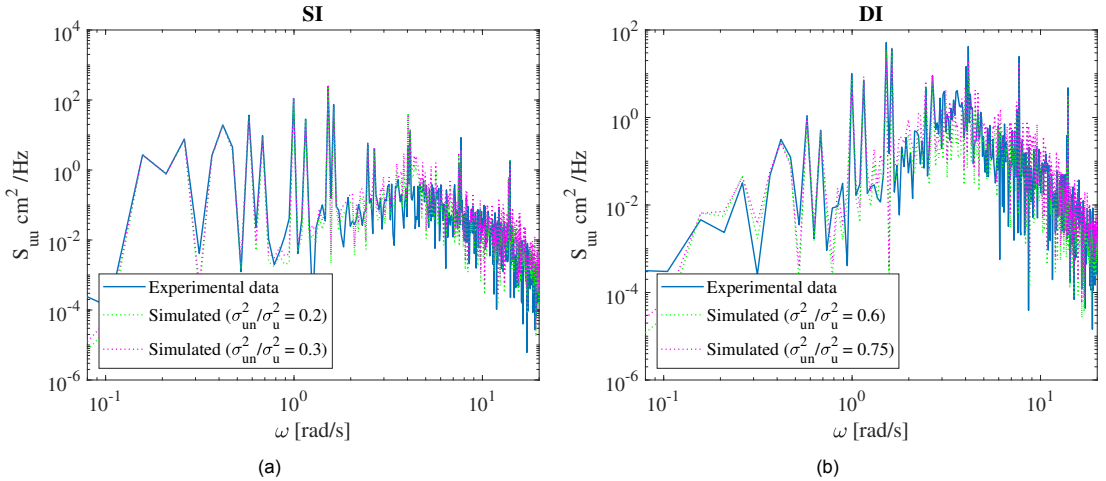


Figure 4.1: Experimentally obtained- and two simulated power spectral densities for SI (a) and DI (b). Data from subject two, obtained using the $U2_{td}$ method is used.

Table 4.1: Overview of the identified parameters for both dynamics for subject 2 using the $U2_{td}$ method.

	$K_{e^\star}$	$T_{L,e^\star}$	τ_v	ω_{nms}	ζ_{nms}	K_f	τ_f	$T_{l,f}$
SI	1.87	-	0.29	12.06	0.27	1.00	0.70	0.36
DI	0.31	1.32	0.35	7.73	0.10	1.08	1.32	1.39

4.2. Open- and Closed Loop Simulation Comparison

When simulating the control output signal $U(j\omega)$, two ways of simulating are used: the closed- and the open loop simulation. eq. (4.1) shows the closed-loop equation, where U is a function of $F_t(j\omega)$ and $N(j\omega)$ (assuming f_d is omitted for simplicity), and eq. (4.2) shows the open-loop equation, which depends on $F_t(j\omega)$, $X(j\omega)$ and $N(j\omega)$. In both equations, the dependence on $j\omega$ is omitted for brevity. Substituting $X_{CL}(j\omega)$ (eq. (4.3)) in eq. (4.2), it can be shown that $U_{CL}(j\omega) = U_{OL}(j\omega)$.

$$U_{CL}(j\omega) = \frac{H_{o_t}}{1 + H_{o_x}H_{ce}}F_t + \frac{-H_{o_x}}{1 + H_{o_x}H_{ce}} \tag{4.1}$$

$$U_{OL}(j\omega) = H_{o_t}F_t - H_{o_x}(X + N) \tag{4.2}$$

$$X_{CL}(j\omega) = \frac{H_{o_t}H_{ce}}{1 + H_{o_x}H_{ce}}F_t + \frac{-H_{o_x}H_{ce}}{1 + H_{o_x}H_{ce}} \tag{4.3}$$

However, when estimating the model parameters using the $U2_{td}$ method or when calculating the variance-accounted-for (VAF) for the experimental data, the remnant data is not available. Then, $U_{OL}(j\omega)$ is computed using eq. (4.4). Due to the omission of $N(j\omega)$ in this equation, a difference in the spectra is observed, which can be calculated in eq. (4.5). fig. 4.2 shows the different signals and indicates how indeed the term $H_{o_x}N$ is the difference between the open-loop simulation and the closed-loop simulation (or the measured data).

$$\begin{aligned} U_{OL,sim} &= H_{o_t}F_t - H_{o_x}X \\ &= \frac{H_{o_t}}{1 + H_{o_x}H_{ce}}F_t + \frac{H_{o_x}^2H_{ce}}{1 + H_{o_x}H_{ce}}N \end{aligned} \tag{4.4}$$

$$\begin{aligned} U_{OL} - U_{OL,sim} &= \left(\frac{-H_{o_x}H_{ce}}{1 + H_{o_x}H_{ce}} - \frac{H_{o_x}^2H_{ce}}{1 + H_{o_x}H_{ce}}\right)N \\ &= -H_{o_x}N \end{aligned} \tag{4.5}$$

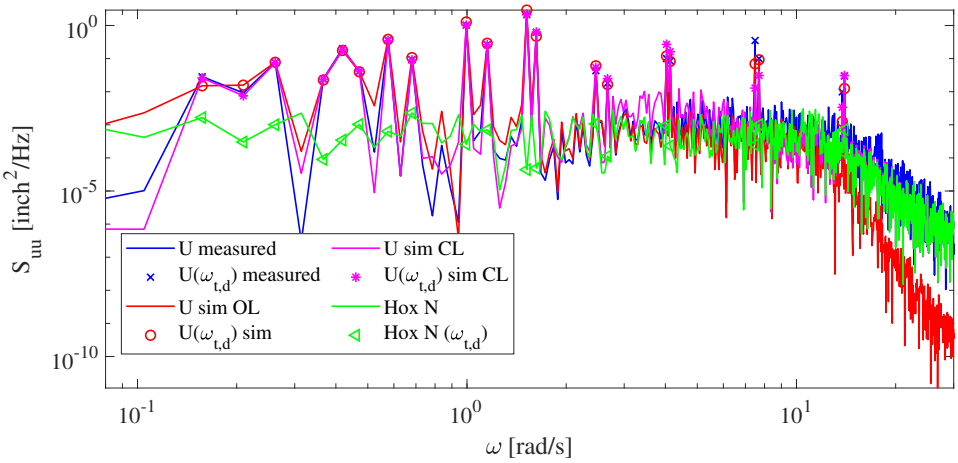


Figure 4.2: Example of the differences between the measured control output and the open- and closed-loop simulations. Data from the SI task is selected for subject 3, with the parameter fits from the $U2_{td}$ method.

5

Human Operator Model Parameter Sensitivity

This appendix discusses the sensitivity of the cost functions treated in this research to the human operator (HO) model parameters. Clear insights of the effects of different parameters on the cost function can explain why specific methods perform better than other for system identification. In this appendix, this is done by quantifying the impact of the parameter in a single number. This is done for a number of sensitivity factors and for both types of controlled element (CE) dynamics: single- and double integrator (SI and DI, respective) dynamics. First, the method is explained, followed by a discussion of the results

5.1. Method

For the comparison, five different cost functions are evaluated, presented by eqs. (5.1) to (5.5). These cost functions correspond to the cost functions evaluated in methods 2 until 6 in Paper 2. In the equations, the subscripts CL, OL and U refer to the cost function data used, being either the closed-loop (CL) describing function (DF) H_{cl}, the open-loop (OL) describing function (H_{o_t} and H_{o_x}) and the control output signal $U(j\omega)$. The subscripts t and d indicate the forcing functions used for this cost function. Then, Θ and Θ_0 represent the parameter vector and the reference parameter vector, respectively. The values of the reference parameter vector are identical to those used in Paper 1 and Paper 2, see also table 5.1.

$$J_{CL,t} = \sum_{k=1}^{N_f} \left(H_{cl_t}(j\omega_k|\Theta) - H_{cl_t}(j\omega_k|\Theta_0)\right)^2 \tag{5.1}$$

$$J_{CL,td} = \sum_{k=1}^{N_f} \left(H_{cl_t}(j\omega_k|\Theta) - H_{cl_t}(j\omega_k|\Theta_0)\right)^2 + \left(H_{cl_d}(j\omega_k|\Theta) - H_{cl_d}(j\omega_k|\Theta_0)\right)^2 \tag{5.2}$$

$$J_{OL} = \sum_{k=1}^{2N_f} \left(H_{o_t}(j\omega_k|\Theta) - H_{o_t}(j\omega_k|\Theta_0)\right)^2 + \left(H_{o_x}(j\omega_k|\Theta) - H_{o_x}(j\omega_k|\Theta_0)\right)^2 \tag{5.3}$$

$$J_{U,t} = \sum_{k=1}^{N_f} \left(U(j\omega_k|\Theta) - U(j\omega_k|\Theta_0)\right)^2 \tag{5.4}$$

$$J_{U,td} = \sum_{k=1}^{2N_f} \left(U(j\omega_k|\Theta) - U(j\omega_k|\Theta_0)\right)^2 \tag{5.5}$$

Table 5.1: Overview of the model parameters used in the simulations.

CE	H_{ce} [−]	$K_{e^\star}$ [-]	$T_{L,e^\star}$ [s]	τ_v [s]	ω_{nms} [rad/s]	ζ_{nms} [-]	K_f [-]	τ_f [s]	$T_{l,f}$ [s]
SI	$\frac{1.5}{(j\omega)}$	1.25	–	0.23	13.0	0.13	1.0	0.60	0.15
DI	$\frac{5}{(j\omega)^2}$	0.33	1.27	0.33	7.4	0.13	0.95	0.95	1.15

To quantify the parameter sensitivity of the cost function, one of the parameters $\theta_i \in \Theta$ is multiplied by both a factor f_θ and its inverse $\frac{1}{f_\theta}$. For both new parameter sets, the new cost function values are determined, which are summed to form the sensitivity s_{x,θ_i}, see eq. (5.6). Here, the subscript x represents any type of cost function as sketched earlier.

$$s_{x,\theta_i} = J_x(\Theta(\theta_i \cdot f_\theta) + J_x(\Theta(\theta_i \cdot \frac{1}{f_\theta}) \tag{5.6}$$

In order to compare the different cost functions, the value of s_{x,θ_i} is normalized by the maximum value of s_{x,θ_i} for all $\theta_i \in \Theta$, see eq. (5.7). This results in

$$s_{norm_{x,\theta_i}} = \frac{s(\theta_i)}{\max_{i\in\Theta} s(\theta_i)} \cdot 100\% \tag{5.7}$$

For f_θ, a number of values are used and table 5.2 shows how the parameters vary for $f_\theta = 2$ and DI parameters. In the evaluation, values for f_θ of 1.1, 1.33, 2 and 4 are used.

Table 5.2: Examples of the different values for $\theta_i \cdot f_\theta$ and $\theta_i \cdot \frac{1}{f_\theta}$ for the different parameters for a value of $f_\theta = 2$ and DI CE dynamics.

	$K_{e^\star}$	$T_{L,e^\star}$	τ_v	ω_{nms}	ζ_{nms}	K_f	τ_f	$T_{l,f}$
Θ_0	0.33	1.27	0.33	7.4	0.13	0.95	1.05	1.15
$\theta_i \cdot f_\theta$	0.17	0.64	0.17	3.70	0.07	0.48	0.53	0.58
$\theta_i \cdot \frac{1}{f_\theta}$	0.66	2.54	0.66	14.80	0.26	1.90	2.10	2.30

5.2. Results and Discussion

fig. 5.1 shows the results for the different values of $s_{norm_{x,\theta_i}}$ for all dynamics and all factors of f_θ. Although variations are visible depending on both the type of dynamics and the the value of f_θ a general trend is visible. The parameters of the neuromuscular system and the feedforward lag-time constant $T_{l,f}$ generally have a smaller effect on the cost function and can thus be considered to be less identifiable for all cost functions. However, as the results do vary quite a lot based on the different conditions, a clear optimal cost function cannot be selected based on this metric.

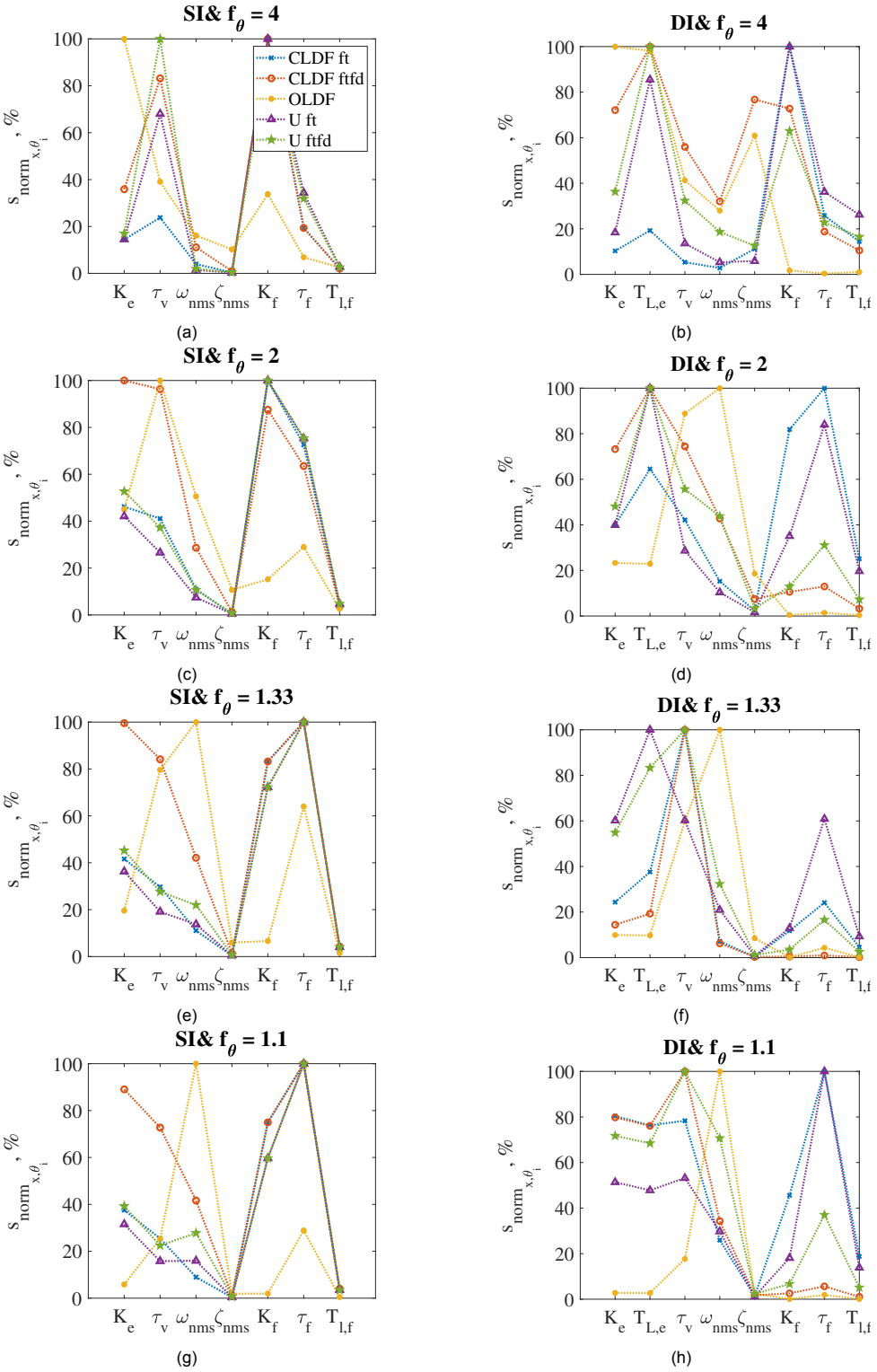


Figure 5.1: Sensitivity value $s_{norm_{x,\theta_i}}$ for both controlled element dynamics and all values of f_θ.

6

Experiment Plan

This chapter contains a brief explanation of the proposed tests to be performed on Wednesday 23/09/2020 and Thursday 24/09/2020 in the HMI-lab. The goal is to generate data for tracking tasks with either one or two forcing functions, to evaluate the performance of several identification techniques. This will be done for preview tracking tasks with SI and DI CE dynamics.

6.1. Experiment Setup

This section describes the setup of the experiment. First, a list of the independent-, control- and dependent variables is provided. Then, the experiment matrix is presented.

6.1.1. Experiment variables

- Independent Variables
  - Forcing functions: f_t only and both f_t and f_d
  - Controlled element dynamics: Single vs. double integrator dynamics
- Control Variables
  - Appraratus: right side of the HMI-Lab at the Faculty of Aerospace Engineering at TU Delft. Side stick settings are identical to those in van der El et al. (2020).
  - Preview display with a preview time of $\tau_p = 2$ seconds.
  - Forcing function sets
    - MR64 from van der El et al. (2020)
    - $\sigma_t^2 = 1.61cm^2$ and $\sigma_d^2 = 0.26cm^2$
  - Instructions and communication
- Dependent Variables
  - Error- and control output variance (see chapter 7)
  - Crossover frequencies and phase margins (see chapter 7)
  - Describing function variance
  - Parameter estimates
  - Variance-accounted-for

6.1.2. Experiment matrix

The experiment matrix is shown in table 6.1. Subjects for the experiment were all staff or students from TU Delft, experienced in performing tracking tasks.

Table 6.1: Balanced Latin Square of the experimental conditions.

Subject	Condition 1	Condition 2	Condition 3	Condition 4
1	2, DI	1,SI	1, DI	2, SI
2	1,SI	2, DI	2, SI	1, DI
3	2, SI	1, DI	1, SI	2, DI
4	1, DI	2, SI	2, DI	1, SI

6.2. Procedures

Before the experiment is performed, participants will perform one run with each condition to familiarize themselves with the dynamics. After that, the official measurements start. Per condition, a participant will then perform at least 6 runs until 5 consecutive measurements with a stable error-variance are obtained.

During the runs, participants will not be disturbed. After the runs, the score will be communicated, after which the participant is asked whether he is ready for the next run. After performing two conditions, a 10-minute break will be taken to let the participant rest. Finally, there will be a 5-minute wrap-up. Combined, an entire session is then expected to last around 2 hours.

7

Experiment Results

This appendix presents and discusses the experiment results involving the signal variances and the crossover frequency and phase margin. All results are presented for the four different test conditions, in which the number of forcing functions (f_t only or f_t and f_d) and the type of controlled element dynamics (single- and double integrator, SI and DI, respectively) are varied.

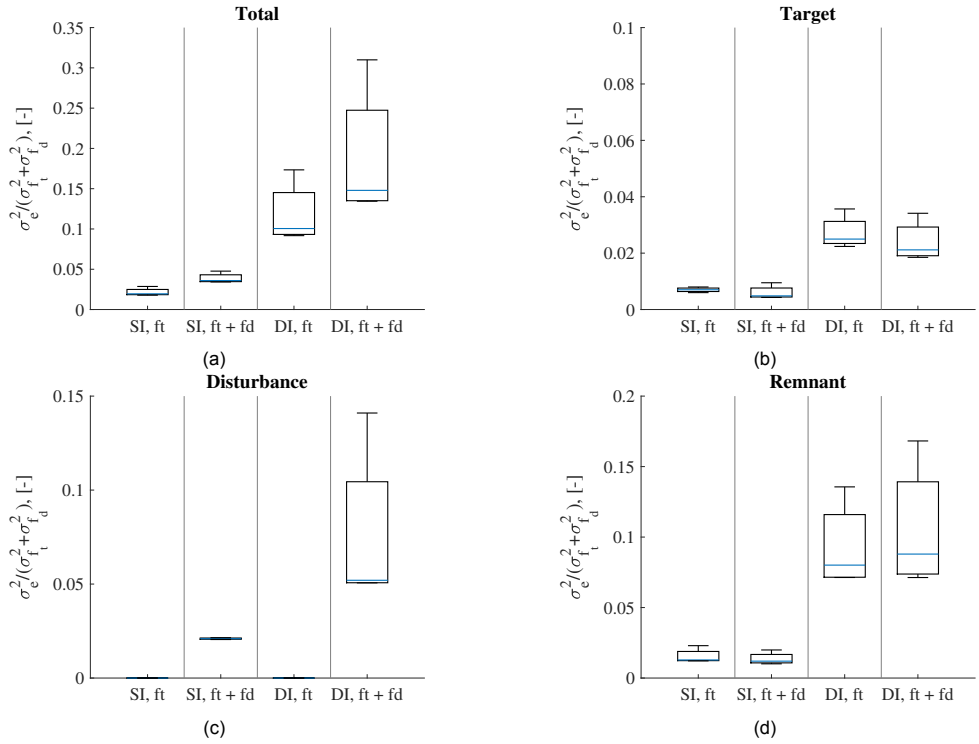


Figure 7.1: Total tracking performance and the separate contributions due to the target- and disturbance functions and the remnant.

7.1. Signal Variances

The signal variances will first be discussed. Here, the error variance — normalized by the combined variance of f_t and f_d —, the control output-variance and the remnant-to-control-output ratio will be presented. For the first two, both the total variance and the contribution due to the target, disturbance and the remnant are shown. This is done by evaluating the power spectra at the frequencies at which the respective signals act, e.g., the target function variance by integrating the power spectral density at the

target frequencies only. For the remnant contribution, the contribution of the remnant at these frequencies is calculated by interpolating using the neighbouring frequencies van der El et al. (2019). This is taken into account when estimating both the remnant variance, as well as the target- and disturbance function.

fig. 7.1 shows the normalized error variance. As expected, the error variance increases both when adding a second forcing function and when changing the dynamics from single- to double integrator. Both increases are not surprising. It is also noted that the addition of the disturbance function does not appear to affect the variance of the target- and remnant contributions to the error.

Next, fig. 7.2 shows the control output variance, which is divided in the same way as the error variance. The figure shows how the human operator (HO) is less active for DI dynamics, which corresponds to earlier observations by van der El et al. (2018b). Furthermore, the HO shows more control activity when a second forcing function is added. Then, just as for the error variance, the addition of the disturbance function does not have a large effect on the target function and the remnant contributions.

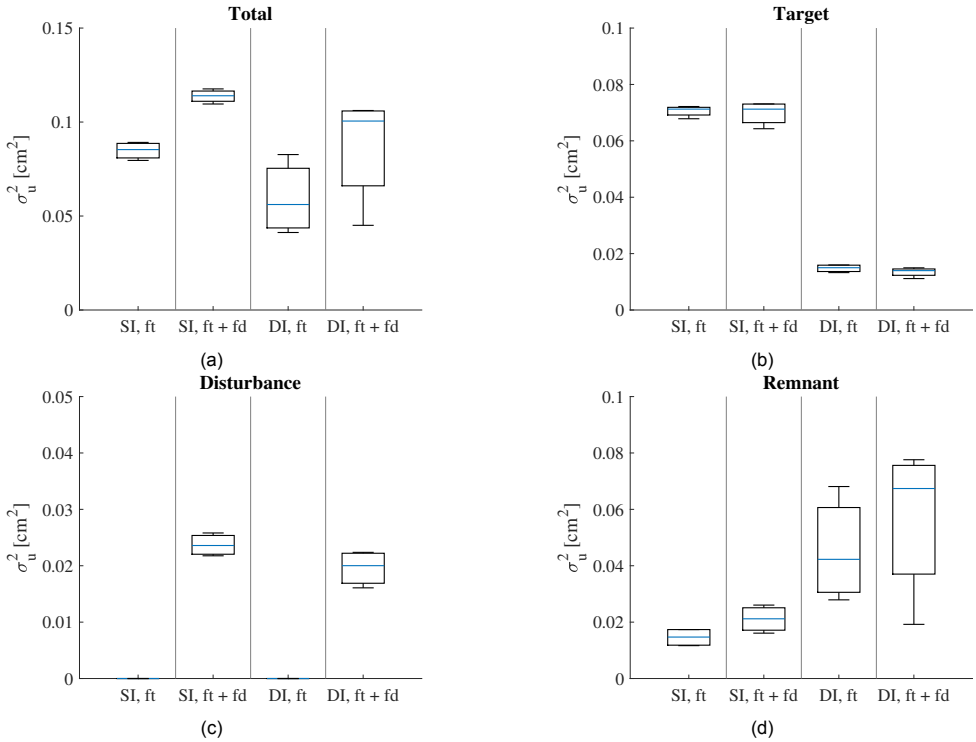


Figure 7.2: Total control output variance and the separate contributions due to the target- and disturbance functions and the remnant.

Finally, when looking at the remnant-to-control-output ratio $\frac{\sigma_{u_n}^2}{\sigma_u^2}$, fig. 7.3 shows how this ratio is affected by the controlled element dynamics and the number of forcing functions. In accordance with the results obtained by van der El et al. (2019), the noise ratio is larger for DI dynamics compared to SI dynamics. Compared to their results, the levels are slightly smaller (0.35 against around 0.2 in these results) for SI and slightly larger for DI (0.6 against around 0.65 in this paper). Furthermore, the figure shows how the addition of a second forcing function does not have a large effect for SI dynamics, while for DI, the 'level of non-linearity' appears slightly higher for just a single forcing function. As the sample size of this experiment was relatively small (there were four subjects), a larger experiment is recommended to address this more specifically.

7.2. Crossover Frequency and Phase Margin

Next to the signal variances, the crossover frequencies and phase margins are determined. fig. 7.4 shows the crossover frequencies ω_c and the phase margins ϕ_m for all conditions. The figure shows both

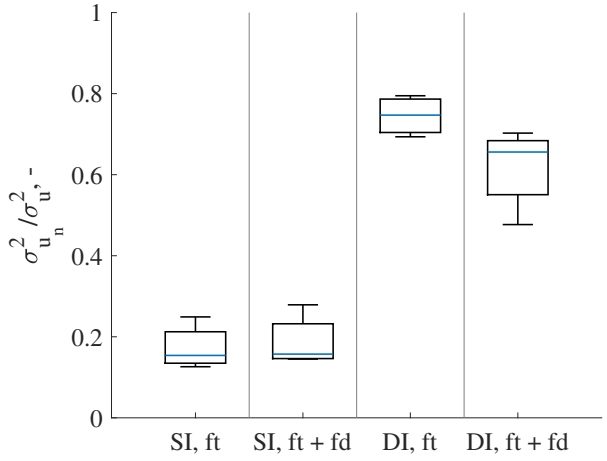


Figure 7.3: Remnant-to-control-output ratios $\frac{\sigma^2_{u_n}}{\sigma^2_u}$ for the different conditions.

metrics obtained from the raw data and from the parameter fits from the $U2_n$ method. Furthermore, for comparison, values obtained by van der El et al. (2020) — who performed an experiment with identical settings — and theoretical estimates using the verbal adjustment rules developed by McRuer and Jex (1967) for compensatory tracking tasks are shown. These estimates are calculated using eqs. (7.1) to (7.3), where ω_{c_0}, τ_0 and $\Delta\tau$ are empirically determined constants found by McRuer and Jex (1967).

$$\omega_c = \omega_{c_0}(H_c) + 0.18\omega_i \tag{7.1}$$

$$\tau_e = \tau_0(H_c) - \Delta\tau(H_c)\,\omega_i \tag{7.2}$$

$$\phi_m = \frac{\pi}{2} - \tau_e\omega_c \tag{7.3}$$

The figure shows a number of things. First, the results obtained in this experiment correspond well to those found by van der El et al. (2020). Furthermore, with SI dynamics, higher crossover frequencies and higher phase margins are found, as expected.

When comparing the results with the theoretical estimates, it is seen that for the target function, the crossover frequency corresponds fairly well, while the phase margin is about 5 times larger than estimated. This is because of the additional preview, that is available in this tracking task. After all, the verbal adjustment rules are developed for compensatory tracking tasks. When looking at the disturbance, on the other hand, the verbal adjustment rules should apply fairly well, as the disturbance task only is a compensatory tracking task. Here, the crossover frequency is lower than expected, while the phase margin is higher. These differences are considered to be slight offsets from the model that can be compensated for by selecting slightly different values for the crossover frequency. From the theory of the crossover model it is known that a higher crossover frequency corresponds to a lower phase margin. It thus appears as if the balance of this experiment was shifted compared to the expected levels.

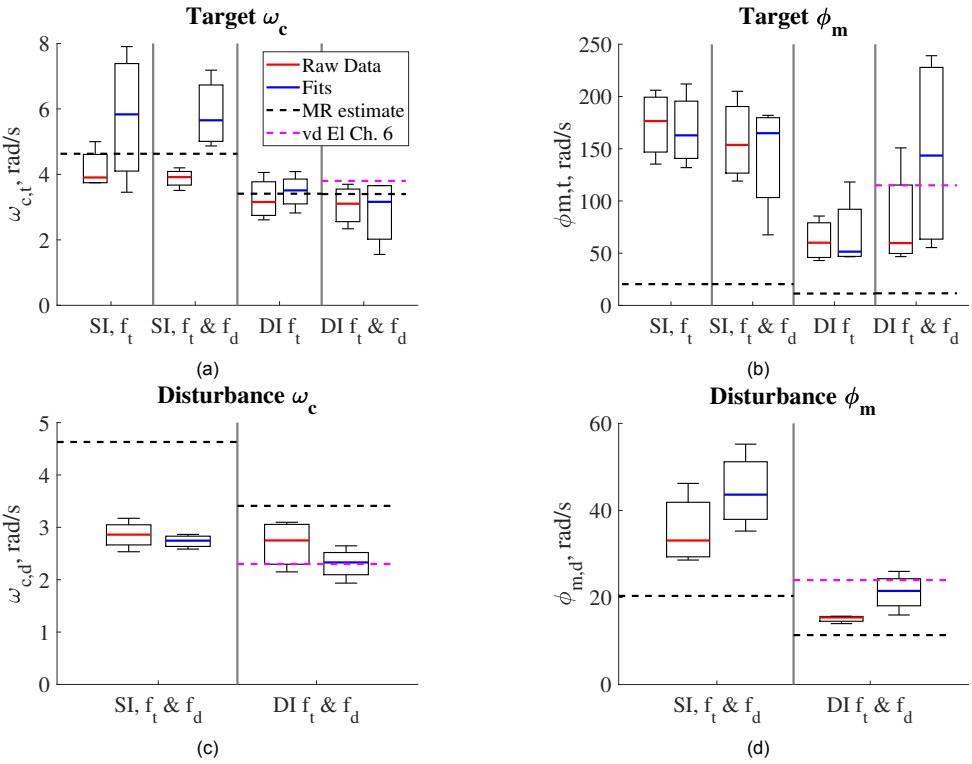


Figure 7.4: Crossover frequencies and phase margins for the target- and disturbance functions for the different experiment conditions. Results from the raw experimental data, as well as fits from the $U2_n$ method are shown. Furthermore, reference values obtained by van der El et al. (2020) and theoretical estimates from the theory of McRuer and Jex (1967) are shown.

8

Comparison U-Methods

This appendix shows the results obtained when comparing the different methods that fit on the control output $U(j\omega)$. Besides the $U1_t$ and the $U2_{td}$ methods (discussed in Paper 2), the $U1_n$ and $U2_n$ methods are added. These methods fit on the remnant frequencies, i.e., all frequencies. table 8.1 gives the overview of the different methods. The parameter limits used for these methods are the same as described in Paper 2. The results discussed in this appendix involve the mean absolute error (MAE), the parameter fits (both for simulations and experiment), the variance-accounted-for (VAF) and the open-loop frequency domain error ϵ_{OL}.

Table 8.1: Overview of the methods evaluated in this research.

Method #	Abbreviation	FoFus	CF data	Fitting freqs	Optimization	Reference
1	U1t	f_t	$U(j\omega)$	ω_t	NM simplex	-
2	U1n	f_t	$U(j\omega)$	ω_n	NM simplex	-
3	U2td	f_t, f_d	$U(j\omega)$	ω_t, ω_d	NM simplex	van der El et al. (2018a, 2020)
4	U2n	f_t, f_d	$U(j\omega)$	ω_n	NM simplex	van der El et al. (2018b)

8.1. Mean Absolute Error

fig. 8.1 shows the MAE for both dynamics. The figure shows how the best fits are obtained when only fitting on the target- and disturbance frequencies. The methods that fit on all frequencies have an MAE of approximately twice the size of the $U1_t$ and $U2_{td}$ methods, for all conditions.

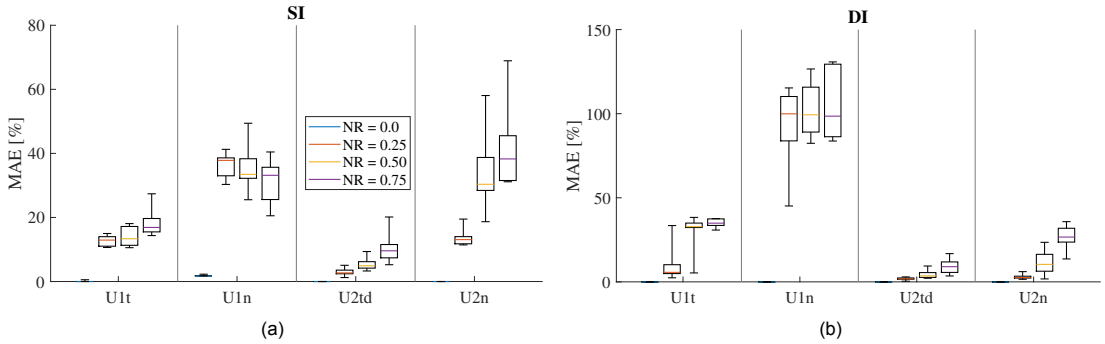


Figure 8.1: Mean absolute error (MAE) of the different methods for the evaluated noise ratios. Results are shown for single- (a) and double integrator (b) dynamics.

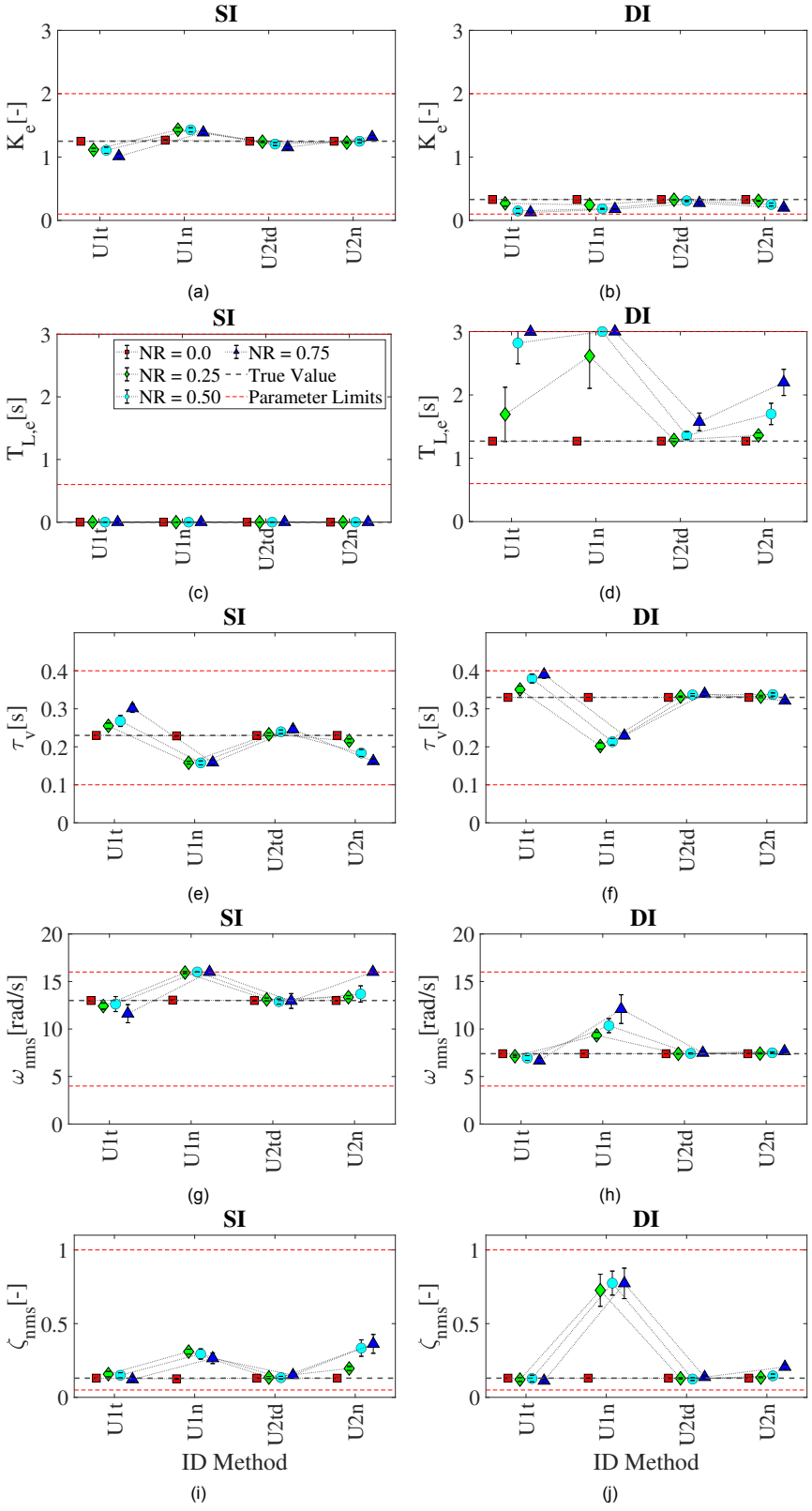


Figure 8.2: Parameter estimates of the feedback subsystem $H_{o_{cmp}}$ for SI (a, c, e, g & i) and DI (b, d, f, h & j) CE dynamics. Errorbars indicate the 95% confindence interval.

8.2. Parameter Fits

figs. 8.2 and 8.3 show the parameter estimates of the simulations for the different noise levels for both dynamics for both $H_{o_{cmp}}$ and H_{o_p}. For $H_{o_{cmp}}$, the $U1_n$ and $U2_n$ methods cannot properly estimate $T_{L,e}$,

τ_v, ω_{nms} and ζ_{nms}, although this varies depending on the controlled element dynamics. For H_{o_p} the estimates are more accurate, except for $T_{l,f}$, where the methods that fit on all frequencies show errors of over 0.2 seconds again.

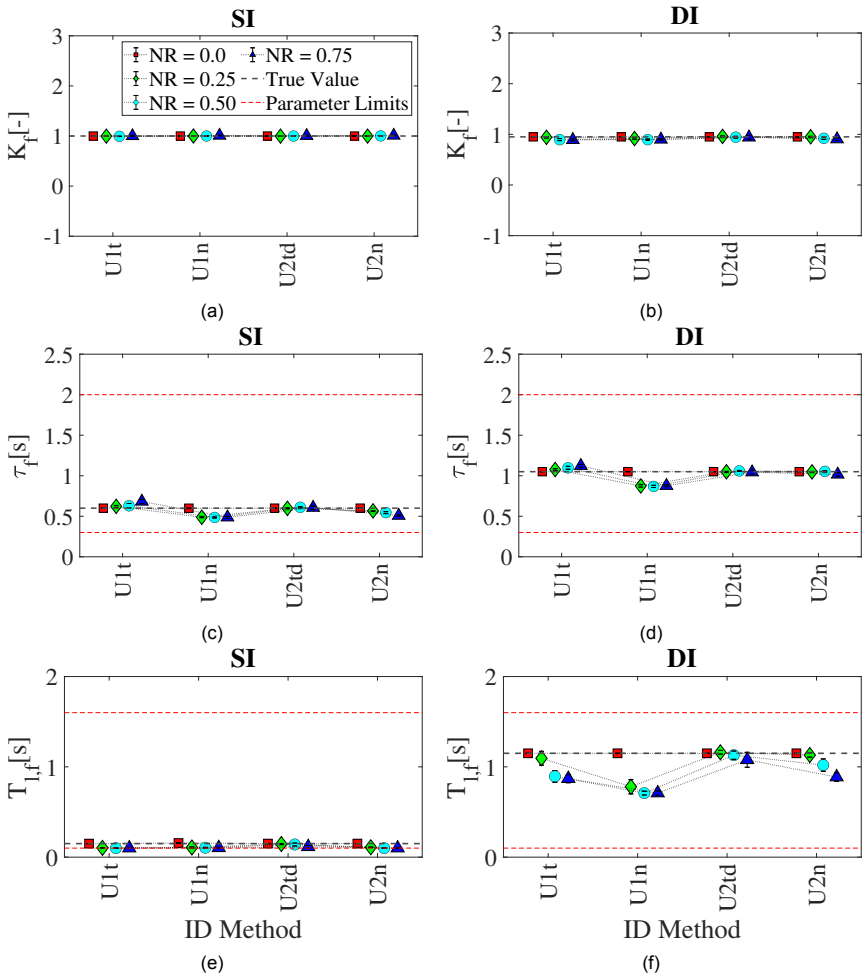


Figure 8.3: Parameter estimates of the feedforward subsystem H_{o_p} for SI (a, c & e) and DI (b, d & f) CE dynamics. Errorbars indicate the 95% confindence interval.

fig. 8.4 shows the parameter estimates from the experiment data. Just as in paper 2, different human control behavior is observed when using one or two forcing functions for single integrator dynamics. This is apparent in the feedback gain $K_{e^\star}$. Besides this, the $U2_{td}$ and $U2_n$ methods show similar results. For DI dynamics, the $U1_n$ method often shows a large mismatch compared to the other three methods.

8.3. Variance-Accounted-For

fig. 8.5 shows the VAFs obtained from the experiment results. Except for the results of $U1_t$ for DI dynamics, the VAFs are all fairly similar. An explanation for this is the fact that the VAF is calculated on the control output, corresponding to the cost function data of all methods.

8.4. Frequency Domain Error

fig. 8.6 shows the frequency domain error obtained from the different fits. These results correspond to the trends observed in the parameter fits, where the methods that fit on all frequencies have lower performance.

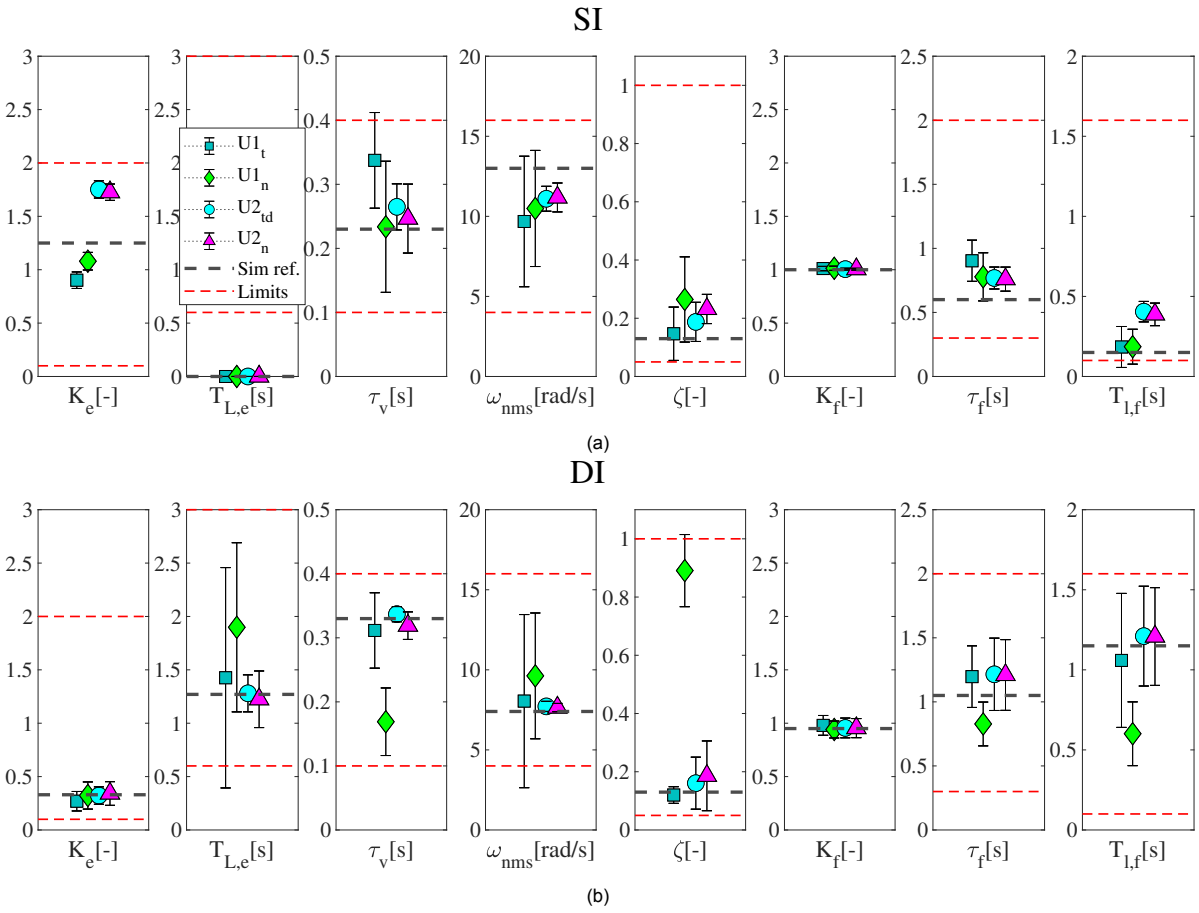


Figure 8.4: Parameter estimates of the experiment obtained with the evaluated methods for both SI- (a) and DI (b) dynamics.

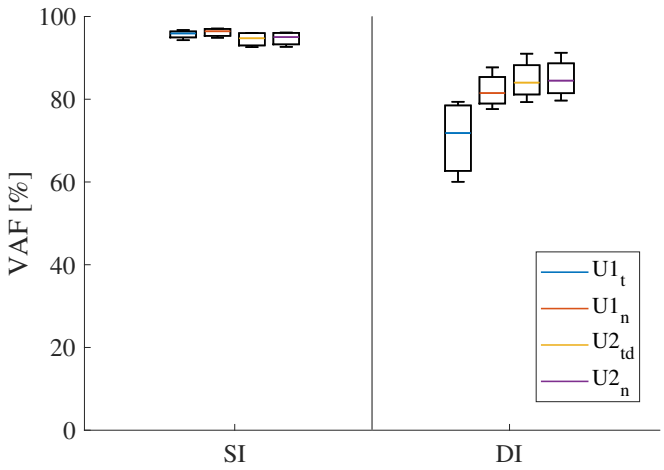


Figure 8.5: Box plots showing the VAFs of the experiment results for both types of dynamics and all methods.

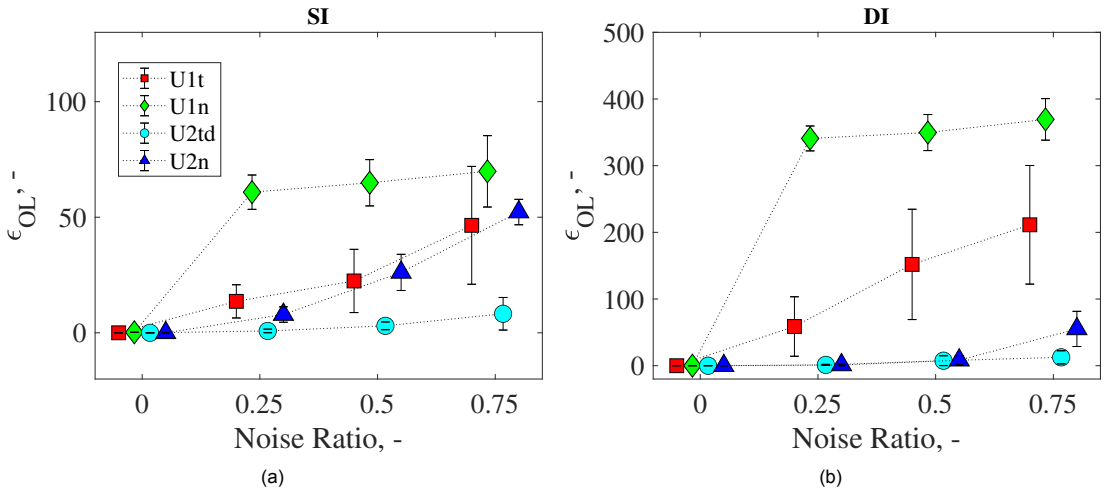


Figure 8.6: Frequency domain error of the estimated and the true subsystems H_{o_t} and H_{o_x}. Results are shown for single- (a) and double integrator (b) dynamics, with errorbars indicating the mean and 95% confidence intervals.

IV

Preliminary Report

1 Introduction

Manual control cybernetics is the research field that tries to understand the interaction between humans and machines using a system-theoretical and model-based approach (Wiener, 1965; Mulder et al., 2018). An example of its value has been illustrated by the crossover model from McRuer and Jex (1967), which is able to describe human control behavior in compensatory (feedback only) tracking tasks. The knowledge obtained from this model has often been used to tune manual control devices and to understand among others human bio-dynamics.
Although this compensatory control behaviour is well understood, the application area is relatively limited. Human control tasks generally require adaptive- and/or anticipatory feedforward control behaviour. This last aspect is seen in for example driving along roads or when tracking a predictable signal. Despite recent advances in this research field, this is still not understood well (Mulder et al., 2018, 2019). Being able to understand and model the strategies that humans apply when they are enabled to use feedforward control, will be a stepping stone in the development of better support systems and human-like automation.
To identify and model human control behavior which includes both feedback- and feedforward components, tools such as system identification techniques are needed. Besides the existing techniques (van Paassen and Mulder, 1998; Nieuwenhuizen et al., 2008; Zaal et al., 2009), recently a new subsystem identification (SSID) technique that can identify feedback- and feedforward behavior was presented (Zhang and Hoagg, 2016a,b; Zhang et al., 2018; Sheffler et al., 2019). To have a clear overview of the advantages and disadvantages of this new technique with respect to the current state-of-the-art, it is valuable to thoroughly analyse the workings of this technique. With this in mind, the objective for this research can be formed:

Contribute to the set of tools used to identify feedback- and feedforward behaviour in manual control cybernetics by exploring the capabilities and limitations of subsystem identification.

Based on this research objective, the aim of this report is to provide a background on the problem and present a research proposal to achieve this objective. The background of the problem here considers the modelling of the human controller, the type of tracking tasks that allow for feedback- and feedforward behavior and the other methods that are able to identify both these types of control behavior. Regarding the research proposal itself, it will be important to determine how- and when SSID works well, and under which circumstances it does not work. This can then be compared to the currently available identification techniques.

The report is structured as follows. First, a background on the modelling and identification of the human operator in feedback- and feedforward tracking tasks is provided in Chapter 2. Then, the SSID technique is explained and examples are evaluated in Chapters 3 and 4, respectively. After that, the plans for this research are presented in Chapter 5. A test for the application of SSID to a different model structure is then described in Chapter 6, followed by descriptions of the simulation setup in Chapter 7. Finally, the next steps of the research are presented in Chapter 8 and the conclusions are described in Chapter 9.

2

Modelling and identification of feedback and feedforward in manual control

To analyse the capabilities and limitations of a method such as subsystem identification, it is important to have a good understanding of its context. As mentioned in the introduction, subsystem identification (SSID) is a method that can identify feedback- and feedforward subsystems in manual control tracking tasks. This chapter will therefore provide background on just that. First, an overview of the different organisations for modelling human controllers is given. Then, model structures and research performed for two types of control tasks using feedback- and feedforward are discussed. Finally, methods used to identify feedforward components in manual control are discussed.

2.1. Modelling the human controller

The modelling of human controllers (or human operators) in manual control cybernetics is applied to skill-based behaviour (Rasmussen, 1983). However, due to human's adaptive nature, different situations might evoke different control strategies. In the attempt of modelling humans as controllers, different model structures might therefore be suitable depending on the control task. Possible variations in the control task could be caused by for example the number- or type of inputs provided to the human, or the way the system responds to the humans inputs. This section addresses different model structures that might be selected, and tools that might be used to select an appropriate structure. First, the Successive Organisation of Perception (SOP), first developed by Krendel and McRuer (1960) and later adapted by McRuer et al. (1968), discusses a more general framework of different organisations of human control behaviour. Then, deeper focus is put on the two- and three-channel model structures that allow for both feedback- and feedforward components in the models.

2.1.1. Successive Organisation of Perception

The SOP is setup to provide a framework for selecting a suitable model structure for a particular situation. The theory describes the different modes of human control behaviour in three stages: compensatory, pursuit, and precognitive behaviour. Figure 2.1 shows schematic representations of these three stages.

First, the compensatory stage (Figure 2.1a) describes the situation in which the only input given to the operator is the system error (e), i.e. the input (f_t) minus the output (x), see Figure 2.2a for the display. The pilot acts as a quasi-linear system, generating a linear output to the error with H_{o_e}, which is combined with the non-linear remnant term (n) to generate the human control output (u). This is the most constrained tracking task of the three and until recently, the only one for which well-fitting models had been found. With these models, such as the crossover model (McRuer and Jex, 1967), predictions about the human operator (HO) dynamics can be made for different task variables. However, assuming the target signal is unpredictable, the HO response for this stage is purely reactive and no feedforward component is present. The relevance of the compensatory stage for this research is therefore limited.

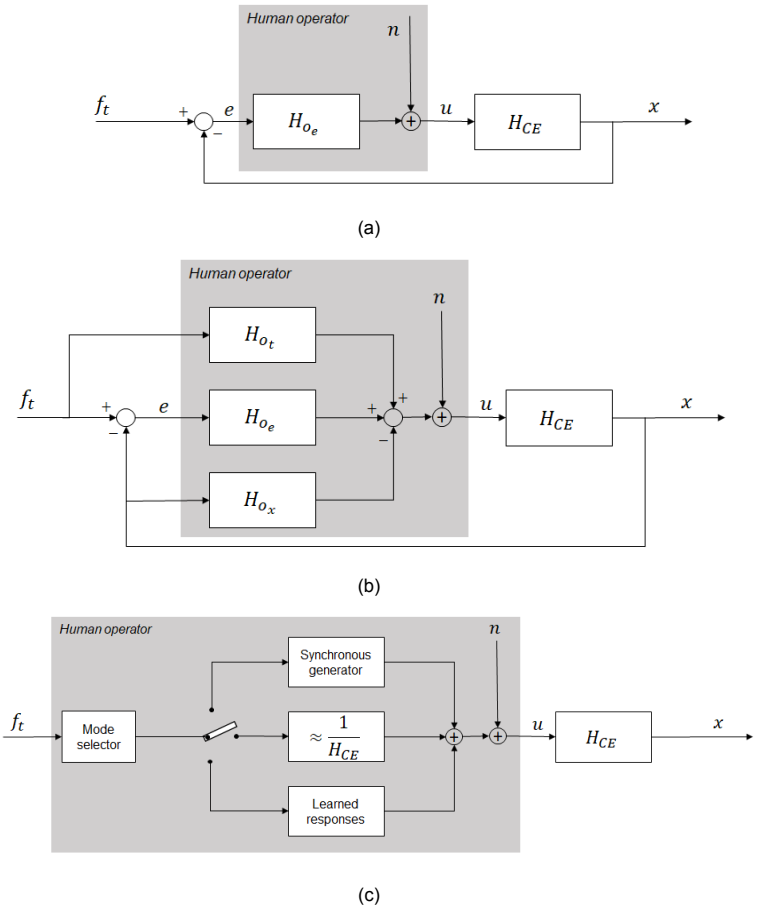
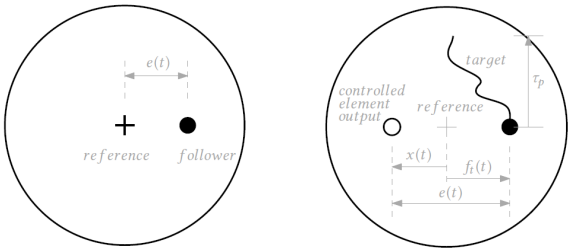


Figure 2.1: Schematic representations of the three stages of skill-based manual control behaviour, according to the Successive Organisation of Perception (McRuer et al., 1968): compensatory- (a), pursuit- (b) and precognitive (c) behaviour. The figures are reproduced largely based on the adaptions according to Mulder et al. (2018).

The next stage of the SOP, pursuit behaviour, is more relevant for this research. As shown by Figure 2.1b, in this phase the human operator can see the target signal and the controlled element output, and can then derive the error, see also Figure 2.2b. The HO can thus respond to all three signals. This allows the HO to generate both feedback- and feedforward control responses. Although pursuit tracking tasks are more realistic compared to compensatory tracking tasks and experiments have shown that pursuit tracking tasks provide better tracking results, pursuit has received fewer attention than compensatory(Mulder et al., 2018). This is because the additional signals increase the complexity of the modelling process. Furthermore, additional non-linearities in the human control behaviour arise due to the probabilistic characteristics of the target signal (Mulder et al., 2019). Some of these problems are discussed in Section 2.1.2. An extension of the pursuit display is the preview display, in which a part of the future of the target signal is visible, see Figure 2.2b (Hess, 1981).

The last stage of the SOP is the stage of precognitive control behaviour. In this stage, the HO has full knowledge of the tracking signal and its future. Therefore, the HO does not actively control the displayed error, but applies open loop feedforward control behaviour (Krendel and McRuer, 1960; McRuer et al., 1968), see Figure 2.1c. Feedback control behaviour is then not applicable. Such situations could be achieved in tracking tasks with predictable target signals, although this would probably only hold for periods within the tracking task, as pure pre-cognitive behaviour is an idealization (McRuer et al., 1968). After all, the human knowledge is never perfect and human limitations introduce noise in the control loop.



(a) (b)

Figure 2.2: Schematic representations of compensatory (a) and pursuit/preview displays (b). A pursuit display is basically a preview display with 0 preview ($\tau_p = 0$). Figures taken from van der El (2018).

Summarizing the three stages of the SOP, only the last two allow the HO to utilize feedforward control. However a side-note regarding this theory should be made. When setting up the SOP, the theory was not fully supported. The main reason for this was the lack of identified human feedforward responses. Over the years though, empirical results have shown to support the SOP. As summarized by Mulder et al. (2018); Drop (2016), improved tracking results using pursuit displays or predictable signals (Pew et al., 1967), decreased time delays and phase lags for predictable signals (Yamashita, 1989), decent tracking of a predictable signal after occlusion of the display (Hess, 1965) and different describing functions using pursuit displays (Wasicko et al., 1966) provide indirect evidence of a precognitive control strategy. The evidence is, however, not conclusive and alternative explanations not requiring precognitive strategies are possible as well (Drop, 2016), see for example the work of van der El (2018).

Regarding the SOP, the last two stages of pursuit- and precognitive behaviour are thus relevant with respect to the goal of this research. When translating these stages into tracking tasks, they can be divided into two, based on the task variable that invokes the feedforward behaviour. With the display as the task variable, pursuit- and preview tracking tasks can be combined, while tracking tasks with predictable target signals have the forcing function as task variable. The research performed in these two directions will be explored in more detail in Sections 2.2.1 and 2.2.2. First, however, the model structures that are often used for these types of tracking tasks are elaborated on.

2.1.2. Model structures

Feedforward behaviour in human control is thus found in the last two stages of the SOP: pursuit/preview and precognitive control. Although attempts were made to unify all stages of the SOP in a single model (e.g. by Hess (1981), these have not been widely accepted. As the pursuit stage is more clearly defined and provides an obvious opportunity for an HO to utilize a feedforward component, the majority of research concerning feedforward control focused on tracking tasks with pursuit and preview displays. As Figure 2.1b already showed, the HO can respond to three signals: the target f_t, the output x and the (derived) error e.

However, problems arise when trying to measure the HOs responses to all three inputs. Due to the linear relation between f_t, e and x ($e = f_t - x$), the three-channel response system is overdetermined. Furthermore, in order to measure a separate HO control response, a separate disturbance function is required (Stapleford et al., 1967; van Paassen and Mulder, 1998), while only two useful disturbing functions can be inserted, as shown in Figure 2.3a. The figure shows the same control diagram as in Figure 2.1b, however, an additional disturbance signal f_d is added. Both signals have an obvious contribution: where the target function represents the signal to follow (such as a road along which to drive), the disturbance signal provides a perturbation that disturbs the tracking (such as a gust of wind). Adding another signal that has a separate function is then impossible.

Therefore, the three-channel model is often reduced to a two-channel model, see Figure 2.3. The figure shows the three-channel model in (a), and two reduced two-channel models in (b) and (c). For both two-channel models, the human responds to the target signal, as this signal allows for feedforward control behaviour. The other response that can then be measured is either the human's output to the controlled element output x or the error e. The two models in Figure 2.3b and Figure 2.3c are therefore

called the XT- and the ET model structures, respectively. Both model structures contain the missing responses (if active) in the available responses or subsystems. The used subsystems can thus be said to be contaminated with the missing subsystem (van der El et al., 2018b).

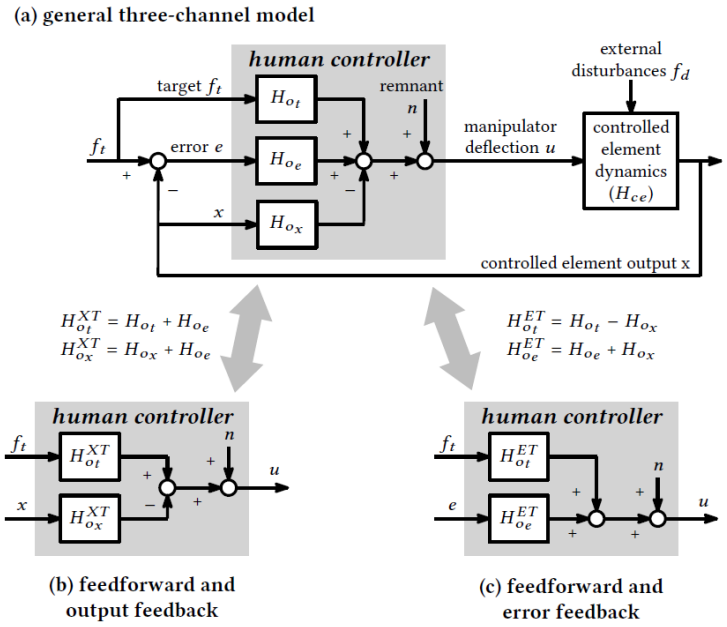


Figure 2.3: A three-channel block diagram of a human in a pursuit tracking task (a) and two reduced two-channel models in either the XT- (b) or the ET (c) configuration. Figure taken from van der El (2018).

Although a third two-channel model could also be formed (the EX-model), this structure is not as relevant as both the error and the control output signals are closed-loop signals. As this research focuses on feedback- and feedforward in manual control, this structure will not be used. The other two structures on the other hand, will both return. In previous research, the ET model structure was most commonly used (Wasicko et al., 1966; Sheridan, 1966; Drop et al., 2013), as it adds the feedforward target response to the compensatory error signal and thus allows for an easy extension of the cross-over model developed by (McRuer and Jex, 1967). However, van der El et al. (2016), showed that a well-fitting model based on control theoretical insights can be made for pursuit- and preview tracking tasks using the TX structure.

2.2. Tracking tasks including feedback- and feedforward

The previous section discussed the different stages of skill-based manual control behaviour. With the focus for this research on manual control with feedback- and feedforward subsystems, two types of tracking were considered to be interesting: tracking tasks with pursuit or preview displays and tracking tasks that use feedforward control. This section will further elaborate on these two types of tasks.

2.2.1. Pursuit- and preview tracking tasks

Over the years, much evidence was found regarding an increased performance of pursuit- and preview tracking tasks with respect to compensatory tasks (Wasicko et al., 1966; Pew et al., 1967), but widely accepted models such as the crossover model for compensatory tracking had not been found despite numerous attempts (Sheridan, 1966; Ito and Ito, 1975; McRuer et al., 1977; Hess and Modjtahedzadeh, 1989).

The majority of the proposed models for pursuit- and preview used the ET model structure, both in early research projects (Wasicko et al., 1966; Sheridan, 1966; McRuer et al., 1977) as in more recent work (Yu et al., 2014; Laurense et al., 2015; Zhang and Hoagg, 2016a; Yamagami et al., 2019). One reason for this is the easy extension with respect to the widely accepted cross-over model, as was previously

mentioned. Another reason behind this model structure is found with the internal model hypothesis (IMH) (Wolpert et al., 1998; Shadmehr and Mussa-Ivaldi, 1994). This hypothesis entails that humans are able to build an internal model of their body and its surroundings (Zhang and Hoagg, 2016a), which can then be applied in control. For the ET model structure this means that the human feedforward response approximates the inverse controlled element dynamics to achieve perfect-target tracking.

Recently, however, van der El et al. (2016) came up with a new model that appears to describe human control behaviour in preview tracking tasks well (Mulder et al., 2019). Instead of using an ET model structure, van der El et al. (2016) selected the TX model structure as the basis for their model. The top figure of Figure 2.4 shows this multi-loop model for pursuit- and preview tracking tasks. The feedback response to the system output x, is similar to that of a compensatory tracking task, consisting of a delay term, neuromuscular system (NMS) dynamics and a term H_{o_x} that adapts to the (CE) dynamics. This is in correspondence with the crossover model of McRuer and Jex (1967).

The target response is more elaborate. It consists of two parts: a 'far-viewpoint' response and a 'near-viewpoint' response. Both responses consist of 'negative delay' terms, meaning that the HO is looking into the future. The far-viewpoint response then consists of a low-pass filtering function H_{o_f} and the same 'compensatory' adaption term H_{o_x}. The near-viewpoint response, on the other hand, contains a high-pass filtering function H_{o_n} as it only responds to the high frequency components of the target signal. Both parts also consist of the same visual delay and NMS terms as the output response.

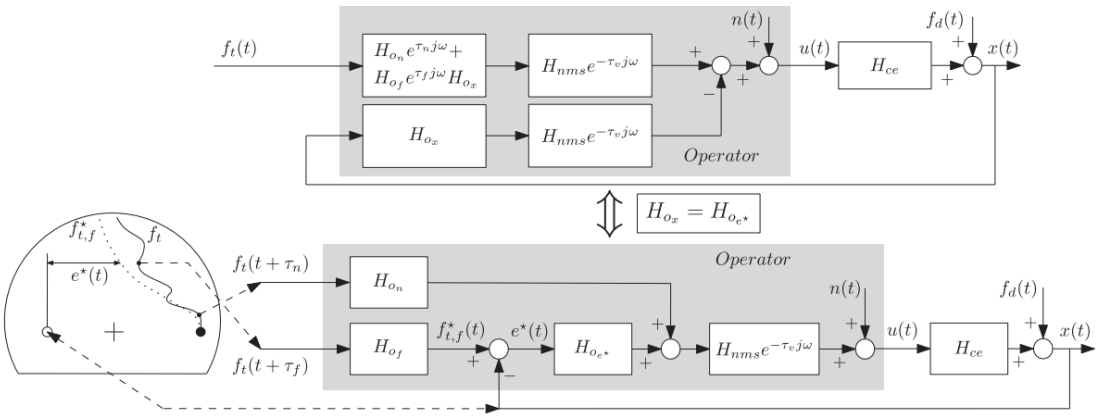


Figure 2.4: Block diagrams of the pursuit- and preview model proposed by van der El et al. (2016). Although the structure of the two models is different, the dynamics are identical. The figure is taken from van der El et al. (2016).

Taking a closer look at this initial model, it can be observed that the far-viewpoint target response can be merged with the system output response. When the visual delay and the NMS dynamics are separated, the model can be written as in the lower half of Figure 2.4. This model implies that the HO applies a low-frequency feedback strategy on the error $e^\star$ between the current output and the future target location, while applying high-frequency feedforward control to an earlier point of the target signal. In pursuit tracking, the near-viewpoint response disappears and the low-pass filtering function H_{o_f} reduces to a gain (van der El et al., 2016). In contrast to the ET model structure, perfect-target tracking is not achieved when the feedforward component approaches the inverse system dynamics. Instead, to achieve perfect-target tracking for this model structure, the feedforward channel should approximate the sum of both the inverse controlled element dynamics and the output dynamics H_{o_x} (van der El, 2018).

Further research using this model showed that the contribution due to the near-viewpoint response was very small, while coming at the cost of substantially more control activity (van der El et al., 2018b). In later research, this term is therefore omitted (van der El et al., 2020). The model on the bottom side of Figure 2.4 then reduces to a model consisting of a compensatory model with a filtering function H_{o_f} in front of it. With this simplification, the multi-loop structure of Figure 2.3 has been replaced by a single

loop structure, which makes it more similar to the previously proposed model by Ito and Ito (1975).

It is, however, important to realise that even though the multi-loop structure can be replaced by a single loop structure, identification of the system still happens based on the multi-loop Fourier Coefficients (FC) method as described by Wasicko et al. (1966); van Paassen and Mulder (1998). This method is described more extensively in Section 2.3.1. The main conclusion of this research by van der El (2018) is that the proposed model is well able to model human control behaviour with pursuit- and preview displays, even when neglecting the contribution of the near-viewpoint response. Furthermore, this model is the first one that explicitly identifies how humans use pursuit and mainly preview in manual control (van der El et al., 2016).

2.2.2. Predictable target signals

Besides pursuit- and preview displays, another way to allow for humans to utilize feedforward control behaviour is by using predictable target signals. Although evidence for this has been presented in multiple publications such as Wasicko et al. (1966); Yamashita (1990), attempts to analyse this using system identification and parameter estimation are scarce. One reason for this is that common systems identification methods that use Fourier Coefficients give problems when applied to predictable signals like ramps, which have power at all frequencies. Therefore other options such as auto-regressive exogenous (ARX) model methods (Section 2.3.2) are to be used (Drop et al., 2016, 2018).

For the attempts that have been made to identify frequency domain models of human controllers, the model structure mainly used is the ET structure (bottom right of Figure 2.3) (Drop et al., 2013; Laurense et al., 2015). In these papers, the feedback response to the error (H_{o_e}) is modeled using a structure identical to that of compensatory tracking tasks (McRuer and Jex, 1967), while the feedforward response (H_{o_t}) can be modelled with a gain, inverse system dynamics, a low-pass filter and a time delay. Although these studies confirm that predictable signals affect control behaviour, it remains unclear how this is done exactly.

2.3. Identification methods

In order to identify the models describing feedback- and feedforward behaviour in human controllers, system identification methods are used. Over the past decades, several different methods have been used to identify these systems, each with different advantages and disadvantages. This section will provide an overview of the most relevant methods that are able to identify different control responses from a single set input-output data.

When evaluating the different identification methods, a couple of distinguishing characteristics can be observed. First, the domain in which the identification method operates. Some methods first translate the time response to the frequency domain and estimate the model parameters from there. Others directly estimate the parameters from time responses. Another characteristic is the ability of a method to identify a system recursively or not. Recursive methods have the advantage to be able to identify time-varying behaviour.

2.3.1. Multi-loop Fourier Coefficients method

The multi-loop (FC) method is the most conventional method when it comes to the different options (van Paassen and Mulder, 1998; Stapleford et al., 1967). With the help of an additional forcing function, it estimates the frequency response functions (FRFs) of the different responses, after which a non-linear solver can be used to fit the parameters to the responses.

The working of this technique is as follows. To start, the TX model structure from Figure 2.3 (b) is used to derive the Fourier transform of the HO control output in terms of the target function f_t and the system output x at the target function frequencies ω_t, see Equation (2.1). Here $U(j\omega_t)$, $F_t(j\omega_t)$ and $X(j\omega_t)$ are the Fourier transforms of u, f_t and x. Furthermore, H_{o_t} and H_{o_x} represent the target- and system output response dynamics, respectively.

$$U(j\omega_t) = H_{o_t}(j\omega_t) F_t(j\omega_t) - H_{o_x}(j\omega_t) X(j\omega_t) \tag{2.1}$$

This equation contains 2 unknowns (H_{o_t} and H_{o_x}), and can therefore not be solved right away. However, interpolating the Fourier coefficients of the three signals at the disturbance frequencies to the target frequencies provides a second equation with $\tilde{U}(j\omega_t)$, $\tilde{F}_t(j\omega_t)$ and $\tilde{X}(j\omega_t)$. Equation (2.2) shows the resulting system of equations.

$$\begin{bmatrix} U(j\omega_t) \\ \tilde{U}(j\omega_t) \end{bmatrix} = \begin{bmatrix} F_t(j\omega_t) & -X(j\omega_t) \\ \tilde{F}_t(j\omega_t) & -\tilde{X}(j\omega_t) \end{bmatrix} \begin{bmatrix} H_{o_t}(j\omega_t) \\ H_{o_x}(j\omega_t) \end{bmatrix} \tag{2.2}$$

The system of equations of Equation (2.2) can then be solved to find $\hat{H}_{o_t}(j\omega_t)$ and $\hat{H}_{o_x}(j\omega_t)$ as shown in Equations (2.3) and (2.4). This can in the same way be solved for the disturbance frequencies ω_d.

$$\hat{H}_{o_t}(j\omega_t) = \frac{\tilde{U}(j\omega_t) X(j\omega_t) - U(j\omega_t) \tilde{X}(j\omega_t)}{\tilde{F}_t(j\omega_t) X(j\omega_t) - F_t(j\omega_t) \tilde{X}(j\omega_t)} \tag{2.3}$$

$$\hat{H}_{o_x}(j\omega_t) = \frac{\tilde{U}(j\omega_t) F_t(j\omega_t) - U(j\omega_t) \tilde{F}_t(j\omega_t)}{\tilde{F}_t(j\omega_t) X(j\omega_t) - F_t(j\omega_t) \tilde{X}(j\omega_t)} \tag{2.4}$$

Then, two subsystem FRFs are obtained for H_{o_t} and H_{o_x}. These FRFs can be used to estimate the frequency domain model parameters as given in Figure 2.4. This is done with a non-linear solver such as `fmincon` or `fminsearch` in MATLAB.

Overall, it can be concluded that the mutli-loop FC method is reliable and well-working, and the translation to the frequency domain allows for easy estimation of certain model structures. However, multi-sine signals with high signal-to-noise ratios are needed and per identified response a separate disturbance function is required (van Lunteren, 1979).

2.3.2. Identification using linear time-invariant models

Linear time-invariant models (LTI) such as autoregressive exogenous models (ARX) (Nieuwenhuizen et al., 2008; van Grootheest et al., 2018; Drop et al., 2018) or autoregressive moving-average exogenous (ARMAX) models (Vos et al., 2014) are often used for identifying unknown systems (Ljung, 1999). These methods work by fitting a discrete polynomial regression model on the time-domain data of the system in- and outputs(Vos et al., 2014). For ARX models for example, this can be done with a simple Ordinary Least Squares (OLS) or Recursive Least Squares (RLS) regression, making it a computationally efficient method. However, the noise model is coupled to the model structure and data pre-processing might be required to reduce noise levels. On the other hand, the ARMAX models, describes the error as a moving average of white noise. This restricts this method from using OLS. Finally, both models can be fit recursively, thus it allows to identify time-varying behaviour.

2.3.3. Time domain maximum likelihood estimation

Another method presented by Zaal et al. (2009) considers time domain maximum likelihood estimation, which directly fits the parameters to the time domain data. As it has fewer requirements for the forcing functions and fitting is done in one step, there are some advantages to this method. However, increased computational complexity and difficulties with recognizing the model structure from the response do cause problems with this technique. Furthermore, the technique does not guarantee to find the global optimum. Zaal et al. (2009) combined this technique with a genetic algorithm to improve the performance on this last point.

2.3.4. Kalman filtering

Although Kalman filters have been applied when estimating manual control behaviour (Popovici et al., 2017), this has only been done for compensatory tasks. However, the computational efficiency and the recursive properties of the filter make it a promising candidate for identifying feedback- and feedforward behaviour. For this research, however, the method is not relevant.

2.3.5. Subsystem identification

SSID is a new identification method developed by Zhang and Hoagg (2016a). Similar to the Fourier Coefficient method, it first translates the time response to the frequency domain before estimating the parameters. The large difference is that it requires only one forcing function and estimates two subsystems from the single closed loop response. This technique will be the main topic of this research and is treated in the next chapter.

2.3.6. Predictor-Based Subspace Identification

A misunderstanding can easily occur when comparing the name of the previously mentioned subsystem identification with another technique called subspace identification or predictor-bases subspace identification (PBSID). This technique, based on the work of van Wingerden and Verhaegen (2009) and applied in manual control by Duarte et al. (2017) for a compensatory task, identifies linear parameter varying (LPV) systems. It is a time-varying method (as opposed to SSID) that can identify time-varying operator equalisation parameters. However, problems arise regarding the estimation of the time delay. Furthermore, this method has only been applied to single loop systems and is thus not relevant for this research.

2.4. Synthesis

From the overview of the identification and modelling of feedback- and feedforward behaviour in manual control, a number of conclusions can be drawn. First, when modelling feedback- and feedforward behaviour, the last two stages of the SOP are relevant. Especially the second stage of pursuit- and preview tracking tasks allows for direct modelling of feedback and feedforward. However, due to an overdetermined system, the complete three-channel model structure is reduced and either the ET or the XT model structures are used for modelling such control behaviour.

From the two types of tracking tasks that allow the use of feedback- and feedforward control, pursuit/preview tasks and tasks using predictable target signals, the first one is better understood. For these pursuit- and preview tasks, the recently developed model by van der El et al. (2016) is both well-working and intuitive. Furthermore, the multi-sine target signals used for identification work well with frequency domain identification techniques, something that is more difficult when using predictable target signals that often have power at all frequencies.

Lastly, from the number of existing identification techniques for feedback- and feedforward control, the multi-loop FC method and the newly developed SSID technique both operate in the frequency domain and therefore show the largest resemblance.

3

Subsystem identification

When modelling humans as controllers, system identification techniques are essential and with the rapid developments in this research field, many new techniques arise. One of these new techniques, called subsystem identification (SSID) identifies feedback- and feedforward subsystems in the frequency domain. By using a candidate pool approach, the method guarantees asymptotic stability of the identified closed loop system. However, to judge this technique with respect to conventional techniques, a comparison in performance and usability should be made.

For comparing the new SSID technique, it is important to gain a thorough understanding of its working principles. This chapter will therefore elaborate on the algorithm and the different extensions that have been published. First, the problem will be formulated, after which the basic algorithm is discussed. Then, the different extensions of this first algorithm will be evaluated, followed by an explanation of the most relevant extension. Finally, the algorithm is summarized.

3.1. Problem formulation

As mentioned, the SSID technique identifies the feedback and feedforward subsystems encountered in for example pursuit- and preview tasks or compensatory tracking tasks with predictable signals. The technique, developed and used in (Zhang and Hoagg, 2016a,b; Koushkbaghi et al., 2018; Zhang et al., 2018; Sheffler et al., 2019), uses an ET-model to estimate the responses. This model, simplified from the three-channel model used for pursuit- and preview displays as provided in Chapter 2, is provided in Figure 3.1. In this figure, r, y, σ_r and σ_y are the input, output, input noise and output noise, respectively. Regarding the different subsystems, the controlled element dynamics (H_{CE}) are known, but the dynamics of feedforward (H_{o_t}) and error feedback (H_{o_e}) subsystems are to be identified.

A last remark regarding the figure involves the noise, which is inserted at f_r and y in the form σ_r and σ_y. Generally, manual control models contain noise insertions *inside* the loop (e.g. see Figure 2.4). Having the noise components inserted before, but especially after the closed loop is unconventional. Especially the noise at y does not represent the non-linear control behaviour of humans.

If the two noise terms are 0, then the closed loop transfer function from the input to the output $\tilde{H}$ is given by Equation (3.1). Besides, let $\{H(\omega_k) = \frac{y(j\omega_k)}{r(j\omega_k)}\}_{k=1}^{N}$ be the closed loop frequency response data, where $\omega_k \in [\omega_1, \omega_2, \cdots, \omega_N]$ being the positive frequencies of increasing magnitude (i.e., $\omega_1 < ... < \omega_N$). N thus resembles the number of frequency response data points, which is selected to be sufficiently large but is independent of the discrete fourier transform (DFT).

$$\tilde{H}(s) \equiv \frac{H_{CE}(s)H_{o_t}(s) + H_{CE}(s)H_{o_e}(s)}{1 + H_{CE}(s)H_{o_e}(s)} \tag{3.1}$$

Now, for the basic algorithm, the subsystems H_{CE}, H_{o_t} and H_{o_e} can be modeled according to Equation (3.2), whereas for some of the extensions delay terms are added to the feedback- and feedforward

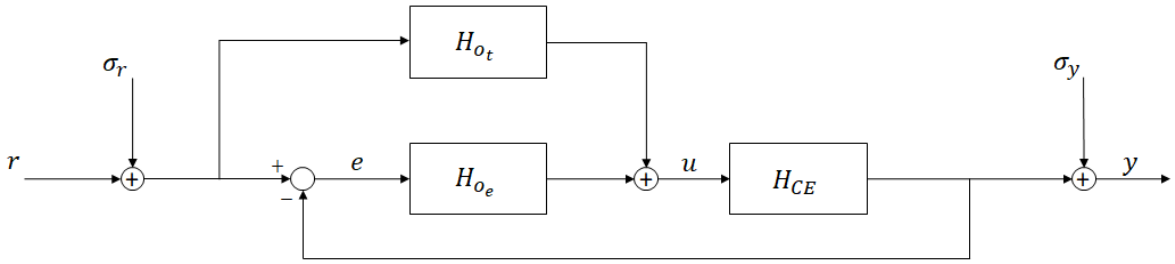


Figure 3.1: Block diagram of the ET system for which H_{ot} and H_{oe} are to be identified.

subsystems. The numerators and denominators of the separate subsystems are coprime polynomials, meaning they do not have a shared root. Furthermore, all denominators are monic, meaning that the coefficient that the leading coefficient (i.e. the coefficient that belongs to the highest order of the polynomial) is equal to 1.

$$H(s) = \frac{N(s)}{D(s)} \tag{3.2}$$

In order to fit the proposed model structure to the frequency response data points, a cost function is set up according to Equation (3.3). The cost function expresses the absolute difference between the frequency response data $\{H(\omega_k)$ and the results of the evaluated model. Minimising this cost function will result in the identified transfer functions for H_{o_t} and H_{o_e}. The following sections will elaborate on the exact working of the algorithms, starting with the basic version in the next section. However, as sign conventions throughout the research differs quite a bit, first an overview of these differences is provided.

$$J(H_{o_t}, H_{o_e}) \equiv \sum_{k=1}^{N} \left|\tilde{H}(H_{o_t}, H_{o_e}, \omega_k) - H(\omega_k)\right|^2 \tag{3.3}$$

3.1.1. Sign conventions of subsystems

Throughout the different papers and this report, the symbols of the different subsystems change various times. When looking at the feedforward subsystem for example, the symbol H_{o_t} is mainly used throughout this report. In the research on SSID, Zhang and Hoagg (2016a) first call this subsystem G_1. However, in the publications that follow, it is being renamed to G_{ff}. To give a clear overview of the symbols used in the different publications, Table 3.1 lists the names of the different subsystems. Naturally, the names of the subsystems are also found in the numerator and denominator terms as was given by Equation (3.2) Throughout this chapter, the conventions used in the respective research will be followed, to not confuse the reader when comparing this report with the actual papers.

Table 3.1: Overview of sign conventions throughout the research on SSID

Research	FF SS	FB SS	CE SS	Closed loop system
General	H_{o_t}	H_{o_e}	H_{CE}	$\tilde{H}$
Zhang and Hoagg (2016a)	G_1	G_2	G_0	$\tilde{G}$
Zhang and Hoagg (2016b)	G_{ff}	G_{fb}	G_y/G_v	$\tilde{G}$
Koushkbaghi et al. (2018)	$G_{ff,p}$	G_{fb}	$G+\eta$	-
Zhang et al. (2018)	G_{ff}	G_{fb}	G	$\tilde{G}_{yr}$
Sheffler et al. (2019)	G_{ff}	G_{fb}	G_d	$\tilde{G}_{yr}$

Another difference found throughout the publications are the symbols of the (to be explained) parameter vectors of the coefficients of the feedback- (FB) and feedforward (FF) subsystems. Whereas these are

called β and ϕ in the first two publications of SSID by Zhang and Hoagg (2016a,b), they are replaced by α and β in the later papers of Zhang et al. (2018) and Sheffler et al. (2019).

Finally, it should be stressed that the N used throughout this chapter represents the amount of data points of the frequency response function (as explained before). This is in contrast with the N used in combination with the DFT, where it represents the amount of samples of a time series (and thus the Fourier Transform).

3.2. Basic technique

The initial algorithm that was published in Zhang and Hoagg (2016a) works in the s-domain and, as mentioned before, only identified feedback- and feedforward transfer functions. Time delays were thus not included. Figure 3.2 shows the model with the subsystems sign convention according to the paper, as was explained in the previous section. The system is a single-input single-output (SISO) system. With the identified subsystems assumed to have the shape of Equation (3.2), Equation (3.1) can be rewritten to Equation (3.4). Note that the $N_{0/1/2}$ and $D_{0/1/2}$ terms correspond to the numerators and denominators of the subsystems $G_{0/1/2}$ as shown in Figure 3.2. $\tilde{G}$ is then the closed-loop transfer function. Furthermore, the dependencies of s are omitted for brevity.

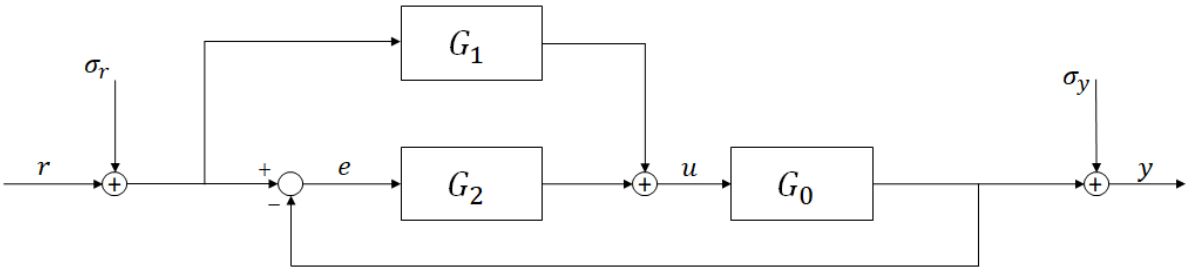


Figure 3.2: Block diagram of the ET system with sign conventions according to Zhang and Hoagg (2016a).

$$\tilde{G} = \frac{N_0[N_1D_2 + N_2D_1]}{D_1[N_0N_2 + D_0D_2]} \tag{3.4}$$

Considering the degrees ($n \equiv deg(N)$ and $d \equiv deg(D)$), a number of assumptions are made.

1. d_1, d_2, n_1, n_2 are known. This implies that the order of each transfer function is known.
2. $d_0 + d_2 > n_0 + n_2$, meaning that H_0H_2 is strictly proper (a transfer function is proper when the order of the numerator does not exceed the degree of the denominator).
3. $N > d_0+d_1+d_2+n_0+max\{n_1+d_2, n_2+d_1\}$. This implies that there is a sufficiently large amount of frequency response data points. This is required to avoid the possibility of each frequency being a zero of the to be defined $\tilde{\mathcal{N}}_1(s,\phi)$ (Equation (3.8)), see also Appendix A, Proof of Proposition 1 in (Zhang and Hoagg, 2016a).
4. If $\lambda \in \mathbb{C}$ and $D_1(\lambda)[N_0(\lambda)N_2(\lambda) + D_0(\lambda)D_2(\lambda)] = 0$, then $Re(\lambda) < 0$. This last assumption guarantees that the poles of the closed-loop system are in the left-half plane and thus guarantees that the closed-loop system is asymptotically stable.

In the next step, the coefficients of the subsystems will be parameterized. Define ϕ as the set of coefficients of the feedforward denominator, feedback numerator and feedback denominator (in that specific order) and β as the set of feedforward numerator coefficients. In order to identify the parameters, possible combinations of ϕ will be evaluated to find an optimum value for J. The corresponding values for β are computed during this optimization. This will be explained later Section 3.2.2.

First, define $d \equiv d_1 + n_2 + d_2 + 1$ and for all non-negative integers j let $\Gamma_j(s) \equiv [s^j \quad s^{j-1} \quad \cdots \quad s \quad 1]^T$. Next, define the following functions

$$\mathcal{N}_1 \equiv \Gamma_{n_1}^T(s)\beta, \quad \mathcal{D}_1 \equiv s^{d_1} + \Gamma_{d_1-1}^T(s)E_1\phi, \quad \mathcal{N}_2 \equiv \Gamma_{n_2}^T(s)E_2\phi, \quad \mathcal{D}_2 \equiv s^{d_2} + \Gamma_{d_2-1}^T(s)E_3\phi \tag{3.5}$$

$$\mathcal{G}_1(s,\beta,\phi) \equiv \frac{\mathcal{N}_1(s,\beta)}{\mathcal{D}_1(s,\phi)}, \quad \mathcal{G}_2(s,\phi) \equiv \frac{\mathcal{N}_2(s,\phi)}{\mathcal{D}_2(s,\phi)} \tag{3.6}$$

Here, $E_1 \equiv [I_{d_1} \quad 0_{d_1\times(d_2+n_2+1)}] \in \mathbb{R}^{d_1\times d}$, $E_2 \equiv [0_{(n_e+1)\times d_1} \quad I_{n_2+1} \quad 0_{(n_2+1)\times d_2}] \in \mathbb{R}^{(n_e+1)\times d}$ and $E_3 \equiv [0_{d_e\times(d_1+n_e+1)} \quad I_{d_2}] \in \mathbb{R}^{d_e\times d}$. Also, as mentioned earlier, $\beta \in \mathbb{R}^{n_1+1}$ and $\phi \in \mathbb{R}^d$. By substituting the terms from Equation (3.5) in Equation (3.4), a new function $\tilde{\mathcal{G}}$ can be defined. This function is the closed-loop transfer function in terms of β and ϕ and is given by the first part of Equation (3.7). The right hand side of Equation (3.7) shows a simplified version of the original equation, with $\tilde{\mathcal{D}}(s,\phi)$, $\tilde{\mathcal{N}_1}(s,\phi)$ and $\tilde{\mathcal{N}_2}(s,\phi)$ given by Equations (3.8) to (3.10). These terms are only dependent on ϕ, one entry in the candidate pool and β is thus separated. The reason for this is discussed later, as first the candidate pool is discussed in further detail.

$$\tilde{\mathcal{G}}(s,\beta,\phi) \equiv \frac{N_0(s)\left[\mathcal{D}_2(s,\phi)\,\mathcal{N}_1(s,\beta)+\mathcal{D}_1(s,\phi)\,\mathcal{N}_2(s,\phi)\right]}{\mathcal{D}_1(s,\phi)\left[D_0(s)\,\mathcal{D}_2(s,\phi)+N_0(s)\,\mathcal{N}_2(s,\phi)\right]} = \frac{\tilde{\mathcal{N}_1}(s,\phi)\beta+\tilde{\mathcal{N}_2}(s,\phi)}{\tilde{\mathcal{D}}(s,\phi)} \tag{3.7}$$

$$\tilde{\mathcal{N}_1}(s,\phi) \equiv N_0(s)\mathcal{D}_2(s,\phi)\Gamma_{n_1}^{\top}(s) \tag{3.8}$$

$$\tilde{\mathcal{N}_2}(s,\phi) \equiv N_0(s)\mathcal{D}_1(s,\phi)\mathcal{N}_2(s,\phi) \tag{3.9}$$

$$\tilde{\mathcal{D}}(s,\phi) \equiv \mathcal{D}_1(s,\phi)\left[D_0(s)\mathcal{D}_2(s,\phi)+N_0(s)\mathcal{N}_2(s,\phi)\right] \tag{3.10}$$

3.2.1. Candidate pool

The parameter ϕ thus contains one entry from the candidate pool with all the coefficients of the feedforward denominator ($\mathcal{D}_1$) and the feedback numerator ($\mathcal{N}_2$) and denominator ($\mathcal{D}_2$). For every coefficient, a range of possible values is determined based on expectations and the initial candidate pool Π consists of the set of all possible combinations of the different coefficients.

However, as the candidate pool was the reason the SSID technique guarantees an asymptotically stable identified system, this needs to be refined. For the closed-loop transfer function to be stable, the closed-loop denominator $\tilde{\mathcal{D}}$ needs to have the roots in the left-half of the complex plane. For this basic SSID method, where no delay term is present, the closed-loop system is then guaranteed to be stable when $\tilde{\mathcal{D}}$ satisfies the Routh-Hurwitz criterion. The domain $\mathcal{S}$ is therefore defined as the region of values of ϕ for which $\tilde{\mathcal{D}}$ has the roots in the left-half plane (or $\mathcal{S} \equiv \left\{\phi \in \mathbb{R}^d : \tilde{\mathcal{D}}(s,\phi) \text{ is Hurwitz}\right\}$). The candidate pool Φ is then defined as $\Phi \equiv \Pi \cap \mathcal{S}$. It contains M elements ($\phi_1 \ldots \phi_M$) and define $\mathcal{M} \equiv \{1, 2, \ldots, M\}$. Furthermore, each element is unique (i.e., if $i \neq j$, then $\phi_i \neq \phi_j$).

The definition given above holds in case the true parameters β_* and ϕ_* are in the candidate pool. In practice, however, this is often not the case. Therefore, the candidates in the candidate pool should provide a certain margin regarding stability. Therefore, the parameter $\rho < 0$ is defined. From assumption 4 (see before), it is defined that the true parameters result in an asymptotically stable system. If ρ is small enough, it can be ensured that $\mathcal{S}_\rho \equiv \left\{\phi \in \mathbb{R}^d : \tilde{\mathcal{D}}(s+\rho,\phi) \text{ is Hurwitz}\right\}$ contains ϕ_*. This modified space $\mathcal{S}_\rho$ is required in (Zhang and Hoagg, 2016a) to proof that the algorithm identifies parameters that are arbitrarily close to the true parameters (which are not in the candidate pool), if the candidate pool is sufficiently dense. In practice, the candidate pool is thus generated with a small stability margin defined by ρ and is defined as $\Phi \equiv \Pi \cap \mathcal{S}_\rho$.

There is one more pitfall regarding the candidate pool. Depending on the ranges of values selected for the different coefficients, it might occur that two different sets of Φ, for example ϕ_i and ϕ_j, combined with the identified β_i and β_j, result in the same closed-loop transfer function $\tilde{\mathcal{G}}(s)$. If this exact configuration would turn out to be the optimal solution, two possibilities remain. As β is still unknown when forming the candidate pool, this cannot be checked when forming the candidate pool. Therefore, this should be verified after running the algorithm by checking if there are multiple minima of the cost function.

3.2.2. Identification algorithm

To identify the parameters of the feedback- and feedforward subsystems, the simplified function on the right side of Equation (3.7) needs to be combined with the cost function of Equation (3.3). Inserting $\tilde{\mathcal{G}}(s,\beta,\phi)$ for $\tilde{H}$ in Equation (3.3) and evaluating at the frequencies of the frequency response data $H(\omega_k)_{k=1}^{N}$ gives Equation (3.11). The right-hand side of the equation shows how the cost function is split up into two parts: one being dependent on β and one being independent of β. This is because β is the unknown in the equation, as ϕ is assumed from the candidate pool.

$$J(\beta,\phi)=\sum_{k=1}^{N}\left|\frac{\tilde{\mathcal{N}}_1(j\omega_k,\phi)\beta+\tilde{\mathcal{N}}_2(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}-H(\omega_k)\right|^2=\sum_{k=1}^{N}\left|\frac{\tilde{\mathcal{N}}_1(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}\beta+\left(\frac{\tilde{\mathcal{N}}_2(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}-H(\omega_k)\right)\right|^2 \tag{3.11}$$

Equation (3.12) provides a relation to get rid of the squared term, where c is a complex number. Applying this to the right side of Equation (3.11) and working out the product results in Equation (3.13).

$$|c|^2=c^*c \tag{3.12}$$

$$\begin{aligned}J(\beta,\phi)=\sum_{k=1}^{N}&\beta^T\left(\frac{\tilde{\mathcal{N}}_1(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}\right)^*\frac{\tilde{\mathcal{N}}_1(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}\beta+\beta^T\left(\frac{\tilde{\mathcal{N}}_1(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}\right)^*\left(\frac{\tilde{\mathcal{N}}_2(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}-H(\omega_k)\right)+\\&\left(\frac{\tilde{\mathcal{N}}_2(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}-H(\omega_k)\right)^*\frac{\tilde{\mathcal{N}}_1(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}\beta+\left(\frac{\tilde{\mathcal{N}}_2(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}-H(\omega_k)\right)^*\left(\frac{\tilde{\mathcal{N}}_2(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}-H(\omega_k)\right)\end{aligned} \tag{3.13}$$

By defining Ω_0, Ω_1 and Ω_2 as in Equations (3.14) to (3.16), respectively, Equation (3.13) can be simplified to Equation (3.17). During this simplification, Equation (3.12) has been used for forming Ω_2. Besides, the second and third terms of Equation (3.13) have been merged according to Equation (3.18) (c_1 and c_2 are arbitrary complex numbers) to form Ω_1. For Ω_2, the imaginary parts cancel during the multiplications with β and thus only the real part is used. This cancellation occurs because the product of $\frac{\tilde{\mathcal{N}}_t}{\tilde{\mathcal{D}}}^*\frac{\tilde{\mathcal{N}}_t}{\tilde{\mathcal{D}}}$ is a Hermitian (or self-adjoint) matrix, which is a matrix that is equal to its conjugate transpose.

$$\Omega_0(\phi)\equiv\sum_{k=1}^{N}\left|\frac{\tilde{\mathcal{N}}_2(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}-H(\omega_k)\right|^2\in\mathbb{R} \tag{3.14}$$

$$\Omega_1(\phi)\equiv 2\,\mathrm{Re}\sum_{k=1}^{N}\left[\frac{\tilde{\mathcal{N}}_1(j\omega_k,\phi)}{\tilde{\mathcal{D}}(\mu_k,\phi)}\right]^*\left[\frac{\tilde{\mathcal{N}}_2(j\omega_k,\phi)}{\tilde{\mathcal{D}}(l\omega_k,\phi)}-H(\omega_k)\right]\in\mathbb{R}^{n_1+1} \tag{3.15}$$

$$\Omega_2(\phi)\equiv\mathrm{Re}\sum\left[\frac{\tilde{\mathcal{N}}_1(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}\right]^*\left[\frac{\tilde{\mathcal{N}}_1(j\omega_k,\phi)}{\tilde{\mathcal{D}}(j\omega_k,\phi)}\right]\in\mathbb{R}^{(n_1+1)\times(n_1+1)} \tag{3.16}$$

$$J(\beta,\phi)=\beta^{\mathrm{T}}\Omega_2(\phi)\beta+\Omega_1^{\mathrm{T}}(\phi)\beta+\Omega_0(\phi) \tag{3.17}$$

$$c_1^*c_2+c_2^*c_1=2Re\left(c_1^*c_2\right) \tag{3.18}$$

Equation (3.17) is convex in β and as the candidate pool guarantees asymptotic stability of the system, Ω_2 is positive definite. To clarify, convex means the function points 'outwards' and thus has a single minimum; in this example the function has a quadratic form in terms of β and thus there is one minimum for it. As the cost function is convex, the maximum value of β can be found by setting the derivative equal to 0 and solving for β, see Equation (3.19). This is the global minimizer of J provided that ϕ is known.

$$\frac{\partial J}{\partial\beta}=2\Omega_2\beta+\Omega_1=0,\rightarrow\beta=-\frac{1}{2}\Omega_2^{-1}\Omega_1 \tag{3.19}$$

As each element in the candidate pool, as defined in Section 3.2.1, is unique, the minimum value of J can be found. Define $\mathcal{J}_i(\beta) \equiv J(\beta, \phi)\big|_{\phi=\phi_i}$, then for each element in Φ, the global minimum is found by Equation (3.20).

$$\mathcal{J}_i\left(-\frac{1}{2}\Omega_2^{-1}(\phi_i)\,\Omega_1(\phi_i)\right) = \Omega_0(\phi_i) - \frac{1}{4}\Omega_1^{\mathrm{T}}(\phi_i)\,\Omega_2^{-1}(\phi_i)\,\Omega_1(\phi_i) \tag{3.20}$$

With this equation, the value for $\mathcal{J}$ can be found for all ϕ_i. Then $\mathcal{J}_l(\beta_l) = min_{i\in\mathcal{M}}\mathcal{J}_i(\beta_i)$. The identified parameters are then $\beta^+ \equiv \beta_l$ and $\phi^+ \equiv \phi_l$. They form the identified transfer functions $G_1^+(s) = \mathcal{G}_1(s, \beta^+, \phi^+)$ and $G_2^+(s) = \mathcal{G}_2(s, \phi^+)$ (see Equation (3.6)) and approximate the true parameters β^* and ϕ^* if the candidate pool is sufficiently dense and the noise is sufficiently low. This is further elaborated on in Chapter 4.

3.3. Algorithm extensions

Besides the basic algorithm first mentioned by Zhang and Hoagg (2016a), several extensions to this algorithm have been published. First, Zhang and Hoagg (2016b) extended the algorithm by making it applicable to identify subsystems for a multiple-input multiple-output (MIMO) system, by allowing the output to be different from the feedback. Figure 3.3 shows the block diagram used for this research. Besides the two outputs of G_y and G_v, another difference that is observed from the block diagram is the number of noise terms that are used. Here, noise (in the form of γ) is added at numerous points. Some of these, such as the noise at e and u but also at r, are now inserted inside the loop. This implies that the noise is generated by the human itself, which is significantly more realistic compared to the first version of the algorithm. However, the noise input at y remains unconventional and counter-intuitive.

Besides the modifications that can be seen from the diagram, there are some additional changes. Namely, the method works in discrete time and assumes a feedforward subsystem to be modelled as a Finite Impulse Response (FIR). This last part means that only the numerator coefficients are assumed, transforming the transfer function into the form of Equation (3.21). As this requires fewer coefficients to be estimated (the denominator coefficients disappear), the method has an increased computational efficiency.

$$G_{ff} = a_n z^0 + a_{n-1} z^{-1} + \ldots + a_0 z^{-n} \tag{3.21}$$

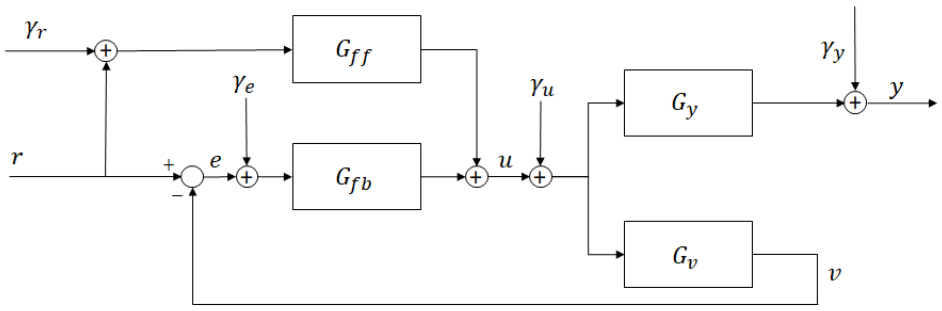


Figure 3.3: Simplified block diagram of the MIMO system with two controlled element outputs according to Zhang and Hoagg (2016b).

Although most of the modifications with respect to the basic technique can be considered to be improvements, not all of them are equally relevant. Mainly the output subsystems G_y and G_v are generally the same. This is also something that did not come back in consequent work on SSID. Besides, the described algorithm identifies the same subsystems as was done in the initial version (Zhang and Hoagg, 2016a), and time delays are still not identified. As the extensions following this version of Zhang and Hoagg (2016b) did allow to estimate time delays, this version will not be further evaluated for this report.

Another version of the SSID algorithm involves non-linear outputs of the controlled element dynamics(Koushkbaghi et al., 2018). Figure 3.4 shows the model for which parameters are identified in this algorithm. As can be seen, multiple feedforward subsystems are identified and the input to these subsystems is non-linear. Furthermore, the block containing η is a nonlinear function of $\bar{y}_k$. This algorithm was developed to investigate whether humans are able to perform dynamic inversion when controlling a plant with output non-linearities. This would then possibly support the internal model hypothesis (see Section 2.2.1 for nonlinear systems). The algorithm is used in the z-domain, cannot estimate time delays and uses a single candidate pool. As the current research project focuses on the control of linear systems and this method does not allow for estimation of time delay parameters, no further evaluation of this algorithm will be performed.

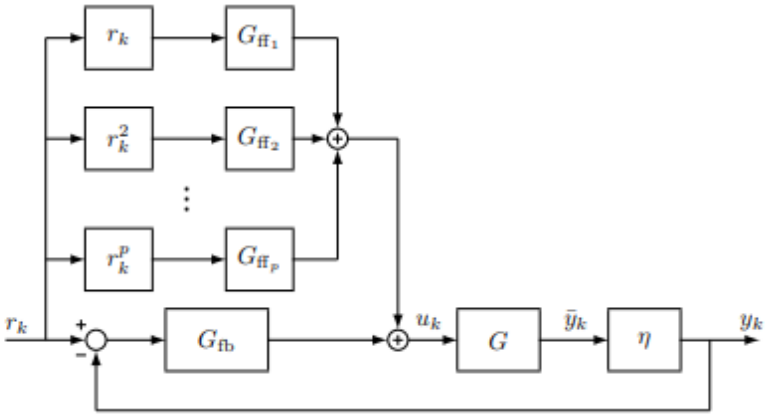


Figure 3.4: Block diagram of the model with multiple feedforward subsystems with nonlinear inputs according to Koushkbaghi et al. (2018).

The next publication that uses the SSID technique describes a SISO algorithm that identifies both a feedback- and a feedforward subsystem as well as a feedback delay using a discrete time approach (Zhang et al., 2018). The same method is applied by Matveeva et al. (2016). Figure 3.5 shows the block diagram of this model, where the z-domain is used. With Equation (3.22) providing the transform from the s-domain to the z-domain (with T_s being the sampling time), the normal delay term e^{-sd} becomes z^{-d}. Note that d is then an integer number that represents the amount of time steps T_s that add up to the delay. To identify the feedback time delay d, the delay factor is added as a coefficient in the candidate pool. The algorithm itself does therefore not differ much from the basic technique described in Section 3.2. Regarding the noise, no noise input is applied in this research. This is unconventional, as the human operator (HO) is generally modelled as a quasi-linear system (McRuer and Jex, 1967). However, considering the fact that it is derived from earlier versions, the noise can be expected to behave fairly similar.

$$z = e^{sT_s} \tag{3.22}$$

The modifications in this paper appear to be quite relevant, as similar research on models that use feedback- and feedforward in manual control (van der El et al., 2016; Drop et al., 2013) also assumes a feedback time delay. However, most of these models also use some sort of feedforward time delay, which is a logical expectation. Therefore the method used by Sheffler et al. (2019), which directly extends the method of Zhang et al. (2018) by adding a feedforward time delay (see Figure 3.6), is even more relevant. Note again that the time delays τ_{ff} and τ_{fb} are integer multiples of the time step T_s. In order to identify all the parameters in this algorithm, two candidate pools are used. Just as previously commented regarding (Zhang et al., 2018), no noise is added. Therefore, the same comments as mentioned shortly before hold for this method. Overall, this version of SSID is most applicable to cybernetics and is therefore further explained in the next section.

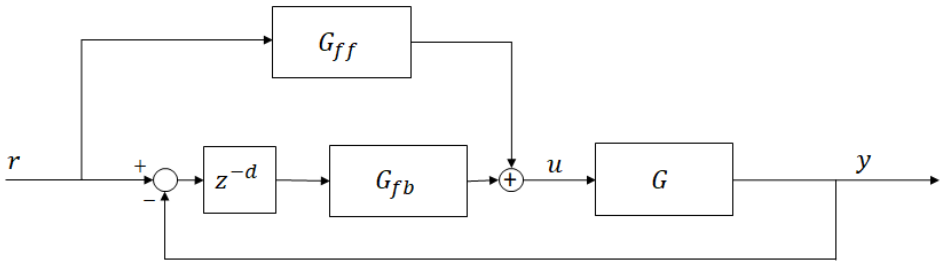


Figure 3.5: Block diagram of the model for which feedback- and feedforward subsystems as well as a feedback delay are identified by Zhang et al. (2018)

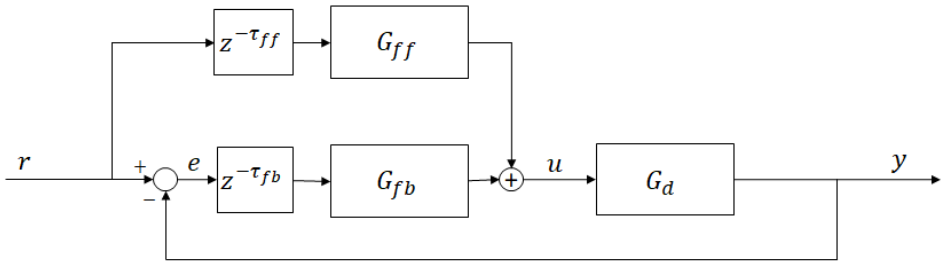


Figure 3.6: Block diagram of the model for which feedback- and feedforward subsystems and time delays are identified by Sheffler et al. (2019)

To give an overview of the different versions of the SSID algorithm and the possibilities of each, Section 3.3 highlights the most important characteristics.

Table 3.2: Overview of the different versions of the SSID algorithm.

Research	Domain	SISO/MIMO	Subsystems	Delays	# Candidate pools
Zhang & Hoagg, 2016a	s	SISO	FB and FF	-	1
Zhang & Hoagg, 2016b	z	MIMO	FB and FF (FIR)	-	1
Koushkbaghi et al., 2018	z	SISO	FB and FF (FIR and non-linear)	-	1
Zhang et al., 2018	z	SISO	FB and FF (FIR)	FB	1
Sheffler et al., 2019	z	SISO	FB and FF (FIR)	FB & FF	2 (FB and FF delay)

3.4. SSID including feedback- and feedforward time delays

The main principle of the extension of the SSID technique of Sheffler et al. (2019) is the same as that of Zhang and Hoagg (2016a) (Sections 3.1 and 3.2). The block diagram of Figure 3.6 matches that of Figure 3.2, only the time delays are added besides the subsystem blocks. This allows the functions of G_{ff} and G_{fb} to be modelled as transfer functions that can be split up in a numerator and denominator. Another difference with respect to Section 3.1 is the domain in which the functions work, as the method in Sheffler et al. (2019) works in the z-domain. Equation (3.23) shows the closed loop system equation in this domain.

$$\tilde{G}_{yr}(z) \equiv \frac{G_d(z)G_{ff}(z)z^{-\tau_{ff}} + G_d(z)G_{fb}(z)z^{-\tau_{fb}}}{1 + G_d(z)G_{fb}(z)z^{-\tau_{fb}}} \tag{3.23}$$

Regarding the frequency response data, the definition of $\{H(\omega_k)\}_{k=1}^{N}$ is the same as in Section 3.1. Furthermore, the cost function is also the same, although in this method it also depends on τ_{ff} and τ_{fb}, see Equation (3.24). Here $\sigma_k \equiv e^{j\omega_k T_s}$.

$$J(G_{ff}, \tau_{ff}, G_{fb}, \tau_{fb}) \equiv \sum_{k=1}^{N} \left| \tilde{G}_{yr}(\sigma_k) - H(\omega_k) \right|^2 \tag{3.24}$$

Similar to the basic method in Section 3.2, the numerator and denominator coefficients of the subsystems G_{ff} and G_{fb} are going to be estimated using the SSID algorithm. Let n_{ff} and d_{ff} be the degrees of the numerator and denominator of the feedforward subsystem and n_{fb} and d_{fb} the degrees for the feedback subsystem. Define $d \equiv n_{fb} + d_{fb} + 1$ and assume $G_{ff}(z)$ is FIR. This last assumption means that the FF denominator coefficients are not estimated. Also, define $\Gamma_j(z) \equiv [z^j \quad z^{j-1} \quad \cdots \quad z \quad 1]^T$. Let α again be the numerator coefficients of the feedforward subsystem and let β be the numerator- and denominator coefficients for the feedback subsystem. As mentioned before in Section 3.1.1, β and ϕ in the basic algorithm have become α and β in this version. Next, define the functions $\mathcal{N}_{ff}$, $\mathcal{N}_{fb}$ and $\mathcal{D}_{fb}$ in Equation (3.25) and $\mathcal{G}_{ff}$ and $\mathcal{G}_{fb}$ in Equation (3.26). Note that $G_d = \frac{N_d}{D_d}$. Compared to Section 3.2, $\mathcal{D}_{ff}$ is equal to 1, which is a consequence of the assumption that $G_{ff}(z)$ is FIR.

$$\begin{aligned} \mathcal{N}_{ff}(z,\alpha) &\equiv \Gamma_{n_{ff}}^T(z)\alpha, \quad \mathcal{N}_{fb}(z,\beta) \equiv \left[\Gamma_{n_{fb}}^T(z) \quad 0_{1\times d_{fb}}\right]\beta, \\ \mathcal{D}_{ff} &= 1 \quad \mathcal{D}_{fb}(z,\beta) \equiv z^{d_{fb}} + \left[0_{1\times(n_{fb}+1)} \quad \Gamma_{d_{fb}-1}^T(z)\right]\beta \end{aligned} \tag{3.25}$$

$$\mathcal{G}_{ff}(z,\alpha) \equiv \frac{\mathcal{N}_{ff}(z,\alpha)}{z^{d_{ff}}}, \quad \mathcal{G}_{fb}(z,\beta) \equiv \frac{\mathcal{N}_{fb}(z,\beta)}{\mathcal{D}_{fb}(z,\beta)} \tag{3.26}$$

Furthermore, define $\gamma \in \mathbb{Z}$ and $\psi \in \mathbb{Z}$ as the feedback- and feedforward time delay, respectively. Due to the discrete time technique they are integer multiples of the time step T_s. The product of the integer and the time step then results in the actual time delay in seconds.

In their research, Sheffler et al. (2019) only treat positive (i.e., resulting in a causal system) time delays. However, for preview tracking tasks, some models require a negative time delay for the feedforward subsystem, as the operator is able to 'look into the future'. To test whether the algorithm is able to identify negative time delays, Section 4.4.2 describes a simulated test.

Next, the full closed loop system $\tilde{\mathcal{G}}$ can be defined in terms of the parameters to be identified α, β, γ and ψ in Equation (3.27). Here, the right side of the equation has the terms of α and ψ separated from the terms depending on β and γ, which are combined in $\tilde{\mathcal{N}}_{ff}$, $\tilde{\mathcal{N}}_{fb}$ and $\tilde{\mathcal{D}}$, see Equation (3.28). Splitting β and γ from the other parameters is a consequence of the use of the feedback- and feedforward candidate pool, which is described in the next subsection.

$$\begin{aligned} \tilde{\mathcal{G}}(z,\alpha,\psi,\beta,\gamma) &\equiv \frac{z^{\gamma-d_{ff}-\psi} N_d(z)\, \mathcal{N}_{ff}(z,\alpha)\, \mathcal{D}_{fb}(z,\beta) + N_d(z)\, \mathcal{N}_{fb}(z,\beta)}{D_d(z)\, \mathcal{D}_{fb}(z,\beta)\, z^{\gamma} + N_d(z)\, \mathcal{N}_{fb}(z,\beta)} \\ &= \frac{z^{-\psi}\tilde{\mathcal{N}}_{ff}(z,\beta,\gamma)\alpha + \tilde{\mathcal{N}}_{fb}(z,\beta)}{\tilde{\mathcal{D}}(z,\beta,\gamma)} \end{aligned} \tag{3.27}$$

$$\begin{aligned} \tilde{\mathcal{N}}_{ff}(z,\beta,\gamma) &\equiv N_d(z)\, \mathcal{D}_{fb}(z,\beta)\, z^{\gamma-d_{ff}} \Gamma_{n_{ff}}^T(z), \quad \tilde{\mathcal{N}}_{fb}(z,\beta) \equiv N_d(z)\mathcal{N}_{fb}(z,\beta) \\ \tilde{\mathcal{D}}(z,\beta,\gamma) &\equiv D_d(z)\, \mathcal{D}_{fb}(z,\beta)\, z^{\gamma} + N_0(z)\, \mathcal{N}_{fb}(z,\beta) \end{aligned} \tag{3.28}$$

3.4.1. Feedback- and feedforward candidate pool

Just as in the basic algorithm, a candidate pool approach is used to identify the coefficients of the feedback- and feedforward subsystems while guaranteeing asymptotic stability. However, as the current method also needs to identify feedback- and feedforward time delays, the same approach cannot be used. Instead, the method of Sheffler et al. (2019) makes use of two candidate pools: the feedback- and the feedforward-delay candidate pool.

First, the feedback candidate pool is similar to the one described in Section 3.2.1, although now the feedback time delay γ is included in this candidate pool. Then, for each coefficient of β or γ, a range of values is set and all combinations of these coefficients form Π. To guarantee stability, the feedback

candidate pool Φ then consists of all entries of Π that also lie in the domain of $\mathcal{S}$, for which $\tilde{\mathcal{D}}$ leads to a stable closed loop system. For the discrete time system this is when the roots of $\tilde{\mathcal{D}}$ lie within the unit circle, i.e. $\mathcal{S} \equiv \left\{(\beta,\gamma) \in \mathbb{R}^d \times \mathbb{N} : \textbf{if } \widetilde{\mathcal{D}}(\lambda,\beta,\gamma) = 0, \text{ then } |\lambda| < 1\right\}$). The candidate pool Φ is then defined as $\Phi \equiv \Pi \cap \mathcal{S}$. The feedback candidate pool contains M elements ($\phi_1 \dots \phi_M$) and define $\mathcal{M} \equiv \{1, 2, \dots, M\}$. With $E_\beta \equiv [I_d \quad 0_{d\times 1}]$ and $E_\gamma \equiv [0_{1\times d} \quad 1]$, it then follows that $\beta_i = E_\beta \phi_i$ and $\gamma_i = E_\gamma \phi_i$.

The feedforward candidate pool $\Psi \subset \mathbb{Z}$ is the new addition of this extension. It consists of the possible time delays of the feedforward part, which can be either positive or negative (as a human can 'look into the future' in some tasks). Just as for the feedback candidate pool, let every entry of Ψ ψ_i be unique, i.e. if $i \neq j$, $\psi_i \neq \psi_j$ if $\psi_i, \psi_j \in \Psi$. Let Ψ have p elements and define $\mathcal{P} \equiv 1, 2, \dots, p$. The two candidate pools Φ and Ψ will then be used to identify all parameters.

3.4.2. Identification of subsystems and time delays

To identify the unknown parameters, the cost function of Equation (3.24) is being worked out as was done in Section 3.2.2. First $\tilde{\mathcal{G}}(\sigma_k, \alpha, \beta, \gamma, \psi)$ is substituted in the cost function. Working away the squared term as was done in Section 3.2.2 results in eq. (3.29).

$$\mathcal{J}(\alpha,\psi,\beta,\gamma) \equiv \alpha^T \Omega_2(\beta,\gamma)\alpha + \Omega_0(\beta,\gamma) + 2Re\mathrm{Z}^T(\psi)\Omega_1^T(\beta,\gamma)\alpha \tag{3.29}$$

Here, Ω_2, Ω_0 and Ω_1 are given by Equations (3.30) to (3.32), respectively. Υ_1 and Υ_2 are then given by Equations (3.33) and (3.34) and Equation (3.35) shows the equation for Z. The equations for Ω_0 and Ω_2 are similar to the ones in Section 3.2.2, although a term depending on ψ appears to be missing in Equation (3.30). This term has canceled out in the process of multiplying it with its conjugate transpose. Compared to the basic algorithm, Ω_1 looks quite different, however. The summation is removed and replaced by a matrix multiplication. This difference will be explained next.

$$\Omega_2(\beta,\gamma) \equiv \mathrm{Re}\sum_{k=1}^{N}\left[\frac{\tilde{\mathcal{N}}_{ff}(\sigma_k,\beta,\gamma)}{\tilde{\mathcal{D}}(\sigma_k,\beta,\gamma)}\right]^*\left[\frac{\tilde{\mathcal{N}}_{ff}(j\omega_k,\beta,\gamma)}{\tilde{\mathcal{D}}(\sigma_k,\beta,\gamma)}\right] \in \mathbb{R}^{\left(n_{ff}+1\right)\times\left(n_{ff}+1\right)} \tag{3.30}$$

$$\Omega_0(\beta,\gamma) \equiv \sum_{k=1}^{N}\left|\frac{\tilde{\mathcal{N}}_{fb}(j\sigma_k,\beta,\gamma)}{\tilde{\mathcal{D}}(j\sigma_k,\beta,\gamma)} - H(\omega_k)\right|^2 \in \mathbb{R} \tag{3.31}$$

$$\Omega_1(\beta,\gamma) \equiv \Upsilon_2^{\mathrm{T}}(\beta,\gamma)\left(\mathrm{diag}\,\Upsilon_1(\beta,\gamma)\right)^* \in \mathbb{C}^{(d_{ff}+1)\times N} \tag{3.32}$$

$$\Upsilon_1(\beta,\gamma) \equiv \left[\frac{\tilde{\mathcal{N}}_{ff}(\sigma_1,\beta,\gamma)}{\tilde{\mathcal{D}}(\sigma_1,\beta,\gamma)} \quad \cdots \quad \frac{\tilde{\mathcal{N}}_{ff}(\sigma_N,\beta,\gamma)}{\tilde{\mathcal{D}}(\sigma_N,\beta,\gamma)}\right]^{\mathrm{T}} \in \mathbb{C}^N \tag{3.33}$$

$$\Upsilon_2(\beta,\gamma) \equiv \left[\frac{\tilde{\mathcal{N}}_{fb}(\sigma_1,\beta,\gamma)}{\tilde{\mathcal{D}}(\sigma_1,\beta,\gamma)} - H(\omega_1) \quad \cdots \quad \frac{\tilde{\mathcal{N}}_{fb}(\sigma_N,\beta,\gamma)}{\tilde{\mathcal{D}}(\sigma_N,\beta,\gamma)} - H(\omega_N)\right]^{\mathrm{T}} \in \mathbb{C}^{N\times(d_{ff}+1)} \tag{3.34}$$

$$\mathrm{Z}(\psi) \equiv \left[\sigma_1^{-\psi} \cdots \sigma_N^{-\psi}\right]^{\mathrm{T}} \in \mathbb{C}^N \tag{3.35}$$

The derivation of getting to the term with Ω_1 is provided by Equation (3.36). The initial form is retrieved by only looking at the term dependent on a first order α, substituting Equation (3.27) into Equation (3.24) and working out the squared term using Equation (3.12), as was done in Section 3.2.2. This term, including all components exact for α itself, is called $\hat{\Omega}_1$. After working away the summation term and performing some matrix operations with transposes, the expression can be expressed in terms of Ω_1 and Z, thus splitting up the terms dependent on the feedback- and feedforward candidate pool, respectively.

$$\begin{aligned}\hat{\Omega}_1 &= 2\sum_{k=1}^{N} Re\left(\frac{\tilde{\mathcal{N}}_{fb}\left(j\sigma_k,\beta,\gamma\right)}{\tilde{\mathcal{D}}\left(j\sigma_k,\beta,\gamma\right)} - H\left(\omega_k\right)\right)^{*} \sigma_k^{-\psi}\frac{\tilde{\mathcal{N}}_{ff}\left(\sigma_k,\beta,\gamma\right)}{\tilde{\mathcal{D}}\left(\sigma_k,\beta,\gamma\right)} \\ &= 2Re\Upsilon_1^{*}(\beta,\gamma)\left(\operatorname{diag}\mathrm{Z}(\psi)\right)\Upsilon_2(\beta,\gamma) \\ &= 2Re\left(\left(\operatorname{diag}\Upsilon_1(\beta,\gamma)\right)^{*}\mathrm{Z}(\psi)\right)^{T}\Upsilon_2(\beta,\gamma) \\ &= 2Re\mathrm{Z}^{T}(\psi)\left(\left(\operatorname{diag}\Upsilon_1\left(\beta,\gamma\right)\right)^{*}\right)^{T}\Upsilon_2(\beta,\gamma) \\ &= 2Re\mathrm{Z}^{T}(\psi)\left(\Upsilon_2^{T}(\beta,\gamma)\operatorname{diag}\Upsilon_1^{*}(\beta,\gamma)\right)^{T} \\ &= 2Re\mathrm{Z}^{T}(\psi)\Omega_1^{T}(\beta,\gamma)\end{aligned} \tag{3.36}$$

The cost function in Equation (3.29) can then be written in terms of only α and ψ by inserting the terms ϕ_i from the candidate pool, see Equation (3.37). For all $\psi \in \mathbb{N}$, this equation is convex in α and if the number of frequency points N is sufficiently large, Ω_2 is positive definite. Then, just as in Section 3.2.2, a unique global minimizer for α_i can be found, now also dependent on ψ. Equation (3.38)

$$\begin{aligned}\mathcal{J}_i(\alpha,\psi) \equiv& \mathcal{J}\left(\alpha,\psi,E_\beta\phi_i,E_\gamma\phi_i\right) \\ =& \alpha^{\mathrm{T}}\Omega_2\left(E_\beta\phi_i,E_\gamma\phi_i\right)\alpha + \Omega_0\left(E_\beta\phi_i,E_\gamma\phi_i\right) + \operatorname{Re}\Gamma^{\mathrm{T}}(\psi)\Omega_1^{\mathrm{T}}\left(E_\beta\phi_i,E_\gamma\phi_i\right)\alpha\end{aligned} \tag{3.37}$$

$$\alpha_i(\psi) \equiv -\frac{1}{2}\Omega_2^{-1}\left(E_\beta\phi_i,E_\gamma\phi_i\right)\operatorname{Re}\Omega_1\left(E_\beta\phi_i,E_\gamma\phi_i\right)\Gamma(\psi) \tag{3.38}$$

Substituting Equation (3.38) in Equation (3.37) results in the auxiliary cost function $\mathcal{Q}_i$ that only depends on ψ. Equation (3.39) gives the equation, where $\mathcal{F}_i$ is given by Equation (3.40). In Equation (3.39), the first and third terms of Equation (3.37) (the terms with Ω_2 and Ω_1) have been merged and the equation is written in such a way that $\mathcal{F}_i$ is only dependent on ϕ_i, making the algorithm significantly faster. The vectors containing the real and imaginary parts of ψ and Ω_1 enabled this by replacing the Re operator in Equation (3.37).

$$\begin{aligned}\mathcal{Q}_i(\psi) &\equiv \mathcal{J}_i\left(\alpha_i(\psi),\psi\right) \\ &= \Omega_0\left(E_\beta\phi_i,E_\gamma\phi_i\right) - \begin{bmatrix}\operatorname{Re}\Gamma(\psi)\\ -\operatorname{Im}\Gamma(\psi)\end{bmatrix}^{\mathrm{T}}\mathcal{F}_i\begin{bmatrix}\operatorname{Re}\Gamma(\psi)\\ -\operatorname{Im}\Gamma(\psi)\end{bmatrix}\end{aligned} \tag{3.39}$$

$$\mathcal{F}_i \equiv \frac{1}{4}\begin{bmatrix}\operatorname{Re}\Omega_1^{\mathrm{T}}\left(E_\beta\phi_i,E_\gamma\phi_i\right)\\ \operatorname{Im}\Omega_1^{\mathrm{T}}\left(E_\beta\phi_i,E_\gamma\phi_i\right)\end{bmatrix}\Omega_2^{-1}\left(E_\beta\phi_i,E_\gamma\phi_i\right)\begin{bmatrix}\operatorname{Re}\Omega_1^{\mathrm{T}}\left(E_\beta\phi_i,E_\gamma\phi_i\right)\\ \operatorname{Im}\Omega_1^{\mathrm{T}}\left(E_\beta\phi_i,E_\gamma\phi_i\right)\end{bmatrix}^{\mathrm{T}} \tag{3.40}$$

To identify the parameters, the feedforward-delay candidate pool $\Psi(=\psi_1,\ldots,\psi_p)$ is then used. For each ϕ_i, calculate the minimum value of $\mathcal{Q}_i$, i.e. find ψ_{q_i} with $\mathrm{Q}_i\left(\psi_{q_i}\right) = \min_{j\in\mathcal{P}}\mathrm{Q}_i\left(\psi_j\right)$. Then, find the minimum value of $\mathcal{Q}_i$ for all possible ϕ_i with $\mathrm{Q}_\ell\left(\psi_{q_\ell}\right) = \min_{i\in\mathcal{M}}\mathrm{Q}_i\left(\psi_{q_i}\right)$. Then the identified parameters are $\alpha^{+} = \alpha_\ell(\psi_{q_\ell})$, $\tau_{ff} = \psi^{+} = \psi_{q_\ell}$, $\phi^{+} = E_\beta\phi_\ell$ and $\gamma^{+} = E_\gamma\phi_\ell$.

3.5. Summary of SSID

Subsystem identification is thus a system identification technique that is used to identify multiple control responses based on a single closed-loop input-output frequency response function (FRF). Of these responses, or subsystems, one should be a feedforward subsystem, while the other subsystem is in the closed-loop. To guarantee asymptotic stability, a candidate pool approach is used. Different versions of the algorithm have been discussed in Section 3.2 and Section 3.4. However, the overall strategy remains the same, as is summarized by Figure 3.7.

The figure shows the three main steps of the algorithm: forming candidate pools, calculating the remaining parameter values and the cost value, and finally find the optimal cost function value and the corresponding parameters. Note that depending on the version of the algorithm, time delays can be identified and an additional candidate pool might be required. Step 1 might thus consist of 1 of 2 candidate pools that are formed, but especially step 2 becomes larger when the last version of the algorithm developed by Sheffler et al. (2019) is used. In this case, a second loop is used to first find the optimal

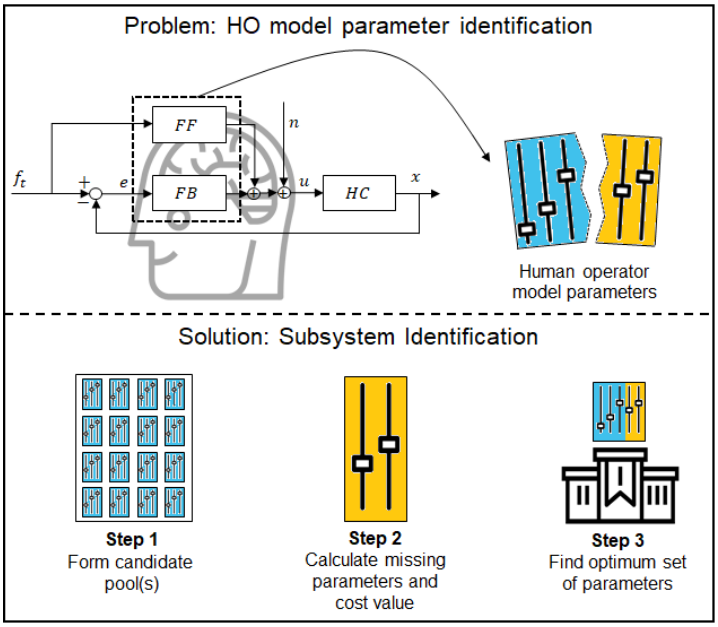


Figure 3.7: Schematic explaining the working of the subsystem identification method.

value of the feedforward time delay, which is then used to calculate the remaining feedforward numerator coefficients.

3.6. Synthesis

Based on the study performed regarding subsystem identification, a number of conclusions can be drawn regarding this research project. Firstly, the SSID algorithm works in the frequency domain and uses the closed-loop frequency response data. This makes it related to the multi-loop Fourier Coefficients (FC) method discussed in Section 2.3.1. With respect to this technique, the big advantage of SSID is that it only requires a single forcing function. With equal performance, this would remove an important constraint for estimating the sub-responses in multi-loop systems.

Over the past couple of years, different versions of the SSID algorithm have been published. While initial versions were only able to identify subsystems without corresponding delay terms (Zhang and Hoagg, 2016a,b), newer versions have shown to be able to identify feedback time delays (Zhang et al., 2018) and later on even both feedback- and feedforward time delays (Sheffler et al., 2019). This last version of the algorithm is especially relevant, as it allows for identification of realistic HO pursuit- and preview models. It is therefore selected to be used in the remainder of this research project.

A final comment regarding the work on SSID involves the noise insertion that is used. Throughout the different publications on SSID, noise has been applied at different locations without a consistent idea behind it. Initial publications had multiple noise input sources, whereas later ones did not contain any. Furthermore, the noise inserted at the output (as seen in Figures 3.2 and 3.3), is not realistic when considering manual control tracking tasks.

4

Application of SSID in examples

Regarding the subsystem identification (SSID) technique discussed in the previous chapter, the initial research by Zhang and Hoagg (2016a,b) focused on the explanation of the method, whereas later publications of Zhang et al. (2018) and Sheffler et al. (2019) provided more relevant methods for identification of realistic human controllers. To demonstrate the technique's ability to successfully identify the parameters of the subsystems, the initial two papers contain examples. Consequently, the first part of this chapter will focus on reproducing four of these examples and will elaborate on the results. This is only done for the examples described by Zhang and Hoagg (2016a) as Section 3.3 already mentioned that the work by Zhang and Hoagg (2016b) is less relevant for this research. After discussing the examples from the basic algorithm, an example using the extension by Sheffler et al. (2019) is provided to verify the original and a slightly modified version of the method.

4.1. Basic algorithm: general information

The information provided in this section will be applicable to all four examples of the basic algorithm. First, the functions G_0, G_1 and G_2 (see Figure 4.1, which is copied from Section 3.1) are given by Equation (4.1). This implies that $\beta_* = [2.1 \quad 3]^T$ and $\phi_* = [6.5 \quad 5.4 \quad 7.1]^T$ are the true parameters. Furthermore, let $N = 20$ and $\omega_k = 0.2\pi k$, where $k \in \{1 \dots N\}$. Just as in Section 3.1, N does not refer to the number of samples of a time domain sequence used for the Discrete Fourier Transform (DFT). It just refers to the number of frequencies that will be evaluated.

$$G_0(s) = \frac{4}{s+2}, \quad G_1(s) = \frac{2.1s+3}{s+6.5}, \quad G_2(s) = \frac{5.4}{s+7.1} \tag{4.1}$$

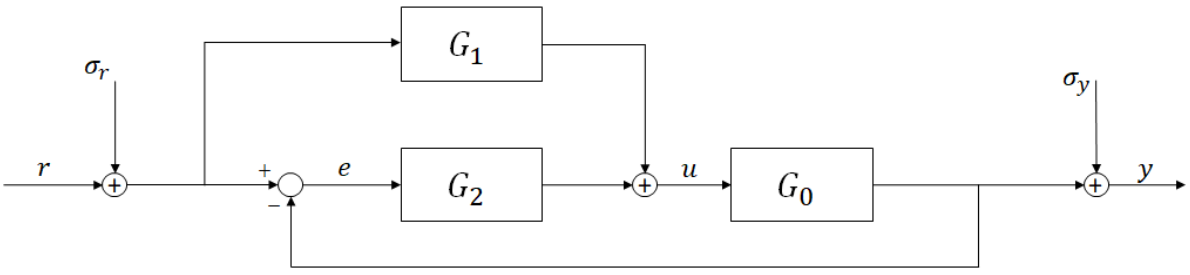


Figure 4.1: Block diagram of the ET system with sign conventions according to Zhang and Hoagg (2016a).

4.1.1. Candidate pools

When constructing the candidate pools, the space $\mathcal{R} = [-8, 8] \times [-8, 8] \times [-8, 8]$, is used. This is a compact set containing ϕ_*. It can be shown that within this range of possibilities, only β_* and ϕ_* result in

the closed loop system. Then, for the case where ϕ_* is in the candidate pool Φ, Equation (4.2) provides the set of all combinations Π_0. Here, the step size (0.1) and the number of steps (160) are selected arbitrarily. As described in Section 3.2.1, the candidate pool is then defined as $\Phi_0 \equiv \Pi_0 \cap \mathcal{S}$, i.e. the combinations of Π_0 that guarantee $\tilde{G}$ (Equation (3.7)) to be stable (or for $\tilde{\mathcal{D}}(s,\phi)$ to have the roots in the left-half plane).

$$\Pi_0 \equiv \left\{\phi \in \mathbb{R}^3 : E_1\phi, E_2\phi, E_3\phi \in \{-8 + 0.1k\}_{k=0}^{160}\right\} \subseteq \mathcal{R} \tag{4.2}$$

On the other hand, the candidate pools for which ϕ_* does not lie in the candidate pool are to be constructed. Here, the question arises how large the offset of the identified parameters ϕ^+ is from the true parameters ϕ_*. Therefore, multiple candidate pools with different densities are formed. For $j = 1, \ldots, 20$ Equation (4.3) gives the equation for the set of combinations Π_j. Note that for all options, ϕ_* is not located in Π_j. Then, the candidate pools Φ_j are defined as $\Phi_j \equiv \Pi_j \cap \mathcal{S}_\rho$ where $\rho = -0.001$ (see Section 3.2.1 for explanation).

$$\Pi_j \equiv \left\{\phi \in \mathbb{R}^3 : E_1\phi, E_2\phi, E_3\phi \in \left\{-8 + \frac{16}{5 + 10(j-1)}k\right\}_{k=0}^{5+10(j-1)}\right\} \subseteq \Psi \tag{4.3}$$

4.2. Basic algorithm: noisy frequency response data

As the transfer functions are all known in the examples, noise needs to be added in order to simulate noisy frequency response data. When looking at the block diagram of Figure 4.1 used in (Zhang and Hoagg, 2016a), the noise is added to the system at the target signal r (or f_t in other diagrams) and at the system output y (or x) in the form of σ_r and σ_y, respectively. For the examples of the paper, these noise terms are the Fourier Transforms of Gaussian white-noise realizations with a variance of $\sigma^2_{\sigma_{r,i}} = \sigma^2_{\sigma_{y,i}} = 4^{-i}$, for $i = 1, \ldots, 20$. These terms, called $\sigma_{r,i}$ and $\sigma_{y,i}$, are used to form the closed loop frequency response data according to Equations (4.4) and (4.5). Here, $\tilde{G}(j\omega_k)$ is the linear closed loop transfer function derived from Equation (4.1) and fig. 4.1 and $r(j\omega_k)$ is the Fourier Transform of the input signal. Regarding this last signal, it is important to note that no values are defined in the work of Zhang and Hoagg (2016a).

$$H_i(\omega_k) \equiv \tilde{G}(j\omega_k) + \sigma_i(j\omega_k) \tag{4.4}$$

$$\sigma_i(j\omega_k) \equiv \left[\tilde{G}(j\omega_k)\,\sigma_{r,i}(j\omega_k) + \sigma_{y,i}(j\omega_k)\right]/r(j\omega_k) \tag{4.5}$$

The paper then defines the noise-to-signal ratio according to Equation (4.6), where more specifically $R_1 = 23.8, R_5 = 1.39, R_{10} = 4.49 \times 10^{-2}, R_{12} = 9.53 \times 10^{-3}$, and $R_{20} = 4.36 \times 10^{-5}$. As an explicit procedure for calculating the noise realizations and the resulting noise-to-signal ratios is lacking, two possibilities are discussed in the next subsection. Consequently, an attempt is made to relate the noise levels in the examples to the conventionally applied human remnant levels.

$$R_i \equiv \frac{1}{N}\sum_{k=1}^{N}\left|\frac{\sigma_i(j\omega_k)}{\tilde{G}(j\omega_k)}\right| \tag{4.6}$$

4.2.1. Generating noise realizations

As mentioned earlier in this chapter, the noise signals $\sigma_{r,i}$ and $\sigma_{y,i}$ and the corresponding noise-to-signal ratio R_i can be computed in two different ways. Note that in this process an attempt need to be made on $r(j\omega)$ as well (as nothing is specified). Note that as the values for R_i as used by Zhang and Hoagg (2016a) are appears to be computed using a random seed, they can only be approximated.

For the noise signals, the first option is to apply the Fast Fourier Transform (FFT) to a random time signal. If this is a time signal of $T = 10$ seconds, sampled at $f_s = 4.2$ Hz, the resulting frequencies match the frequencies that are to be evaluated, namely $\{0.2\pi \ldots 4\pi\}$ rad/s or $\{0.1 \ldots 2\}$ Hz. Multiplying the

time signal with the with the standard deviations (corresponding to the variance of 4^{-i}) before applying the Fourier Transform then incorporates the noise intensities properly.

Although this method creates the noise signal from a time domain signal, it represents a single noise realization. As the seed is not provided in the paper, a default seed is used for generating $\sigma_{r,i}$ and $\sigma_{y,i}$. Assuming that $r\,(j\omega_k)$ is constant at all frequencies, allows the calculation of R_i. $r\,(j\omega_k)$ can then be adjusted such that the calculated values of R_1, R_5, R_{10}, R_{12} and R_{20} approximate the values provided by Zhang and Hoagg (2016a) (and mentioned slightly before). Note again that this depends on the noise realization. With these settings, the average value of $r(j\omega)$ is determined to be 0.18 at all frequencies ω_k. This value implies that a $0°$ phase angle is assumed.

Another way to calculate $\sigma_{r,i}$ and $\sigma_{y,i}$, is to assume an analytical white noise sequence. Then, the power is constant over all frequencies and thus the absolute value of the noise, $\|\sigma(j\omega)\|$, as well. The disadvantage here is that the noise added to the frequency response data (using Equations (4.4) and (4.5) is an absolute value and not a complex number. The impact of this difference is evaluated in Section 4.3.3.

4.2.2. Relating noise levels to the human remnant

With respect to established models of the human remnant (Levison et al., 1969; van der El et al., 2019), the noise addition at r and y applied by Zhang and Hoagg (2016a) is unconventional. Normally, the noise is added as colored white noise inside the control loop. This implies that the human generates the noise, which is realistic as all its non-linear components are modeled as colored white noise. It is therefore counter-intuitive to see noise inserted at the system output y, which is outside of the human control loop. Generally, the noise levels are expressed in terms of the fraction of the generated control output signal u that is caused by the noise (or human remnant) as compared to the total signal. This effect is illustrated by the ratio of $\frac{\sigma_{u_n}^2}{\sigma_u^2}$ (or $\frac{\sigma_{u_{\sigma_r}}^2}{\sigma_u^2}$ for this notation), which is defined by Equation (4.7). Here, $S_{uu}(j\omega)$, $S_{uu_n}(j\omega)$, $S_{uu_r}(j\omega)$ and $S_{uu_{\sigma_r}}(j\omega)$ are the power spectral densities (PSDs) of the control output signal and the parts of this PSD caused by the remnant n, the input signal r and the noise inserted at the input σ_r. Note that in this equation the contribution of σ_y to the noise is not present, as this noise component does not contribute to the control output signal u.

$$\begin{aligned}\frac{\sigma_{u_n}^2}{\sigma_u^2} = \frac{\int_0^\infty S_{uu_n}(j\omega)d\omega}{\int_0^\infty S_{uu}(j\omega)d\omega} &= \frac{\int_0^\infty S_{uu_{\sigma_r}}(j\omega)d\omega}{\int_0^\infty S_{uu_{\sigma_r}}(j\omega) + S_{uu_r}(j\omega)d\omega} \\ &= \frac{\int_0^\infty \|G_{ur}\|^2 S_{\sigma_r\sigma_r}(j\omega)d\omega}{\int_0^\infty \|G_{ur}\|^2 \left(S_{\sigma_r\sigma_r}(j\omega) + S_{rr}(j\omega)\right)d\omega} \\ &= \frac{\int_0^\infty \|G_{ur}\|^2 N_r}{\int_0^\infty \|G_{ur}\|^2 \left(N_r + S_{rr}(j\omega)\right)d\omega}\end{aligned} \tag{4.7}$$

The equation shown above calculates the relative impact of the noise on the control output signal by integrating the (continuous time) PSDs of the output signal caused by r and σ_r. These PSDs can be estimated from the Fourier Transform of the signal through the periodogram, i.e., $S_{uu} = \frac{1}{N_{samples} f_s}\left|U(j\omega)\right|^2$, where $N_{samples}$ is the number of samples of the time sequence used for the DFT and f_s is the sampling frequency. For this example, they were equal to 42 and 4.2 (as mentioned in Section 4.2.1), respectively. However, considering the fact that G_{ur} cancels out, the equations can be further reduced to Equation (4.8). Here, the values of $\sigma_{\sigma_r}^2$ are simply the variances provided by Zhang and Hoagg (2016a), equal to $4^{-}i$.

$$\begin{aligned}\frac{\sigma^2_{u_{\sigma_r}}}{\sigma^2_u} &= \frac{\int_0^\infty \|G_{ur}\|^2 N_r}{\int_0^\infty \|G_{ur}\|^2 \left(N_r + S_{rr}(j\omega)\right) d\omega} \\ &= \frac{\int_0^\infty N_r}{\int_0^\infty \left(N_r + S_{rr}(j\omega)\right) d\omega} \\ &= \frac{\sigma^2_{\sigma_r}}{\sigma^2_{\sigma_r} + \int_0^\infty S_{rr}(j\omega) d\omega}\end{aligned} \tag{4.8}$$

While the variance of the noise is provided for each intensity, the integrated PSD of the target signal $S_{rr}(j\omega)$ needs to be calculated. This can be done with the value of 0.18 for $r(j\omega_k)$ that was used to match the values of R_i, as mentioned in Section 4.2.1. As it is assumed that this value is used at all frequencies, the integral is computed using Equation (4.9), where T is to 10s, corresponding to the DFT that uses 42 samples at 4.2 Hz.

$$\int_0^\infty S_{rr}(j\omega) d\omega = \frac{1}{N_{samples} T f_s} \sum_{k=1}^{N_{samples}} r_k^*(j\omega) r_k(j\omega) \tag{4.9}$$

For each noise intensity, the ratio $\frac{\sigma^2_{u_n}}{\sigma^2_u}$ can then be computed. Table 4.1 shows this ratio and the original and reproduced R_i for each noise intensity. From research performed by van der El et al. (2019), the value of $\frac{\sigma^2_{u_n}}{\sigma^2_u}$ (which is for this example thus $\frac{\sigma^2_{u_{\sigma_r}}}{\sigma^2_u}$) is generally between 0.1-0.65 (depending on the controlled element dynamics and the forcing function bandwidth), which would correspond to the noise levels corresponding to $i = 4, 5, 6$.

It should however be noted that, despite this effort to relate the noise levels of this example to noise levels from other literature, there are a number of assumptions that complicate this. First, in the work by van der El et al. (2020), the tracking tasks involve multi-sine signals that contain a large signal-to-noise-ratio at the frequencies used for parameter estimation. The ratio $\frac{\sigma^2_{u_n}}{\sigma^2_u}$, however, represents an average noise level over all frequencies. As in this example a target signal with an equal power at all frequencies is assumed, it follows that the 'effective' signal-to-noise ratio (or the signal-to-noise ratio at the relevant target frequencies) in this example is relatively low. It is therefore expected that the level of R_i that corresponds to realistic human remnant levels is lower than the previously mentioned $i = 4 - 6$. Another aspect that limits the comparability between the noise-to-signal ratio R_i and the ratio $\frac{\sigma^2_{u_n}}{\sigma^2_u}$ from literature, is the fact that the R_i is also affected by the noise input at the system output y. As this noise component cannot influence the control output u, as it is not fed back into the system, the comparison between R_i and $\frac{\sigma^2_{u_n}}{\sigma^2_u}$ is even more difficult. Therefore, from the examples treated next, only indicative conclusions can be drawn regarding the performance of SSID with respect to realistic noise levels.

Finally, to provide an indication of how the noise levels of the treated examples are developing over the frequencies, Figure 4.2 shows the PSDs of $S_{uu_{\sigma_r}}$ and $S_{\sigma_r\sigma_r}$ for different values of i and the PSD of S_{uu_r}. The noise PSDs are computed using an average of 1000 noise realizations. The figure shows how the noise spectrum of $S_{\sigma_r\sigma_r}$ is indeed white and how S_{uu_r} lies between $S_{uu_{\sigma_r}}$ of $i = 4$ and $i = 5$, which corresponds to Table 4.1. The shape of $S_{uu_{\sigma_r}}$ and $S_{\sigma_r\sigma_r}$ is caused by $\|G_{ur}\|^2$ (see Equation (4.7)), for which G_{ur} is given by Equation (4.10)

$$G_{ur} = \frac{2.1s^3 + 27.51s^2 + 103s + 112.8}{s^3 + 15.6s^2 + 94.95s + 232.7} \tag{4.10}$$

4.3. Basic algorithm: examples and results

This section will elaborate on the last 4 examples found in (Zhang and Hoagg, 2016a) (Examples 2-5). These show how the SSID technique is able to identify the parameters when the true parameters ϕ_* are or are not in the candidate pool, for both noiseless and noisy frequency data. Here, the density of

Table 4.1: Values of the noise-to signal ratio R_i given by Zhang and Hoagg (2016a) and from a reproduction, and the relative contribution of σ_r to the control output u, $\frac{\sigma_{u\sigma_r}^2}{\sigma_u^2}$, for different noise intensities ($var(\sigma_r) = 4^{-i}$).

i	R_i (Original)	R_i (Reproduced)	$\frac{\sigma_{u\sigma_r}^2}{\sigma_u^2}$
1	$2.38 \cdot 10^{1}$	$2.22 \cdot 10^{1}$	$9.95 \cdot 10^{-1}$
2		$1.11 \cdot 10^{1}$	$9.80 \cdot 10^{-1}$
3		$5.56 \cdot 10^{0}$	$9.25 \cdot 10^{-1}$
4		$2.78 \cdot 10^{0}$	$7.56 \cdot 10^{-1}$
5	$1.39 \cdot 10^{0}$	$1.39 \cdot 10^{0}$	$4.38 \cdot 10^{-1}$
6		$6.95 \cdot 10^{-1}$	$1.63 \cdot 10^{-1}$
7		$3.48 \cdot 10^{-1}$	$4.62 \cdot 10^{-2}$
8		$1.74 \cdot 10^{-1}$	$1.20 \cdot 10^{-2}$
9		$8.69 \cdot 10^{-2}$	$3.02 \cdot 10^{-3}$
10	$4.49 \cdot 10^{-2}$	$4.35 \cdot 10^{-2}$	$7.57 \cdot 10^{-4}$
11		$2.17 \cdot 10^{-2}$	$1.89 \cdot 10^{-4}$
12	$9.53 \cdot 10^{-3}$	$1.09 \cdot 10^{-2}$	$4.73 \cdot 10^{-5}$
13		$5.43 \cdot 10^{-3}$	$1.18 \cdot 10^{-5}$
14		$2.72 \cdot 10^{-3}$	$2.96 \cdot 10^{-6}$
15		$1.36 \cdot 10^{-3}$	$7.39 \cdot 10^{-7}$
16		$6.79 \cdot 10^{-4}$	$1.85 \cdot 10^{-7}$
17		$3.39 \cdot 10^{-4}$	$4.62 \cdot 10^{-8}$
18		$1.70 \cdot 10^{-4}$	$1.16 \cdot 10^{-8}$
19		$8.50 \cdot 10^{-5}$	$2.89 \cdot 10^{-9}$
20	$4.36 \cdot 10^{-5}$	$4.20 \cdot 10^{-5}$	$7.22 \cdot 10^{-10}$

the candidate pool and the noise intensity will be adjusted to show the sensitivity of the algorithm to these parameters.

4.3.1. Example 1: Noiseless data and true parameters in the candidate pool

The first example treated the simplest case in which the true solution was present in the candidate pool for noiseless frequency response data. Therefore, the candidate pool Φ_0 is used as it contains ϕ_*. Running the algorithm as described in Section 3.2, the identified parameters are $\beta^+ = \beta_* = [2.1 \quad 3]^T$ and $\phi^+ = \phi_* = [6.5 \quad 5.4 \quad 7.1]^T$.

4.3.2. Example 2: Noiseless data and true parameters outside the candidate pool

Example 2 extends the difficulty by using the candidate pools Φ_j, which do not contain ϕ_*. As the true parameters β_* and ϕ_* are not in the candidate pools, they will never be found. Figure 4.3 shows the difference between the identified parameters and the true parameters for different candidate pool densities, both taken from Zhang and Hoagg (2016a) (Figure 4.3a) and reproduced (Figure 4.3b). The figures, which appear to be identical, show that the identified parameters approach the true parameters with increasing candidate pool density. However, the total difference between the identified- and the true parameters does not decrease with every increase of candidate pool density. The explanation for this is that an increased density might lead to a particular entry of the candidate pool being further removed from the true value, as the spacing is changed. To get an idea of the difference between the identified- and true transfer functions for different levels of candidate pool density, Figure 4.4 shows the bode plots of the true feedforward transfer function and of the transfer function identified for $j = 1, 2$ and 12.

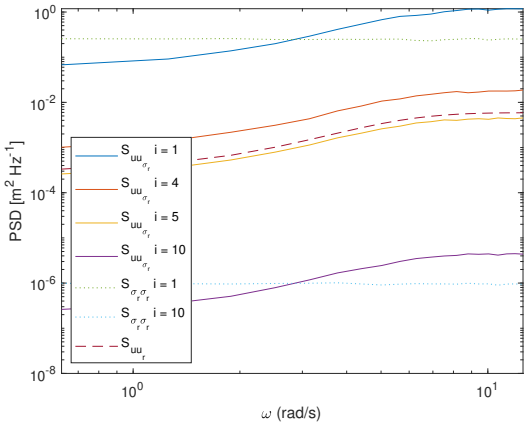
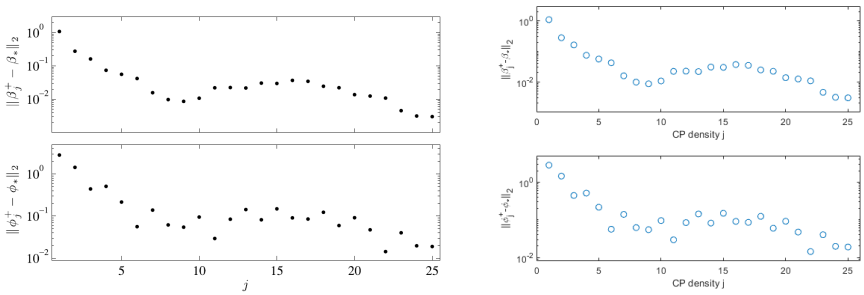


Figure 4.2: PSDs of $S_{uu_{\sigma_r}}$ and $S_{\sigma_r\sigma_r}$ for different values of the noise intensity index i and the PSD of S_{uu_r}.



(a) (b)

Figure 4.3: Difference between the identified parameters ϕ^+ and β^+ and the true parameters ϕ_* and β_* for different candidate pool densities, for noiseless frequency response data. The figure shows the original plots from (Zhang and Hoagg, 2016a) (a) and reproduced plots (b).

4.3.3. Example 3: Noisy data and true parameters in the candidate pool

The next example uses the noisy frequency data sequences $H_i(\omega_k)$ as discussed in Section 4.2. Furthermore, it uses the candidate pool Φ_0, which does contain the true parameters. The SSID algorithm is applied to each set of frequency response data to show how the method is able to identify the parameters at different noise levels. Figure 4.5 shows this by displaying the difference between the true- and the identified parameters.

Just as in the previous example, the results obtained by Zhang and Hoagg (2016a) (Figure 4.5a) are compared to the reproduced results (Figure 4.5b). Here, the reproduced plots are produced using noise sequences generated in different ways: either from the analytical noise or by Fourier Transforming a white noise time sequences. Both these methods were discussed in Section 4.2.1. Between the two different options, little differences are observed. The only obvious difference is that the noise formed using different noise realizations causes more varying results. This is a logical consequence as the noise can influence the actually estimated parameters both favourably or unfavourably.

Although the exact data points of the two plots are slightly different, the general trend corresponds well. This observation can be explained when realizing that the different noise realizations result in slightly different identified parameters. Another observation made from the plot is that after i = 14, the difference between ϕ^+ and ϕ_* is equal to 0, thus $\phi^+ = \phi_*$. However, this does not hold for β^+ and β_*. This is explained by the calculation of β^+ using Equation (3.19), which is a function of the noisy frequency response data $H_i(\omega_k)$.

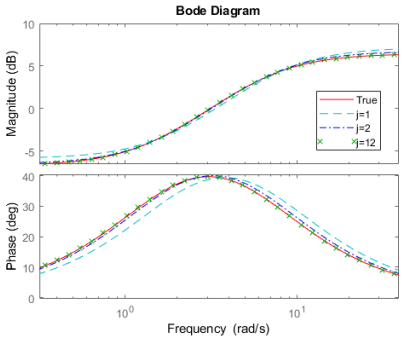
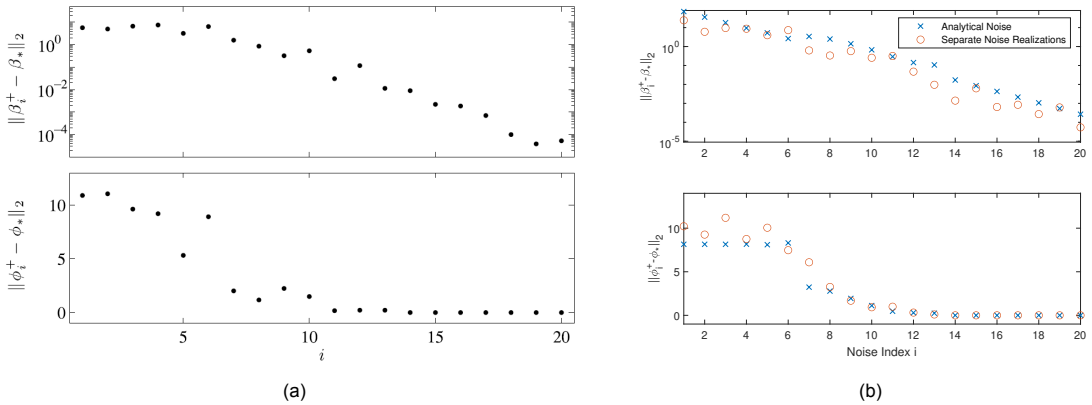


Figure 4.4: Bode plots of the true- and of the identified feedforward transfer functions for j = 1,2 and 12.



(a) (b)

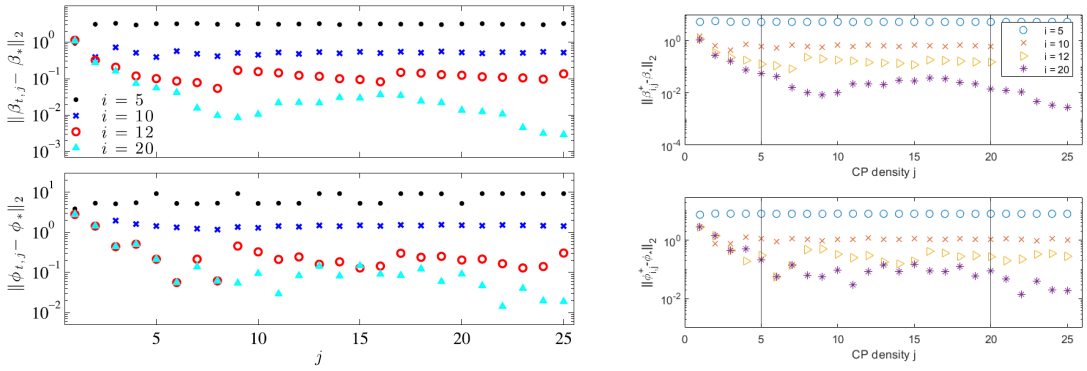
Figure 4.5: Difference between the identified parameters ϕ^+ and β^+ and the true parameters ϕ_* and β_* for different noise intensities, using the candidate pool Φ_0. The figure shows the original plots from Zhang and Hoagg (2016a) (a) and reproduced plots (b). For the reproduced values, the noise is derived in two ways: both from the analytical noise and from a separate noise realizations.

4.3.4. Example 4: Noisy data and true parameters outside the candidate pool

The last example treated by Zhang and Hoagg (2016a) evaluates the performance of SSID for both noisy frequency response data using candidate pools in which the true parameters are not located in the candidate pool. During real-life experiments, this is the case that will be treated. Figure 4.6 shows the offset between the identified- and the true parameters against the candidate pool densities, for different noise intensities. Just as in the previous examples, the results from Zhang and Hoagg (2016a) (Figure 4.6a) are compared to the reproduced results (Figure 4.6b). For the noise intensities, the noise intensities for $i = 5, 10, 12$ and 20 are given. These are the same intensities as for which the noise-to-signal ratio was provided (see Section 4.2).

The reproduced results correspond well with the results published in the original work. The error between the identified- and the true parameters converges to 0 when the candidate pool is sufficiently dense and the noise is sufficiently small. On the other hand, the plots also show that if the noise is too large, the true parameters will not be found, no matter the candidate pool density.

To get a better idea of the actual values of the identified parameters, Table 4.2 shows the values for all noise intensities for the candidate pool densities of 5 and 20 (as indicated by the vertical lines in Figure 4.6b. The table shows that, independent of candidate pool density, the identified parameters start approximating the true parameters around $R_i = 4.49 \times 10^{-2}$, or $i = 10$. When relating this to the realistic noise models as was done in Section 4.2.2, it appears as if this is too low of a noise-to-signal ratio for the technique to work in real life. After all, the comparison showed that realistic noise-to-signal ratios are found around $i = 4$ until $i = 6$. However, it was also concluded that a number of assumptions restrict making clear conclusions on the performance of SSID. Therefore, it is recommended to further investigate the performance of SSID with respect to realistic human noise levels.



(a) (b)

Figure 4.6: Difference between the identified parameters ϕ^+ and β^+ and the true parameters ϕ_* and β_* for different candidate pool densities, for the four noise intensities $i = 5, 10, 12\&20$. The figure shows the original plots from Zhang and Hoagg (2016a) (a) and reproduced plots (b). The vertical lines in (b) indicate the highlighted conditions that are discussed in more detail.

Table 4.2: Values of the identified parameters β^+ and ϕ^+ for different noise intensities and two candidate pool densities.

	CP density 5 (step size 0.3556)					CP density 20 (step size 0.0821)				
	β		ϕ			β		ϕ		
i	β_1	β_2	ϕ_1	ϕ_2	ϕ_3	β_1	β_2	ϕ_1	ϕ_2	ϕ_3
5 ($R_5 = 1.39$)	7.19	4.06	8.00	-1.24	2.67	7.16	4.36	8.00	-1.11	2.50
10 ($R_{10} = 4.49 \times 10^{-2}$)	2.28	3.57	6.93	4.80	6.22	2.29	3.58	6.93	4.80	6.36
12 ($R_{12} = 9.53 \times 10^{-3}$)	2.15	3.12	6.58	5.16	6.93	2.15	3.14	6.61	5.21	6.93
20 ($R_{20} = 4.36 \times 10^{-5}$)	2.10	3.06	6.58	5.51	6.93	2.10	3.01	6.52	5.37	7.02
true values	2.10	3.00	6.50	5.40	7.10	2.10	3.00	6.50	5.40	7.10

4.4. Extended algorithm

In order to test the working of the extended SSID algorithm of Sheffler et al. (2019), a small test similar to that of Example 1 for the basic algorithm (Section 4.3.1) was performed. As this publication did not contain examples, these had to be made manually. After setting up the problem, the initial goal was to test the working of the algorithm for the normal case as treated in Sheffler et al. (2019). After this, the test was extended by checking whether the algorithm was able to identify negative time delays as well. Finally, the algorithm was tested to identify parameters when noisy frequency response data is present.

4.4.1. Problem setup

Assuming the block diagram as given in Figure 3.6, discrete time transfer functions were chosen for the different subsystems. The main requirement in this process was to form an asymptotically stable closed loop transfer function. With this requirement, many different options are available, and the resulting system is given by Equation (4.11). Note that for this system, feedback- and feedforward time delays are also identified. Therefore, τ_{fb} and τ_{ff} are the discrete time delays in time steps. Combined with the sampling time $T_s = 0.02s$, they can give the continuous time delays in seconds.

$$G_d(z) = \frac{0.04}{z - 0.1}, \quad G_{ff}(z) = \frac{0.8z^2 - 0.3z - 0.5}{z^2}, \quad G_{fb}(z) = \frac{0.3}{z - 0.5}, \\ \tau_{ff} = 20, \quad \tau_{fb} = 15 \tag{4.11}$$

The feedback candidate pool Φ is then formed by first assuming a range of possibilities of $[-5.00, -4.99, \ldots, 5.00]$ for each parameter of G_{fb} and a range of $[5, 6, \ldots, 25]$ for the feedback time delay. To form Φ, all possibilities that result in an unstable closed loop system are discarded. For the feedforward delay candidate pool, the same range as for the feedback time delay is chosen.

4.4.2. Noiseless data example

Running the algorithm as described in Section 3.4 results in the identified parameters equal to the ones defined in Equation (4.11). This verifies the method used in Sheffler et al. (2019) very basically. However, the original method as explained by Sheffler et al. (2019), is said to only work for positive time delays, where a human is actually delayed. The technique, however, is not designed to identify negative time delays in which the human operator 'anticipates' to have a negative delay term. Especially considering the recently developed preview model by van der El et al. (2016), it is interesting to check whether SSID can identify such negative delays.

In order to investigate whether SSID can identify negative feedforward time delays, the feedforward discrete time delay τ_{ff} in Equation (4.11) is changed to -10. The range for ψ in the feedforward candidate pool is adjusted to range from $[-15, -14, \ldots, 15]$. When running the SSID algorithm, the identified parameters do in fact correspond to the true parameters. This indicates that SSID appears to be able to identify negative feedforward time delays. As this test has only been performed for a single example, more tests should be performed. However, at first sight it appears that no large objections or problems will arise and the result therefore looks promising.

4.4.3. Noisy frequency response data

After showing that the SSID technique is able to identify positive and negative feedforward time delays, the next step involves the addition of noise to the frequency response data. This was done using the same noise as was generated for the basic algorithm (see Section 4.2), which was added to the deterministic frequency response. Note that the noise index is proportional to the inverse noise intensity, i.e., high noise index i is a low noise intensity. Here the noise derived from the PSD was used.

Just as before, a plot containing the 2-norm of the difference between the identified and the true parameters was made. Figure 4.7 shows how the 'misfit' of the identification decreased with decreasing noise intensity. Notable are the 'steps' that can be observed in the plots, which are caused by the integer changes of identified ψ and γ. These changes appear to have a large influence on the identified parameters. This is supported by the results shown by Section 4.4.3, which shows the values of the identified parameters for several noise intensities. Especially the difference between the identified α parameters of step 12 and 13 are large. While the ϕ parameters are already identified correctly, the change in ψ causes a large change in the results of the α parameters. This is an interesting finding that should be investigated in more detail.

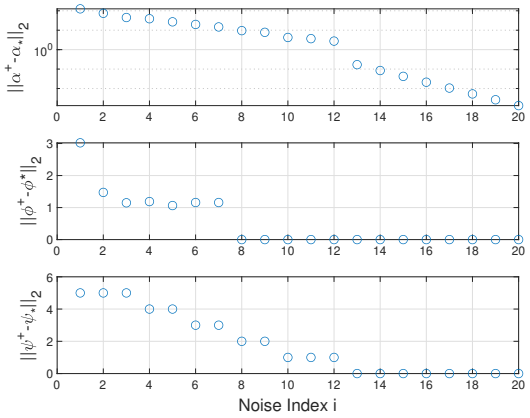


Figure 4.7: Difference between the identified parameters α^+, ϕ^+ and ψ^+, and the true parameters α_*, ϕ_* and ψ_* for different noise intensities, using the defined candidate pools Φ and Ψ. The noise index is proportional to the inverse of the noise intensity.

4.5. Synthesis

This chapter treated the implementation of SSID algorithms based on the work by Zhang and Hoagg (2016a) and Sheffler et al. (2019). For the first algorithm, the basic algorithm as discussed in Sec-

Table 4.3: Values of the identified parameters for several noise intensities and the true parameters of the performed example.

	non-CP			CP			
	α			ϕ			ψ
i	α_1	α_2	α_3	β_1	β_2	γ	ψ
1	37.7258	-96.5813	71.6030	0.50	-0.80	12	-5
5	10.4113	-21.9854	12.3842	0.50	-0.20	16	-6
10	2.4602	-3.7501	1.3136	0.30	-0.50	15	-9
12	-0.3491	1.9530	-1.5982	0.30	-0.50	15	-11
13	0.8619	-0.4365	-0.4225	0.30	-0.50	15	-10
20	0.8005	-0.3011	-0.4994	0.30	-0.50	15	-10
∞	0.8000	-0.3000	-0.5000	0.30	-0.50	15	-10

tion 3.2, examples provided by the original work were reproduced and compared. The results of these reproductions corresponded well with the original work. However, there were some unclarities regarding the noise that was added to the generated data.
First, due to a lack of information, it was impossible to exactly reproduce the noise used in the original example. Information was missing regarding both the exact generation of the noise sequence and the signals required to determine the noise-to-signal ratio. After assuming values for the missing parameters (the input signals) and generating the noise signal using two different ways, results were produced that, taking into account the deviations due to the randomness of the noise, match the original results well.
A second, more relevant problem regarding the noise addition considered the way of modelling the noise. The noise sequences, inserted at the system input and output, and the resulting noise-to-signal ratio, were different from common ratios used in literature. An attempt was made to relate the noise levels defined in the paper of Zhang and Hoagg (2016a) to more intuitive noise models, as this could provide more certainty about the performance of SSID in real-life human tracking tasks. However, due to the many differences in the two definitions, it was found that conclusions can hardly be drawn.

Besides the examples provided by Zhang and Hoagg (2016a), which showed the working of the basic algorithm of SSID, a test was performed for the extended SSID algorithm that is able to identify feedback- and feedforward time delays as well, see Section 3.4. For this algorithm, examples were not present and thus an example was setup. Just as for the basic algorithm, the extended version of SSID appeared to converge given a sufficiently small noise-to-signal ratio. However, it appears that errors in estimating the time delays can significantly affect the performance of the algorithm.

5

Research scope

The previous chapters have provided an overview of the research area surrounding subsystem identification (SSID). Regarding the modelling feedback- and feedforward manual control behavior, the state-of-the-art the model structures, types of tracking tasks and identification methods have been presented. Furthermore, the working of (the different versions of) SSID was explained, including some reproduced examples to verify the method. With this information, a foundation has been laid to support the research objective as was already mentioned in the introduction:

Contribute to the set of tools used to identify feedback- and feedforward behaviour in manual control cybernetics by exploring the capabilities and limitations of subsystem identification.

With this objective and the aforementioned background in mind, the next step in the process consists of formulating the important questions that need to be answered for this research. According to these questions, some key aspects regarding the approach for the research will be defined.

5.1. Research questions

To achieve the set objective, a number of questions need to be answered. The main research question that can be derived from the objective is:

"With respect to the current state-of-the-art, to what extent can subsystem identification successfully identify feedback- and feedforward subsystems including time delays?"

From this main research question, a number of sub-questions originate:

1. How does the level of the human remnant affect the performance of SSID?
2. How do the forcing functions affect the performance of SSID?
3. How do changes in the parameters of the pilot model affect the performance of SSID?
4. How do the controlled element (CE) dynamics affect the performance of SSID?

Just as posed in the main research question, the aim is to answer these questions relative to the state-of-the-art techniques currently available. The four questions address different parts of the human controller model, as can be observed in the block diagrams shown in Chapter 2. It should be noted that the sub-questions have been placed in order of decreasing relevance. As it is currently unclear to what extent answering all these questions is feasible, priorities are set. This means that some aspects are going to be fixed initially to investigate the effects of other parameters. Potentially, some aspects are left to be treated in a next research project.

A first priority is to find out whether SSID can successfully identify human control behavior in realistic situations, and under what circumstances it can do so. As realistic identification processes of human control behavior contain significant amounts of human remnant (or noise), it is important to check to

what extent SSID can deal with this. Although this has already been investigated to some extent in the two papers on SSID (see also Section 4.3 and Zhang and Hoagg (2016b)), it is desirable to determine these noise levels in terms of realistic, non-white human remnant levels. Besides, the initial influence of noise was evaluated without estimating time delays.

Almost equally important as the affect of human remnant levels, are the effects caused by the different forcing functions. Besides the fact that the primary forcing function, the target signal, has a large influence on the type of tracking task and can evoke feedforward control behaviour (i.e. when using predictable signals), the secondary forcing function is more interesting for this research. As was already mentioned in Section 3.5, the SSID technique is in a way similar to the multi-loop Fourier Coefficients (FC) method (Section 2.3.1), as it uses a frequency domain approach. However, as it estimates the multi-loop subsystems from the closed-loop data, it requires only a single forcing function. Determining to what extent the disturbance function, the second forcing function, affects the identified parameters will give valuable information about the properties of SSID, but also about the performance of the multi-loop FC method.

With the first two points treated, another important aspect to check is the dominance of certain parameters or subsystems in the closed-loop frequency response. Depending on the strength of a subsystem or the sensitivity of a parameter, variations might not be able to be detected by SSID. Understanding this parameter sensitivity allows to judge the validity of experimentally obtained results.

A last aspect that can be investigated when evaluating the performance of SSID is the effect of different CE dynamics. As the CE dynamics directly influence the remnant levels (van der El et al., 2019) and the model structure of the different subsystems, they will have an impact on the performance of SSID. However, considering the larger importance of the previously stated aspects, the influence of the CE dynamics has a lower priority.

5.2. Research approach

To find the answers to the questions described in the previous section, a large number of other questions will need to be answered to define the methods and processes to achieve the objective. A couple of main questions that define the research are:

1. With which types of research methods will the performance of SSID be evaluated?
2. Which identification method can function as a good baseline reference, when evaluating SSID?
3. Which version of the subsystem identification algorithm is used for analysis.
4. Which type of tracking task should be selected for identification of feedback- and feedforward subsystems?
5. Which model structure should be used for identifying the different subsystems?

The basic setup of the research is found by answering these questions, which in turn will create new questions. First, the main generic methods to evaluate SSID that will be used are simulations and an experiment. As the identification technique needs to be analysed under different circumstances, tests are required. Simulations are then a good option to verify and find the limits of the performance of such a technique. Validation of the technique is then better performed using an experiment. Most of the research, however, is expected to involve simulation. As the setup of simulations and an experiment requires a significant effort, this is addressed in future chapters. First, answers to the other questions are to be found.

When choosing the identification method that can be used as a reference when evaluating SSID, different options are available (see Section 2.3). From the described options, the best option is the multi-loop FC method discussed in Section 2.3.1. The main reason for this is the similar working of the methods. Figure 5.1 clearly shows the difference between SSID and the multi-loop FC method. Both operating in the frequency domain, the multi-loop FC method works by estimating two FRFs and fitting parameters to these separate responses. SSID, on the other hand, only uses one set of closed-loop frequency

response data and fits the parameters of two subsystems onto this data. An advantage is that a disturbance function is not required, while it can be more difficult to find parameters of a multi-loop model structure based on a single closed-loop FRF. After all, the combined response might hide the effect of certain parameters that have a smaller effect.

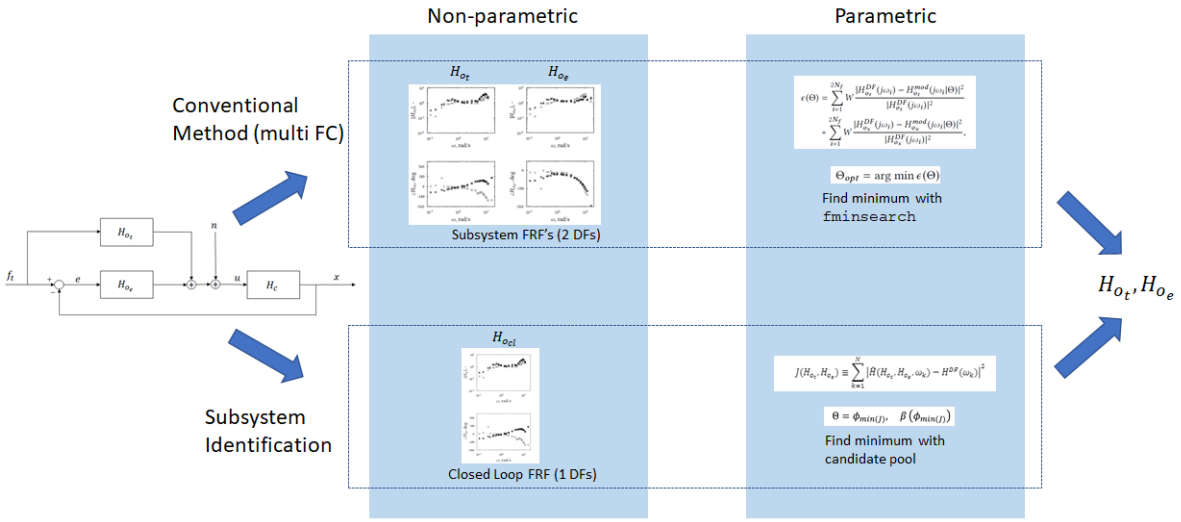


Figure 5.1: Schematic overview of the difference between subsystem identification and the multi-loop Fourier Coefficients method.

Regarding the version of the subsystem identification method that is investigated, an answer was already provided in Chapter 3. As the last version of the algorithm developed by Sheffler et al. (2019) is able to identify both feedback and feedforward time delays, it is considered to be the most complete version of SSID, able to identify the most realistic human operator models. It is therefore selected.
The tracking task that is used for the evaluation of SSID is a pursuit- and preview tracking task. When evaluating SSID, it is important to select a tracking task that allows a human to use both feedback- and feedforward control behavior. Then, pursuit- and preview tracking tasks or tracking tasks that use predictable target signals can be used. However, as mentioned in Section 2.2.2, tasks involving predictable signals often use signals with power at all frequencies (Drop et al., 2019). As both SSID and multi-loop FC use Fourier Coefficients, these types of tasks are not suitable as they require high signal-to-noise ratios at the evaluated frequencies. Another argument for using pursuit- and preview tracking tasks is that well-fitting and intuitive models have been found (see Section 2.2.1).

This almost directly answers the fourth- and last main question, which considers the proposed model structure. Based on the selected tracking tasks, it makes sense to select the TX model structure to apply the models as defined by van der El et al. (2016). However, SSID has not yet been applied to models with such a structure, as it has only been applied to ET model structures (Zhang and Hoagg, 2016a,b; Zhang et al., 2018; Sheffler et al., 2019). It should therefore be investigated whether SSID is able to identify all the parameters in this model before confirming this answer.

As mentioned before, the answers to these questions have either led to multiple other questions, or have required additional research. This will be treated in the coming chapters. First, it will be investigated whether SSID can be applied to a TX model structure. Then the setup of the simulations will be treated.

6

Application of SSID to alternative model structures

To evaluate whether subsystem identification (SSID) can be applied to realistic human control models that apply both feedback- and feedforward, it should be checked whether it can be applied to the model structures of such models. This chapter will therefore try to do this. First, some general information regarding the different notations used throughout the research field and this report is provided. Then, the application of SSID to TX models is evaluated. First for the general TX model structure, and then to the well-working pursuit- and preview model developed by van der El et al. (2016). Finally, the application to a single loop pre-filter model structure is evaluated.

6.1. Notations

As was already mentioned in Section 3.1.1, there are quite some differences regarding the notations used in the work done on SSID (Zhang and Hoagg, 2016a; Zhang et al., 2018; Sheffler et al., 2019). When viewing this in an even broader context by including other relevant work done on preview control (van der El, 2018), this becomes even worse. Therefore, the use of the sign conventions throughout the next steps of the research are discussed in this section.

In this research, the notations used by van der El (2018) will be followed where possible. The reason for this is that the parameters and subsystem have a clear and intuitive meaning. Subsystems such as the target and the output response are therefore modelled as H_{o_t} and H_{o_x} and the feedforward time delay (of the far-viewpoint) is written as τ_f. For the subsystem terms, the subscript o is at times omitted for brevity.

With respect to the candidate pools of SSID and the parameterization of the subsystems, the notation used by Sheffler et al. (2019) is followed when possible. This notation distinguishes the parameterization and the use of different symbols best for identification of subsystems including both feedback (FB) and feedforward (FF) time delays. For both the notation of the model structure and its parameters and the SSID parameters, it will be mentioned when deviating from the general notation (for example in Section 6.2.

6.2. Problem derivation

To apply SSID to the model by van der El et al. (2016), the first step consists of investigating whether SSID can be applied to a TX model structure. Figure 6.1 shows the block diagram for a TX structure. The closed-loop transfer function is then given by Equation (6.1). Note that the right side of the equation uses the assumption that $H(s) = \frac{N(s)}{D(s)}$ and that this first investigation does not involve time delays. Comparing Equation (6.1) to the Equation (3.4) in Section 3.2, the ET-version of the equation, it can be seen that the numerator of the TX model consists only one term. This is due to the fact that the output response H_{o_x} does not appear in the open-loop response for the TX model structure, which does

happen for the error response H_{o_e} in the ET model structure. Here, the SSID notation of the basic SSID algorithm by Zhang and Hoagg (2016a) is used, as this section evaluates the basic application of SSID analogous to that research.

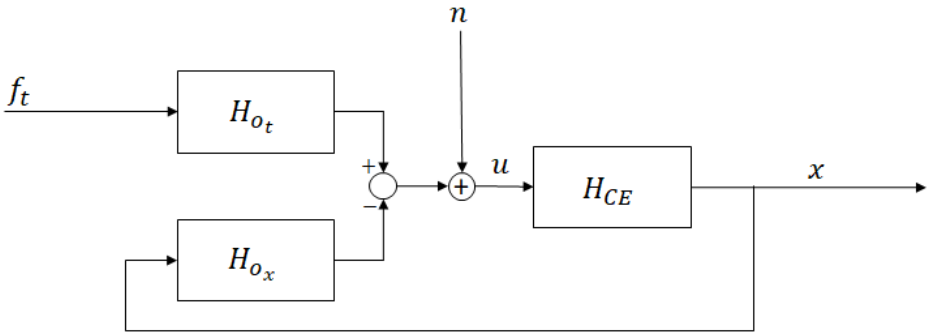


Figure 6.1: Basic TX model structure

$$H_{cl}^{TX} = \frac{H_{o_t} H_{CE}}{1 + H_{CE} H_{o_x}} = \frac{N_{CE} D_{o_x} N_{o_t}}{D_{o_t} \left[D_{CE} D_{o_x} + N_{CE} N_{o_x} \right]} \tag{6.1}$$

Continuing the problem derivation as done for the basic algorithm in Section 3.2, the subsystems can still be parameterized by their coefficients. Again, the feedforward numerator coefficients can be parameterized separately as β (as was done in the basic algorithm), while the feedforward denominator- and the feedback coefficients are the coefficients in ϕ that will form the candidate pool Φ. Equation (6.2) then shows the closed loop function $\tilde{H}_{cl^{TX}}(s, \phi, \beta)$ in terms $\tilde{\mathcal{N}}_{o_t}(s, \phi)$ and $\mathcal{D}(\tilde{s}, \phi)$, which combine the parameterized numerators ($\mathcal{N}_{o_t}(s, \beta)$ and $\mathcal{N}_{o_x}(s, \phi)$) and denominators ($\mathcal{D}_{o_t}(s, \phi)$ and $\mathcal{D}_{o_x}(s, \phi)$) in Equations (6.3) and (6.4). Note that the parameterized numerators and denominators (e.g. $\mathcal{N}_{o_x}(s, \phi)$) are just the subsystem numerators ($N_{o_x}(s)$, but with the coefficients parameterized in terms of the candidate pool entries. Here n_{o_t} is equal to the degree of the feedforward numerator.

$$\tilde{H}_{cl}^{TX}(s, \phi, \beta) = \frac{\tilde{\mathcal{N}}_{o_t}(s, \phi)\beta}{\tilde{\mathcal{D}}(s, \phi)} \tag{6.2}$$

$$\tilde{\mathcal{N}}_{o_t}(s, \phi) = N_{ce}(s)\mathcal{D}_{o_x}(s, \phi)\Gamma_{n_{o_t}}^{\mathrm{T}}(s) \tag{6.3}$$

$$\tilde{\mathcal{D}}(s, \phi) = \mathcal{D}_{o_t}(s, \phi)\left[D_{CE}(s)\mathcal{D}_{o_x}(s, \phi) + N_{CE}(s)\mathcal{N}_{o_x}(s, \phi) \right] \tag{6.4}$$

In these equations, the term second term in the numerator of Equation (3.7) is thus also absent. Continuing the derivation analogous to Section 3.2 this continues, with the consequence that the terms Ω_0 and Ω_1 simplify. The global minimizer β can, however, still be computed for each entry in the candidate pool, as the procedure is not affected. Therefore, the algorithm is therefore expected to work properly.

In short, from a preliminary analysis it can be concluded that the SSID algorithm can be applied to a TX model structure. In fact, as the output response H_{o_x} does not appear in the open-loop, the mathematics simplify with respect to an ET structure. Although this is a good sign regarding the application of SSID to the pursuit- and preview model of van der El et al. (2016), this should still be checked. This is done in the next section.

6.3. Applying SSID to a TX pursuit- and preview model

From the previous section it was concluded that it appears to be possible to perform SSID to a TX model structure. This section will extend this evaluation, by checking whether SSID can be used to estimate the parameters of the preview model as developed by van der El et al. (2016). This will be done for the full- and the reduced model (see Section 2.2.1).

6.3.1. Applying SSID to the full model

According to the model as derived by van der El et al. (2016), the feedforward subsystem H_{o_t} and the feedback subsystem H_{o_x} are given by Equations (6.5) and (6.6), see also Section 2.2.1. These subsystems are build up out of different components: three time delays for the far- and near-viewpoint feedforward and the feedback responses (τ_f, τ_n and τ_v) and the subsystem components $H_{o_f}, H_{o_{e^\star}}, H_{o_n}$ and H_{nms}, for which the response dynamics are given by Equation (6.7). Note that H_{o_f} and H_{o_n} represent the far- and near viewpoint responses, $H_{o_{e^\star}}$ is the compensatory response according to the simplified crossover model of McRuer and Jex (1967) and H_{nms} represents the neuromuscular dynamics. Note that depending on the controlled element dynamics, either $T_{L,e^\star}$, $T_{l,e^\star}$ or both are equal to 0 (McRuer and Jex, 1967).

$$H_{o_t} = \left[H_{o_f} H_{o_{e^\star}} e^{\tau_f j\omega} + H_{o_n} e^{\tau_n j\omega}\right] H_{nms} e^{-\tau_v j\omega} \tag{6.5}$$

$$H_{o_x} = H_{o_{e^\star}} H_{nms} e^{-\tau_v j\omega} \tag{6.6}$$

$$\begin{gathered} H_{o_f}(j\omega) = K_f \frac{1}{1 + T_{l,f} j\omega}, \quad H_{o_{e^\star}}(j\omega) = K_{e^\star} \frac{1 + T_{L,e^\star} j\omega}{1 + T_{l,e^\star} j\omega}, \quad H_{o_n}(j\omega) = K_n \frac{j\omega}{1 + T_{l,n} j\omega}, \\ H_{nms}(j\omega) = \frac{\omega_{nms}^2}{(j\omega)^2 + 2\zeta_{nms}\omega_{nms} j\omega + \omega_{nms}^2} \end{gathered} \tag{6.7}$$

When applying this model to SSID, all the subsystems will be split up in numerators and denominators according to $H(j\omega) = \frac{N(j\omega)}{D(j\omega)}$. Applying this to the subsystem components of Equation (6.7) and substituting Equations (6.5) and (6.6) in Equation (6.1), results in Equation (6.8). Note that the operator subscripts (the 'o' in e.g. H_{o_t}) and the dependencies of all numerators and denominators on $j\omega$ are omitted for brevity.

$$H_{cl}^{TX} = \frac{N_{CE} N_{nms} e^{-\tau_{e^\star} j\omega} \left(N_f D_n N_{e^\star} e^{\tau_f j\omega} + N_n D_{e^\star} D_f e^{\tau_n j\omega}\right)}{D_f D_n \left(D_{CE} D_{nms} D_{e^\star} + N_{e^\star} N_{CE} N_{nms} e^{-\tau_{e^\star} j\omega}\right)} \tag{6.8}$$

In order to apply SSID, a number of problems arise, all of which caused by the feedforward subsystem. First, there are two feedforward delay terms, which need to be estimated. Second, the feedforward numerator coefficients that become the global minimizer (β in Section 3.2 and α in Section 3.4) would now be the coefficients from both N_f and N_n. Finally, the assumption of modelling this feedforward subsystem as FIR causes a problem as it limits the accuracy of estimating the FF subsystem.

From these three problems, the last two might be solved. However, estimating two feedforward time delays appears to be too much for the most recently developed version of SSID by Sheffler et al. (2019). Although they developed the technique such that it could estimate one feedforward time delay, it cannot elegantly estimate two separate feedforward time delays with the current algorithm. A straight forward alternative for solving this problem would be to add all parameters of either the far- or near-viewpoint response to the large main feedback candidate pool (previously referred to as the feedback candidate pool). Then, the amount of calculations would increase significantly, but the problem would be able to be solved. Considering the multiple problems that remain, it is concluded that SSID is currently not deemed suitable to identify the parameters of the full model by van der El et al. (2016).

6.3.2. Applying SSID to the reduced model

There is thus no straight-forward way to apply SSID to the full preview model. However, as was already mentioned in Section 2.2.1, in more recent work by van der El et al. (2018b, 2020), it was determined that the contribution due to the near-viewpoint response can, in most cases, be neglected. This assumption reduces Equation (6.5) to Equation (6.9).

$$H_{o_t} = H_{o_f} H_{o_{e^\star}} e^{\tau_f j\omega} H_{nms} e^{-\tau_v j\omega} \tag{6.9}$$

In order to compare the equation to the setup of SSID by Sheffler et al. (2019), the system is written in terms of the feedforward transfer function, the feedback transfer function and the feedback- and feedforward time delays separately. Equation (6.10) shows the equations for the subsystem transfer functions, which are divided into numerators and denominators.

$$H_{o_t} = \frac{N_t}{D_t} e^{\tau_f j\omega} e^{-\tau_v j\omega}, \quad H_{o_x} = \frac{N_x}{D_x} e^{-\tau_v j\omega} \tag{6.10}$$

Assuming that the components H_{o_t} and H_{o_x} (so H_{o_f}, $H_{o_{e^\star}}$ etc.) are of the form $H = \frac{N}{D}$ (as defined in Equations (6.5) and (6.6)), the numerators N_t and N_x and denominators D_t and D_x can be calculated. Equations (6.11) to (6.14) show these worked out numerators and denominators. Note that either one or both of the lead- and lag time constants of the feedback subsystem ($T_{L,e^\star}$ and $T_{l,e^\star}$) will be zero depending on the controlled element dynamics.

$$N_t = N_f N_{e^\star} N_{nms} = K_f K_{e^\star} \omega_{nms}^2 (T_{L,e^\star}(j\omega) + 1) \tag{6.11}$$

$$\begin{aligned} D_t = D_f D_{e^\star} D_{nms} = {} & (j\omega)^4 T_{l,f} T_{l,e^\star} \\ & + (j\omega)^3 \left(T_{l,f} T_{l,e^\star} 2\zeta_{nms}\omega_{nms} + T_{l,f} + T_{l,e^\star}\right) \\ & + (j\omega)^2 \left(T_{l,f} T_{l,e^\star} \omega_{nms}^2 + 2\zeta_{nms}\omega_{nms}(T_{l,f} + T_{l,e^\star}) + 1\right) \\ & + (j\omega)^1 \left(\omega_{nms}^2 (T_{l,f} + T_{l,e^\star}) + 2\zeta_{nms}\omega_{nms}\right) \\ & + (j\omega)^0 \omega_{nms}^2 \end{aligned} \tag{6.12}$$

$$N_x = N_{e^\star} N_{nms} = (j\omega)^1 K_{e^\star} T_{L,e^\star} \omega_{nms}^2 + (j\omega)^0 K_{e^\star} \omega_{nms}^2 \tag{6.13}$$

$$\begin{aligned} D_x = D_{e^\star} D_{nms} = {} & (j\omega)^3 T_{l,e^\star} \\ & + (j\omega)^2 \left(T_{l,e^\star} 2\zeta_{nms}\omega_{nms} + 1\right) \\ & + (j\omega)^1 \left(T_{l,e^\star} \omega_{nms}^2 + 2\zeta_{nms}\omega_{nms}\right) \\ & + (j\omega)^0 \omega_{nms}^2 \end{aligned} \tag{6.14}$$

The set of equations is presented in this specific way, because SSID identifies the coefficients of the numerators and denominators N_t, N_x, D_t, D_x. From this set of equations, it can be seen that, depending on type of controlled element dynamics, the system will have 9 to 11 equations and 5 or 6 unknowns. This means that the system is overdetermined and problems might arise when trying to solve for the model parameters. This will be further addressed towards the end of this sub-section.

With N_t, D_t, N_x and D_x, the closed loop system can be written as Equation (6.15) and the discretized version $H_{cl,d}^{TX}$ is given by Equation (6.16). Here, d_f and d_v are the discrete feedback and far-viewpoint time delays (given $\tau = dT_s$), respectively.

$$H_{cl}^{TX}(s) = \frac{e^{-\tau_v j\omega} N_{CE} D_x N_t e^{\tau_f j\omega}}{D_t \left[D_{CE} D_x + N_c N_x e^{-\tau_v j\omega}\right]} \tag{6.15}$$

$$H_{cl,d}^{TX}(z) = \frac{z^{-d_v} N_{CE,d} D_{x,d} N_{t,d} z^{d_f}}{D_{t,d} \left[D_{CE,d} D_{x,d} + N_{c,d} N_{x,d} z^{-d_v}\right]} \tag{6.16}$$

The discrete closed loop system can then be parameterized by the coefficients. Just as done by Sheffler et al. (2019), the feedforward numerator coefficients are contained in α, the feedback candidate pool coefficients are used in β, while the feedback- and feedforward delay terms are γ and ψ, respectively. This equation looks quite similar to the form obtained by Sheffler et al. (2019) and it appears that SSID is able to identify the model parameters.

$$\tilde{H}_{cl,d}^{TX}(z, \alpha, \psi, \beta, \gamma) = \frac{z^{-\gamma} N_{CE,d}(z) \mathcal{D}_{o_x,d}(z, \beta) \mathcal{N}_{o_t,d}(z, \alpha) z^{\psi}}{\mathcal{D}_{o_t,d}(z, \beta) \left[D_{CE,d}(z) \mathcal{D}_{o_x,d}(z, \beta) + N_{CE,d}(z) \mathcal{N}_{o_x,d}(z, \beta) z^{-\gamma}\right]} \tag{6.17}$$

Note however that, in contrast to the work of Sheffler et al. (2019), the feedforward subsystem is not modelled as FIR and the feedforward denominator coefficients are also contained in β, see $\mathcal{D}_{t,d}(z, \beta)$. This is analogous with Zhang and Hoagg (2016a), where this assumption had not been made yet. For the extended algorithm, this is not expected to have large consequences except for increased calculation times. If, however, the feedforward subsystem is to be modelled as an FIR, it cannot identify the

parameters of the reduced model used by van der El et al. (2018b, 2020) exactly, but will approximate the results. On the other hand, even without the FIR assumption this is questionable due to the overdetermined set of Equations (6.11) to (6.14).

Overall, it should be concluded that it appears as if SSID is able to successfully identify parameters based on this reduced model structure. Equation (6.17) can be applied to the extended SSID algorithm as developed by Sheffler et al. (2019). However, due to the overdetermined system of equations given by Equations (6.11) to (6.14), it is expected to be difficult to estimate the model parameters of the model proposed by van der El et al. (2016). As this situation is still not ideal, another attempt for identification will involve the reduced model structure containing the pre-filter, as was shown by Figure 2.4.

6.4. Applying SSID to a single loop preview model

As was already shown in Section 2.2.1, the model for pursuit and preview can be rewritten from a TX model structure consisting of a compensatory system combined with a pre-filter. Figure 6.2 shows the model that is formed when omitting the near-viewpoint response. In this model, the pre-filter block H_{o_p} is described by Equation (6.18) and the response to the far-viewpoint error $e^{\star}$ is given by Equation (6.19). H_{o_f}, $H_{o_{e^\star}}$ and H_{nms} are as earlier defined in Equation (6.7).

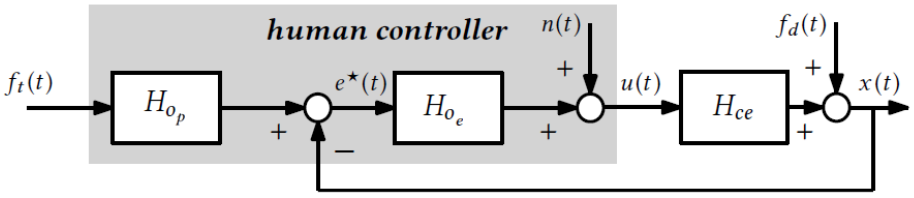


Figure 6.2: Single loop model structure for pursuit- and preview based on van der El et al. (2018b).

$$H_{o_p}(j\omega) = H_{o_f}(j\omega)e^{j\omega\tau_f} \tag{6.18}$$

$$H_{o_e}(j\omega) = H_{o_{e^\star}}(j\omega)H_{nms}(j\omega)e^{-j\omega\tau_v} \tag{6.19}$$

With this model, the closed loop transfer function in terms of the subsystems and its numerators and denominators ($H(j\omega) = \frac{N(j\omega)}{D(j\omega)}$) is as given by Equation (6.20). This transfer function is identical to that of the TX model structure discussed previously, but it is a minimum realization. Based on the structure, SSID should be able to identify the parameters.

$$H_{cl} = \frac{H_{o_p}H_{o_e}H_{ce}}{1 + H_{ce}H_{o_e}} = \frac{N_{ce}N_{o_e}N_{o_f}e^{j\omega(\tau_f - \tau_v)}}{D_{o_f}\left(D_{ce}D_{o_e} + N_{ce}N_{o_e}e^{-j\omega\tau_v}\right)} \tag{6.20}$$

To check whether the identified SSID coefficients can be converted to the HO model parameters, the numerators and denominators of H_{o_p} and H_{o_e} are evaluated in Equations (6.21) to (6.24).

$$N_{o_p} = N_f = K_f \tag{6.21}$$

$$D_{o_p} = D_f = 1 + T_{l,f}j\omega \tag{6.22}$$

$$N_{o_e} = N_{o_{e^\star}}N_{nms} = \omega_{nms}^2 K_{e^\star}(1 + T_{L,e^\star}j\omega) \tag{6.23}$$

$$\begin{aligned} D_{o_e} = D_{o_{e^\star}}D_{nms} = {} & (j\omega)^3 T_{l,e^\star} \\ & + (j\omega)^2\left(T_{l,e^\star}2\zeta_{nms}\omega_{nms} + 1\right) \\ & + (j\omega)^1\left(T_{l,e^\star}\omega_{nms}^2 + 2\zeta_{nms}\omega_{nms}\right) \\ & + (j\omega)^0\omega_{nms}^2 \end{aligned} \tag{6.24}$$

As this system does not contain common divisors (of the symbols) and is thus a minimal realization, there are fewer coefficients to estimate and thus it is better solvable. However, even in this configuration, it is over-determined. This can be solved by writing H_{o_f} in terms of cut-off frequencies instead of lag-time constants, see Equation (6.25). Doing the same for $H_{o_{e^\star}}$ (only for the lag-time coefficient), allows the system of equations to become solvable. Section 6.4 shows the different numerator- and denominator terms that are expected for different types of controlled element dynamics.

$$H_{o_f}(j\omega) = \frac{\frac{K_f}{T_{l,f}}}{j\omega + \frac{1}{T_{l,f}}} = \frac{K_f \omega_f}{j\omega + \omega_f} \tag{6.25}$$

$$H_{o_{e^\star,G}}(j\omega) = \frac{K_{e^\star}}{j\omega + \omega_{l,e^\star}}, \quad H_{o_{e^\star,SI}}(j\omega) = K_{e^\star}, \quad H_{o_{e^\star,DI}}(j\omega) = K_{e^\star}\,(j\omega + TL, e^\star) \tag{6.26}$$

In conclusion, this model structure then appears to be most suitable for identifying the model parameters with SSID, as the identified coefficients can easily be translated to HO model parameters, which in turn can be compared to those identified using the multi-loop Fourier Coefficients (FC) method. It should be noted however, that the FIR assumption needs to be omitted.

Table 6.1: The numerator- and denominator terms $N_f(j\omega)$ and $N_e(j\omega)$ for gain (G), single integrator (SI) and double integrator (DI) dynamics.

CE	G	SI	DI
N_f	$K_f\omega_{l,f}$	$K_f\omega_{l,f}$	$K_f\omega_{l,f}$
D_f	$j\omega + \omega_{l,f}$	$j\omega + \omega_{l,f}$	$j\omega + \omega_{l,f}$
N_e	$\omega_{nms}^2 K_{e^\star}\omega_{l,e}$	$\omega_{nms}^2 K_{e^\star}$	$\omega_{nms}^2 K_{e^\star}(1 + T_{L,e^\star}j\omega)$
D_e	$(j\omega)^3 + (j\omega)^2(\omega_{l,e} + 2\zeta_{nms}\omega_{nms}^2 + (j\omega)(2\zeta_{nms}\omega_{nms}\omega_{l,e} + \omega_{nms}^2) + \omega_{nms}^2$	$(j\omega)^2 + 2\zeta_{nms}\omega_{nms}j\omega + \omega_{nms}^2$	$(j\omega)^2 + 2\zeta_{nms}\omega_{nms}j\omega + \omega_{nms}^2$

6.5. Synthesis

To evaluate the performance of SSID, it is desired to use the results from a well-studied tracking task and human operator model. Considering the capabilities of SSID, the pursuit- and preview model developed by van der El et al. (2016) was selected to be the best option for this. However, as this model used a different model structure (TX instead of ET), it had to be checked whether SSID is able to identify the parameters of such a model structure.

Based on a mathematical derivation, it was concluded that SSID is able to estimate the subsystems of a TX model structure. However, when evaluating whether SSID is also able to identify the parameters of the full pursuit- and preview model, problems regarding the estimation of multiple feedforward time delays, an overdetermined system and the finite impulse response (FIR) assumption arose. Even when considering the reduced model proposed by van der El et al. (2018b) (which contains only one feedforward time delay) and removing the FIR assumption, problems w.r.t. the overdetermined system remained. This overdetermined system causes problems when converting the estimated SSID parameters to the human operator (HO) model parameters.

A promising solution was found in the (reduced) pre-filter model structure proposed by van der El et al. (2018b). This model contains the same input-output behaviour as the TX model but is written in a form that has fewer duplicated coefficients (i.e., it is a minimal realization). Rewriting this model in terms of cut-off frequencies instead of time constants, the coefficients to be estimated reduce the system of equations to one that is neither overdetermined nor underdetermined. This system will be used for the SSID technique in the simulations and experimental analysis of this research.

7

Simulation Setup

The previous chapter addressed one of the main questions regarding the research setup as posed in Section 5.2. With the obtained answers to these questions, a number of other questions have spawned regarding the next steps of the research. These questions mainly address the exact setup of the tests that will be performed, of which are the simulation and the experiment. This chapter will therefore focus on the preliminary simulation setup, which main functions are verifying several of its components and providing a basis for the eventual simulation setup.

This chapter will address a number of topics of the simulations. First, the basic settings of the simulation are discussed. Then, the Simulink implementation is discussed. Thereafter the verification of certain parts of the simulation is treated. Besides these simulation parts, one other aspect that is verified is the application of the multi-loop Fourier Coefficients (FC) method. Finally, some recommendations for future work on the simulation setup are provided.

7.1. Settings

The preliminary simulation setup is largely based on the model obtained by van der El et al. (2016, 2018b). This provides clear parameters and references to verify correctness of the different components of the simulation. The selected model structure thus corresponds to the TX model structure (see Figure 6.1), with the subsystem dynamics of H_{o_t} and H_{o_x} given by Equations (6.5) and (6.6).

Initially, two types of controlled element (CE) dynamics and preview times are used. For the controlled element dynamics, single integrator (SI) ($H_{ce} = \frac{K_{ce}}{s}$) and double integrator (DI) ($H_{ce} = \frac{K_{ce}}{s^2}$) systems are selected with K_{ce} equal to 1.5 and 5, respectively. This is done as SI dynamics are the most intuitive for humans to control, while DI dynamics contain the highest remnant (and thus noise) levels, making it most challenging to successfully identify the DI dynamics(van der El et al., 2020). This last finding is explained in more detail in Section 7.1.3

The preview times are selected to be 0 and 1s, thus resulting in a pursuit- (PS) and preview (PR) task. The different tasks are then labeled SIPS, SIPR, DIPS and DIPR.

7.1.1. Forcing function design

The forcing functions are the functions used to excite the human-in-the-loop system and, depending on the applied identification technique, should be designed carefully. They consist of a target- and a disturbance function, although the latter is not required when using subsystem identification (SSID).

The use of Fourier Coefficients in both SSID and multi-loop FC identification of human control behaviour in pursuit- and preview tasks puts strict requirements on the forcing functions (Damveld et al., 2010). Among others, the forcing functions should be randomly appearing (Yamashita, 1990), sufficiently exciting (McRuer et al., 1965) and have high signal-to-noise ratios at the relevant frequencies

(van Paassen and Mulder, 1998). Therefore, multi-sine signals are generally used in experiments that involve preview displays and FC-based system identification (van der El et al., 2016, 2018a; Sheffler et al., 2019).

The multi-sine signals used for the simulations are based on those used by van der El et al. (2016), with each sine having the form as found in Equation (7.1). The frequencies ω_k are multiples of the base frequency ω_{base}, which corresponds to the radial frequency of a sine with a period of 120s.

Table 7.1 shows the parameters used for constructing the multi-sines, as used by van der El et al. (2016). The signal bandwidth is approximately 1.5 rad/s. In the design, the power distribution was designed to be close to the power distribution used by McRuer et al. (1965). Furthermore, the phases were selected randomly while preventing high or low crest factors. The five different phase realizations that are constructed mainly influence the potential experiment setup, as they are made so that the subjects do not remember the target signal. For the initial simulations, only the first realization will be used.

$$f(t) = \sum_{k=1}^{N_f} A_k \sin(\omega_k t + \phi_k) \tag{7.1}$$

Table 7.1: Forcing function parameters as retrieved from van der El et al. (2016).

	Target forcing function f_t								Disturbance forcing function f_d			
k	n_{t_k} [-]	A_{t_k} [inch]	w_{t_k} [rad/s]	ϕ_{t_1} [rad]	ϕ_{t_2} [rad]	ϕ_{t_3} [rad]	ϕ_{t_4} [rad]	ϕ_{t_5} [rad]	n_{d_k} [rad]	A_{d_k} [inch]	w_{d_k} [rad/s]	ϕ_{d_k} [rad]
1	2	0.240	0.105	3.646	0.174	4.878	2.917	2.709	5	0.093	0.262	2.546
2	3	0.240	0.157	0.030	5.953	2.868	2.040	0.508	6	0.093	0.314	6.264
3	8	0.240	0.419	1.277	0.655	4.205	5.857	3.369	11	0.093	0.576	6.283
4	9	0.240	0.471	2.367	3.526	1.921	4.188	0.477	12	0.093	0.628	1.865
5	14	0.240	0.733	3.901	4.809	3.786	2.602	5.165	18	0.093	0.942	3.196
6	15	0.240	0.785	4.287	3.391	4.201	4.110	3.286	19	0.093	0.995	5.309
7	26	0.240	1.361	5.554	2.965	6.014	3.640	0.619	31	0.093	1.623	3.626
8	27	0.240	1.414	2.411	4.446	5.538	1.626	3.397	32	0.093	1.676	3.229
9	40	0.057	2.094	2.551	1.730	0.183	3.703	3.131	58	0.029	3.037	0.165
10	41	0.057	2.147	0.490	2.952	4.354	6.108	3.926	59	0.029	3.089	0.082
11	78	0.057	4.084	5.431	5.432	1.588	3.406	3.812	93	0.029	4.869	4.233
12	79	0.057	4.136	4.238	3.697	0.270	3.801	5.548	94	0.029	4.922	5.366
13	110	0.057	5.760	5.014	6.200	1.858	5.698	4.196	128	0.029	6.702	5.386
14	111	0.057	5.812	1.768	5.910	0.585	2.903	5.239	129	0.029	6.754	4.756
15	148	0.057	7.749	0.163	5.952	4.385	3.622	1.333	158	0.029	8.273	1.453
16	149	0.057	7.802	0.156	3.563	1.825	4.269	5.889	159	0.029	8.325	3.291
17	177	0.057	9.268	0.214	0.077	0.325	5.841	4.290	193	0.029	10.105	3.243
18	178	0.057	9.320	3.687	1.961	0.269	3.608	0.164	194	0.029	10.158	3.924
19	220	0.057	11.519	5.176	5.484	6.179	4.261	3.196	301	0.029	15.760	3.171
20	221	0.057	11.572	1.266	1.050	0.304	2.325	4.165	302	0.029	15.813	1.976

7.1.2. Human operator models

The human operator (HO) parameters are contained by $\Theta = \left[K_{e^\star}\ T_{L,e^\star}\ T_{l,e^\star}\ \ \tau_v\ \omega_{nms}\ \zeta_{nms}\ K_n\ \tau_n\ T_{l,n}\ K_f\ \tau_f\ T_{l,f}\right]^T$ are given in Table 7.2. They have been determined by van der El et al. (2016) and have later been used in simulations by van der El et al. (2018b). Although the previous chapter mentioned that the near-viewpoint response would be omitted when applying SSID to identify the parameters, it is kept

inside this simulation. This is done for verification purposes, see Section 7.3.2. When running the simulations, these parameters can be set to 0.

Table 7.2: Test conditions and model parameters as obtained from van der El et al. (2016).

H_{ce}	$\frac{K_{CE}}{s}$		$\frac{K_{CE}}{s^2}$	
K_{ce}	1.5		5	
τ_p, s	0	1	0	1
abb.	SIPS	SIPR	DIPS	DIPR
$K_{e^\star}$, -	1.43	1.11	0.14	0.14
$T_{l,e^\star}$, s	-	-	-	-
$T_{L,e^\star}$, s	-	-	2.54	2.22
τ_v, s	0.23	0.18	0.28	0.31
ω_{nms}, rad/s	11.2	10.2	6.15	5.33
ζ_{nms}, -	0.3	0.26	0.67	0.5
K_n, -	-	0.18	-	0.32
τ_n, s	-	0.34	-	0
$T_{l,n}$, s	-	0.04	-	5.89
K_f, -	0.95	1.12	0.54	0.63
τ_f, s	-	0.70	-	0.99
$T_{l,f}$, s	-	0.38	-	0.59

7.1.3. Remnant modelling

Besides the linear part discussed in the previous subsection, the non-linear remnant is an essential part of the quasi-linear human controller model. This remnant, which accounts for all the noise and non-linearities in the HO, is generally modelled as colored white noise. Initial remnant models for compensatory tracking tasks used low-pass filtered Gaussian white noise, injected at the error signal (Levison et al., 1969). Equation (7.2) shows this noise model, where K_r is a gain and $T_{l,r}$ the lag-time constant.

$$H_r(j\omega) = \frac{K_r}{1 + T_{l,r}j\omega} \tag{7.2}$$

More recently, van der El et al. (2019) have come up with remnant models for pursuit- and preview tasks. The same remnant model developed by Levison et al. (1969) can be used, although it is now injected at the system output x, which is fed back into the system. It should be stressed that this is thus significantly different from the approach of Zhang and Hoagg (2016a), who also added noise on the system output but did not feed it back into the system. Depending on the CE dynamics, different break-frequencies ($\omega_b = \frac{1}{T_l}$) are found. This break frequency is smaller than 1 or even 0.1 rad/s for DI CE dynamics, whereas it is between 1-15 rad/s for SI dynamics.

The magnitude of the injected remnant is determined by the relative contribution of the remnant to the control output u. This control output consists of contributions due to the target signal f_t, the disturbance signal f_d and the remnant n. Equation (7.3) shows this relation in equation form for the power spectral densities. To determine the required remnant power, the ratio between the remnant contribution to the control signal and the total control signal $\frac{\sigma_{u_n}^2}{\sigma_u^2}$ (see also Equation (7.4)) is required. Independent of the display and for a target signal bandwidth of 1.5 rad/s, this ratio is approximately equal to 0.33 for SI dynamics and 0.58 for DI dynamics.

$$S_{uu}(j\omega) = S_{uu_t}(j\omega) + S_{uu_d}(j\omega) + S_{uu_n}(j\omega) \tag{7.3}$$

$$\frac{\sigma_{u_n}^2}{\sigma_u^2} = \frac{\int_0^\infty S_{uu_n}(j\omega)d\omega}{\int_0^\infty S_{uu}(j\omega)d\omega} \tag{7.4}$$

The measurement of the output spectrum due to the remnant S_{uu_n} can be found by taking the total control output spectrum at the remnant frequencies ω_n (or all frequencies except for ω_t and ω_d, see Table 7.1). However, this method neglects the power of the remnant at the target- and disturbance frequencies. There are two possible solutions for this problem. The first one is the easiest and uses a correction factor taken from literature. van der El (2018) showed that for the SIPS, DIPS, SIPR and DIPR tasks, the effect of neglecting the remnant power at the target- and disturbance frequencies is an underestimation of the variance of about 18%. Applying this correction factor in combination with the integration of the power spectral density (PSD) at ω_n, then takes into account the effect of this assumption.

An alternative method to account for the remnant power at the target- and disturbance frequencies uses interpolation. The remnant power at the neighbouring remnant frequencies is then used to estimate the power at the target- and disturbance frequencies. This method is more sophisticated and robust, as it is not dependent on the tracking task. Therefore, it is preferred throughout the remainder of the research.

In short, the human remnant is thus modelled as low-pass filtered white noise, injected at the system output x. The white noise intensity and the filter gain are tuned to result in the appropriate relative remnant contribution to the control output signal $\frac{\sigma_{u_n}^2}{\sigma_u^2}$, which is corrected for omitting remnant contribution at the non-remnant frequencies in the calculation of $\sigma_{u_n}^2$. This is either done using a correction factor from literature (van der El et al., 2018b) or by estimating the remnant power at the forcing function frequencies using interpolation. Here, the latter method is preferred.

7.2. Signal shift in Simulink implementation

The models with the settings described above are implemented in a time domain Simulink model. This model uses the time domain target signal as an input and outputs the CE outputs. However, Simulink does not handle the negative time delays of far- and the near-viewpoint very well, as it cannot process the non-causality properly. This can be solved by shifting the target signal forward 'into the future' by at least the far-viewpoint time delay τ_f and model the other delays with respect to this reference. Figure 7.1 explains this time shift in a diagram. In the diagram, the target signal is shifted forward by τ_f and becomes the input to H_{o_t}. Then, near-viewpoint response is delayed by the difference between the far- and the near-viewpoint time delays to shift it correctly. Both responses experience the regular time delay τ_v.

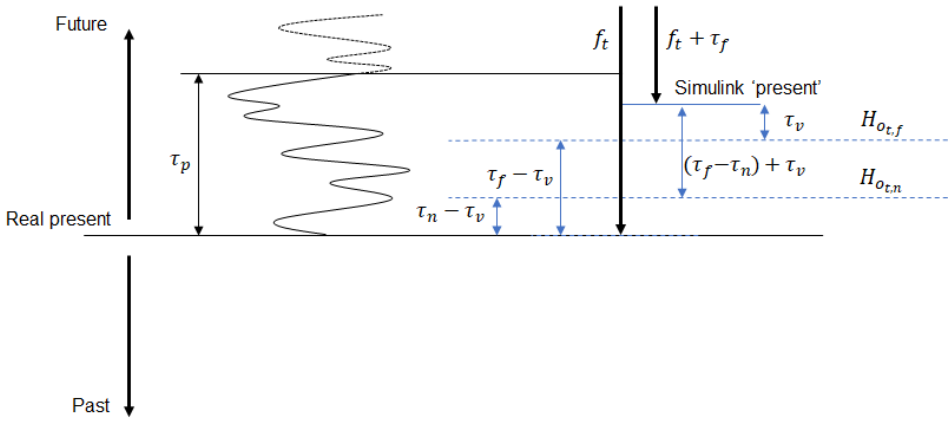


Figure 7.1: Diagram explaining the time shift of the target signal to deal with the negative time delays in the Simulink model

7.3. Simulation verification

When working with simulations, it is crucial to ensure that the obtained results are correct. Therefore, verification activities should be performed at every step of the simulation. This section will focus on the different tests that have been performed to verify certain steps of the simulations. The different parts that are verified concern the subjects that were discussed in the previous three sections.

7.3.1. Forcing function power spectra

To verify the correct implementation of the forcing function data, the power spectra are determined and compared to the results obtained by van der El (2018). Figure 7.2 shows both the reference (from van der El (2018)) and the reproduced power spectra of the target and the disturbance function.

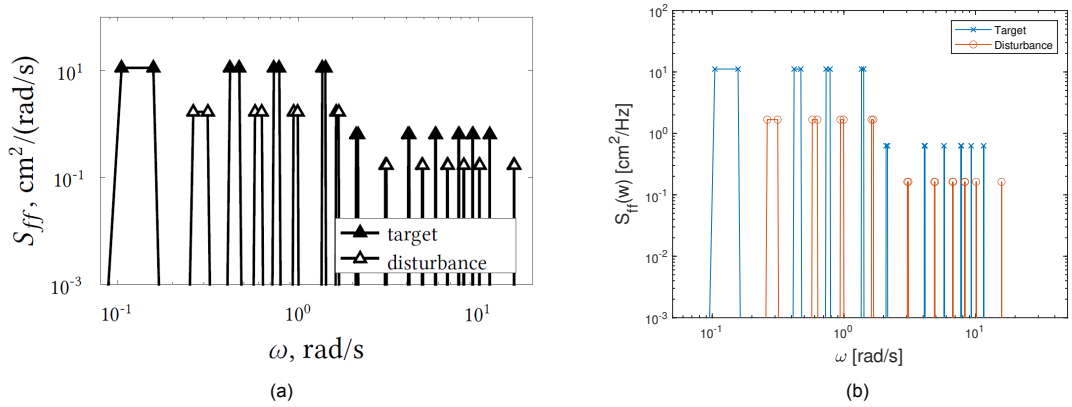


Figure 7.2: Power spectra of the target (f_t) and disturbance (f_d) function by van der El (2018) (a) and reproduced from the data (b).

From the figures it can be concluded that the reproduced power spectra correspond well with the original results. However, it is observed that the y-axis is labelled with a different unit. The reason for this is an erroneous axis label in van der El (2018). As the PSD is initially calculated with the frequency in Hz, the unit is in cm^2/Hz even though it is plotted against ω in rad/s. To convert this unit to $cm^2/(rad/s)$, a division by 2π would be required.

7.3.2. Time domain traces

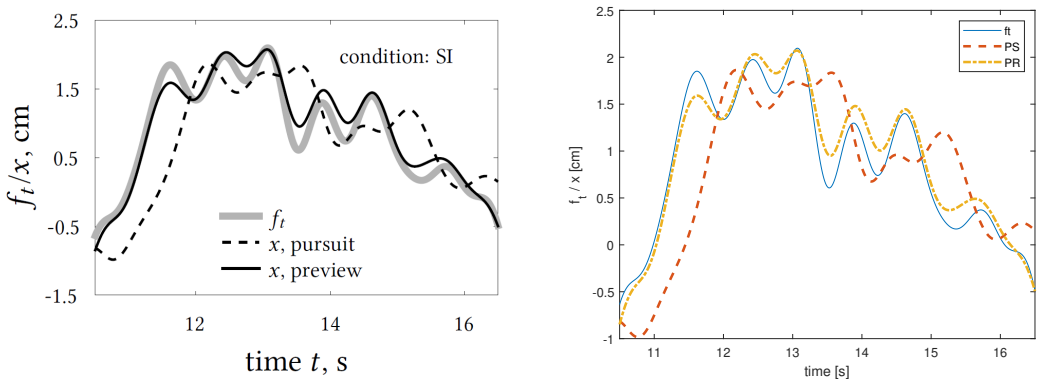
After verifying the forcing functions, the constructed Simulink model with parameters can be tested. This is done by testing the time domain responses of a simulation with only a target function (i.e. $f_d = 0$) and the model parameters as given by Table 7.2. These responses can then be compared to the results obtained by van der El (2018); van der El et al. (2018b). Figures 7.3 and 7.4 show these comparisons for both the SI and the DI simulation.

The figures show identical time responses of the CE output, thus verifying correct implementation of the model. This also implies that the time shift required for simulating the negative time delay of the feedforward channels (see also Section 7.2) is implemented correctly.

7.3.3. Remnant modelling

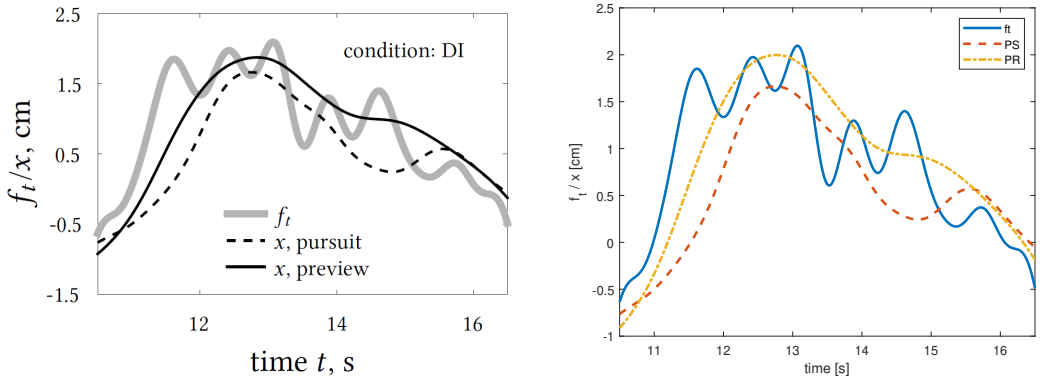
While the two previous parts of the simulation could be verified by exactly comparing the results to those obtained by van der El (2018), this is different when verifying the implementation of the remnant model. As the human remnant is modelled as colored white-noise, an exact comparison is difficult. However, despite the exact noise realization being different, the implementation can be verified by examining the reference results more generally.

Figure 7.5 shows the control output power spectrum for the SIPR control task, both obtained by van der El (2018) and van der El et al. (2019) and by reproduction for this research. Although the exact graphs



(a) (b)

Figure 7.3: Comparison of the simulated CE outputs to the target function for the SI pursuit- and preview models between van der El (2018) (a) and the reproduced data (b).



(a) (b)

Figure 7.4: Comparison of the simulated CE outputs to the target function for the DI pursuit- and preview models between van der El (2018) (a) and the reproduced data (b).

differ, the order of magnitude and shapes of the responses correspond. Just as in Section 7.3.1, the units on the y-axis are different due to an error in the plot of van der El (2018).

A necessary remark addresses the difference in number of realizations between the reference plot (5, see Figure 7.5a) and the reproduced data (1, see Figure 7.5b). The consequence of this difference appears in the plot of Figure 7.5b, where the power at the (specific) disturbance frequencies is lower than at the neighbouring remnant frequencies. This is due to the single noise realization, although even for the reference data, some of the data points at the disturbance frequencies appear to be only barely larger than the remnant power. From this, it can be concluded that it is important to use averages of multiple runs.

7.3.4. Multi-loop identification

A last part that is to be verified concerns the working of the multi-loop FC identification method (as explained in Section 2.3.1). The method results in the estimated frequency response functions (FRFs) of the different subsystems H_{o_t} and H_{o_x}. Figures 7.6 and 7.7 show the 5-run average FRFs and the identified model fits of H_{o_t} and H_{o_x} by van der El (2018), compared to a a single FRF estimation from a reproduced simulation of this same model. This is done for the DIPR case.

The figures show that the reproduced FRFs estimate the simulated models well. Although some offsets are found in, for example, the magnitude of H_{o_x} in Figure 7.7b, these offsets are generally also

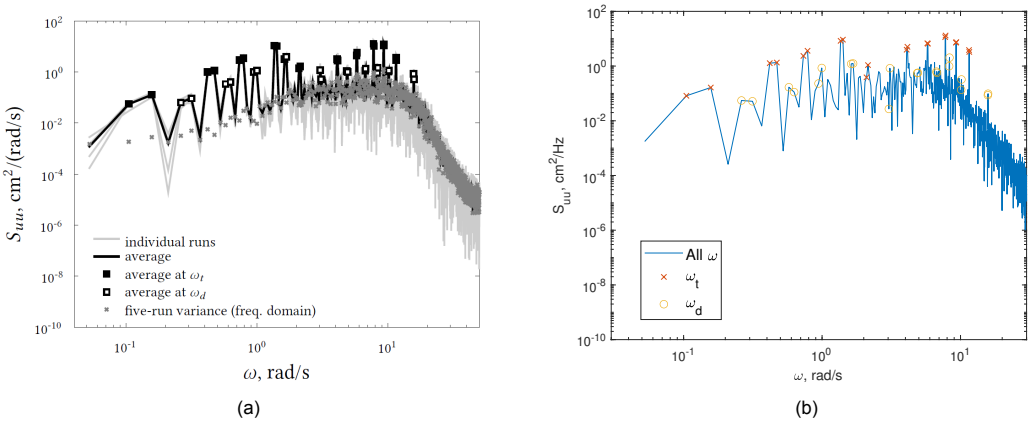


Figure 7.5: Comparison of the control output power spectra for the SIPR simulation between van der El (2018) (a) and the reproduced data (b).

found in the reference data of van der El (2018) (in Figure 7.7a). However, it is observed that for some data points, especially at the edges of the set of frequency data points, large mismatches between the model and the FRFs occur. A clear example of this is found in the magnitude of the target response H_{o_t} in Figure 7.6 for both the reference- and the reproduced data. It is assumed that this is caused by the identification technique and the interpolation of the data points at the target- and disturbance frequencies (as was explained in Section 2.3.1 and by van Paassen and Mulder (1998)). To test this, a closer look should be taken at the results of this interpolation.

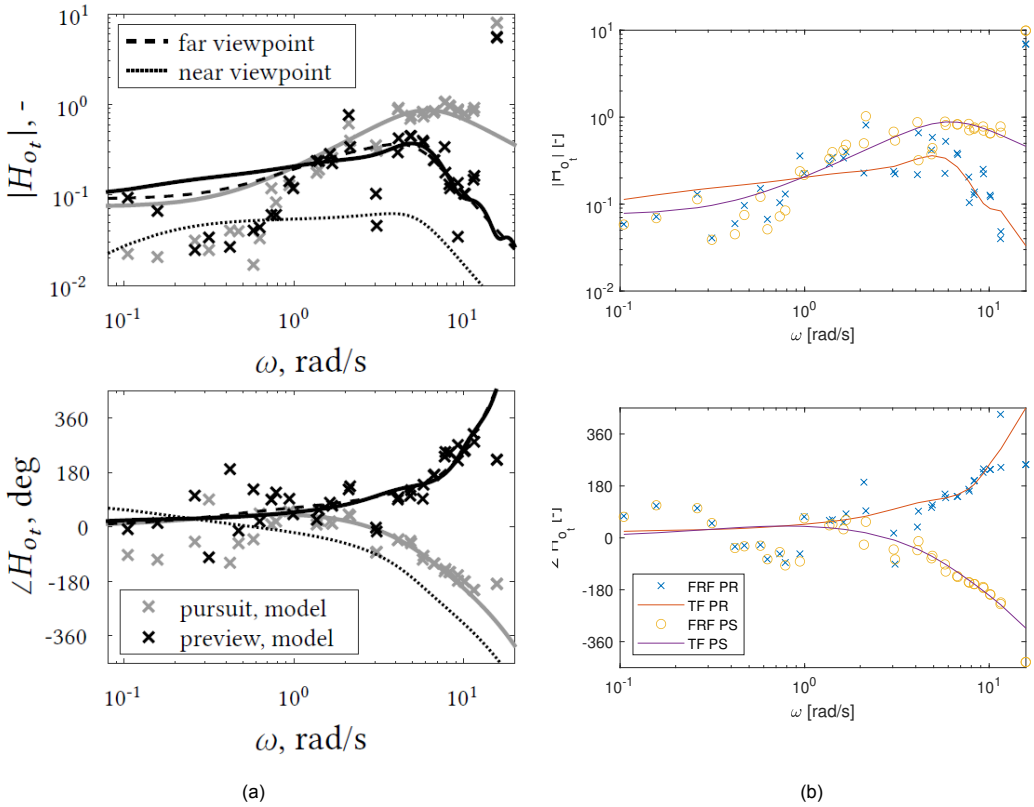


Figure 7.6: Comparison of the experimentally estimated FRFs and the resulting model fits of van der El (2018) (a) and the reproduced FRFs estimated from the simulated models (b) for the target response H_{o_t} for the DIPR experiment.

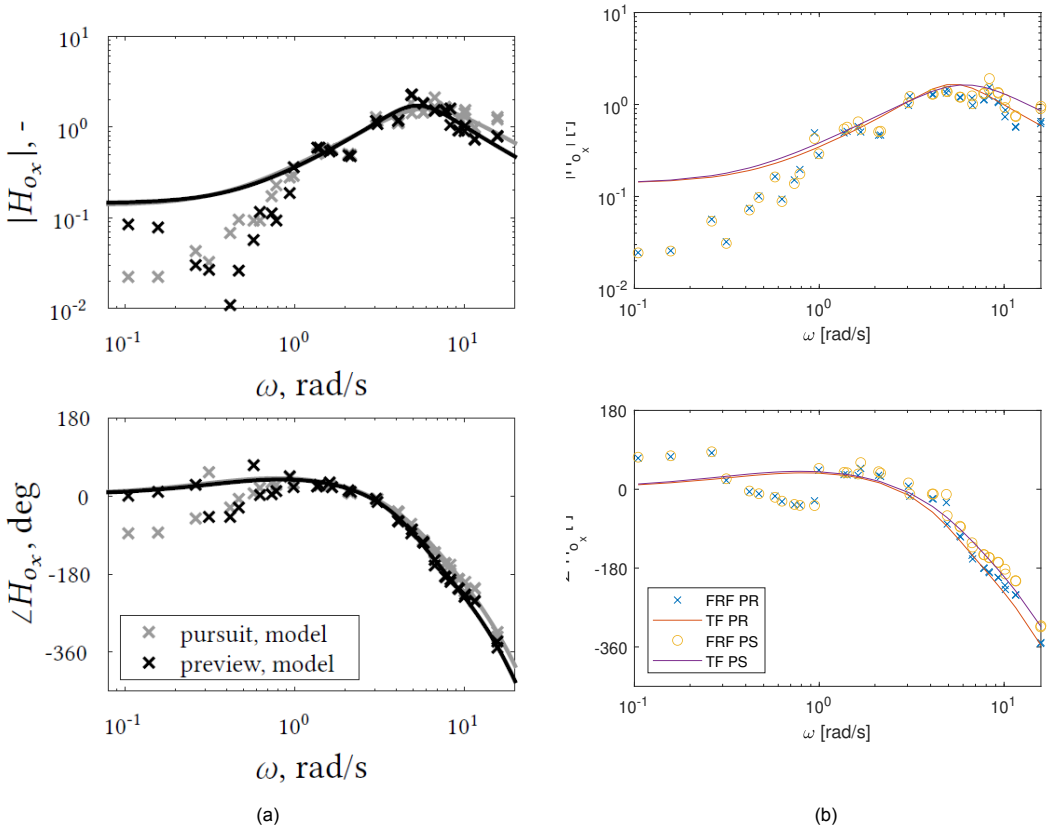


Figure 7.7: Comparison of the experimentally estimated FRFs and the resulting model fits of van der El (2018) (a) and the reproduced FRFs estimated from the simulated models (b) for the output response H_{o_x} for the DIPR experiment.

Figure 7.8 shows plots of the original- and interpolated magnitudes of the Fourier Coefficients of the control output $U(j\omega)$. This is done both ways: estimating the values at the disturbance frequencies by interpolating them from the target frequencies (Figure 7.8a) and vice versa (Figure 7.8b). The figures show that interpolation errors generally occur at the ends, where data are extrapolated, and around the peaks or transitional regions. At the outer edges, this is taken into account during the parameter estimation by van der El et al. (2016), where the outer data points receive a weight of 0 and are thus not used when estimating the parameters. The other errors are not corrected as well and this is one of the reasons why SSID might have an advantage. After all, if only one target function is required, interpolating is not needed and thus interpolation errors will not occur.
Besides at the ends, interpolation errors also appear in the middle of the frequency range, for example around $3rad/s$ in Figure 7.8a. Here, the error cannot occur due to extrapolation, as it is in the middle of the set of frequency points. Instead, the error is caused by the fact that complex numbers are interpolated. Figure 7.9 shows the phase of $U(j\omega)$ (which corresponds to the absolute values in Figure 7.8a). Figure 7.9 shows that around $3rad/s$, the phase goes from positive to negative. As linear interpolation is performed on complex numbers, this explains why relatively large errors are made for this step. This is further explained by Figure 7.10, which shows how the change in direction due to the periodic motion is missed by the linear interpolation. Such errors occur throughout the interpolation process, but in situations as highlighted by Figure 7.10 they become most obvious. It is therefore proposed to interpolate with respect to polar coordinates instead, although it should be verified whether this is an improvement.

7.4. Recommendations

The previous sections described a number of components of the simulation and showed test results obtained to verify correct implementation of these components. However, before the simulation can

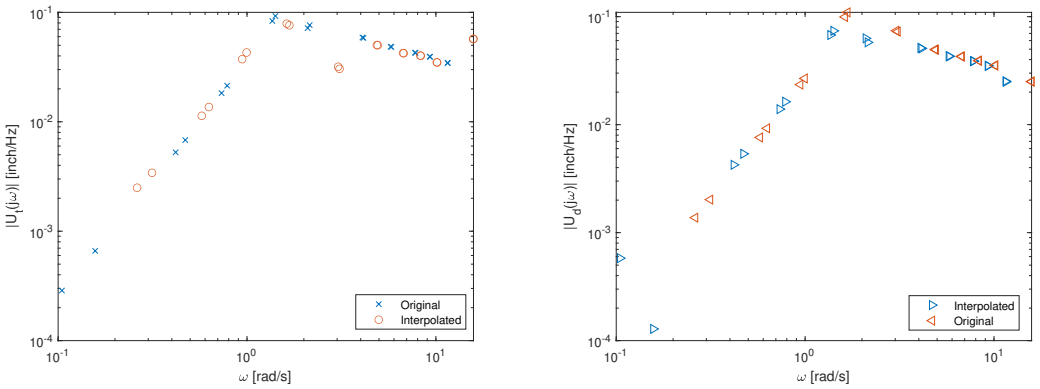


Figure 7.8: Absolute values of the Fourier transform of the control output $U(j\omega)$ at both: the original target frequencies ω_t interpolated at the disturbance frequencies ω_d (a) and vice versa (b).

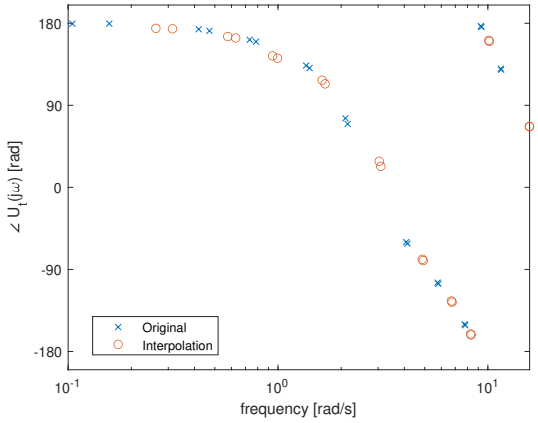


Figure 7.9: Phase angle between $-180°$ and $+180°$ for the control output $U(j\omega)$ at the original target frequencies ω_t interpolated at the disturbance frequencies ω_d (i.e., corresponding to Figure 7.8a).

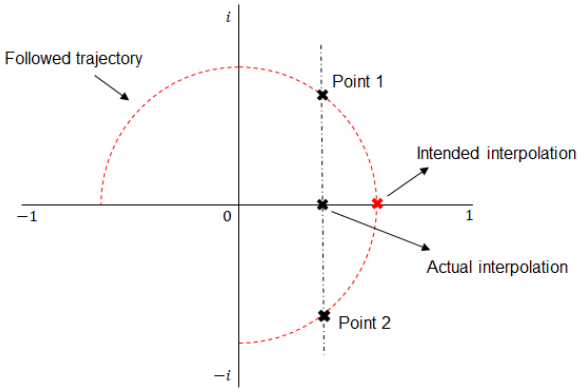


Figure 7.10: Schematic drawing to indicate the error caused by linearly interpolating complex numbers.

be used to answer the research questions posed in Section 5.1, some modifications/additions are required. This section will provide recommendations for these modifications. The recommendations that purely affect the simulation setup address both the domain in which the simulations are setup and the interconnections between the different sub-units. Furthermore, a number of recommendations are broader than just the simulation setup are discussed in Chapter 8.

The first recommendation regarding the simulation setup involves the domain of the simulation. It is recommended to transform the time domain Simulink setup to a frequency domain simulation. As convolution in the time domain becomes multiplication in the frequency domain, this transformation is expected to significantly decrease the computational times.
Another recommendation regarding the setup of the simulation involves the interconnection between the different elements of the simulation. With the initial setup of the simulation, the linkage between the different elements and the tests performed for these elements is not ideal. Integrating all elements into a proper single simulation tool, will improve the overview and reduce the chance of errors.

Besides these recommendations on the simulation setup itself, there are a number of improvements to be made that involve more than just the simulation setup. These recommendations are for example regarding the scoping of the research, the to be investigated sensitivities and the performance metrics. As they will influence more than just the simulation setup, they are addressed in the next chapter.

8

Future research

The previous three chapters have given insights about the approach that is used to answer the research questions posed in Chapter 5. However, a number of steps still need to be taken to answer them. These steps consider the scope of the research, the simulations that are to be performed and the experiment that is done to validate the method. However, besides the questions that are to be answered for this research, additional analyses can be performed. This chapter will therefore describe the next steps of this research.

8.1. Refining the research scope

Before starting the simulations to answer the first research questions, the exact tests need to be specified in more detail. Currently, the preliminary simulation are setup for four different tracking tasks, which are combinations of single integrator (SI) and double integrator (DI), and pursuit (PS) and preview (PR) tracking tasks: SIPS, SIPR, DIPS, DIPR. Although evaluating the performance of subsystem identification (SSID) regarding the presence of preview or the type of controlled element (CE) dynamics can be interesting at some point, it is not the main priority. More important are the performance of SSID regarding noise levels and forcing functions, or the sensitivity to different human operator (HO) model parameters. Therefore, this section will first select specific tracking task settings, which is followed by a brief elaboration on future decisions that are to be made.

8.1.1. Selecting specific tracking task settings

As other parts of the setup have a higher priority when investigating SSID, a reference tracking task should be defined for running the simulations. This means that the preview time and the controlled element dynamics will be fixed. When selecting a tracking task, it is beneficial to use a tracking task for which experiments and simulations have already been performed, as this provides a basis to verify and validate results.

Although there are numerous different pursuit- and preview tracking experiments that have been performed (see Chapter 2), it is logical to select a tracking task that is used in combination with either SSID or the pursuit- and preview model developed by van der El et al. (2016). Furthermore, as it was decided to evaluate the most advanced SSID technique developed by Sheffler et al. (2019), preview is required to allow for estimation of a feedforward time delay. Then, a number of tracking tasks remain, namely the ones used by van der El et al. (2016) on one hand, and by Sheffler et al. (2019) on the other. Note that, although other tracking tasks such as the ones chosen by van der El et al. (2018a) and van der El et al. (2020) also use the pursuit and preview model of van der El et al. (2016) (and would thus also be options), these are considered to be derivatives from the tracking tasks used by van der El et al. (2016). The settings and characteristics of the different tracking tasks are summarized in Table 8.1.

In first place, the table shows the exact CE dynamics and preview time that is used. Second, it provides an evaluation of some criteria that can be relevant when deciding on a specific tracking task. These criteria are defined as follows. First, the availability of HO model parameters evaluates whether model

parameters of the specific tracking task are available for reproduction. While this is the case for the results obtained by van der El et al. (2016) (see also Section 7.1.2), this does not hold for the ones from Sheffler et al. (2019). The second criteria evaluates the number of parameters that are to be identified for each tracking task, as was discussed in Section 6.4. Third, the relative contribution of the human remnant level to the control output $\frac{\sigma_{u_n}}{\sigma_u}$ is evaluated. This parameter, as mentioned in Section 7.1.3 and by van der El et al. (2020), can be relevant when considering a human-in-the-loop experiment. The last criteria evaluated is the degree to which the control task is realistic. Although all types of dynamics are encountered in real-life, single and double integrator dynamics are encountered more often than gain dynamics. Other studies such as (van der El et al., 2018a, 2020) have therefore also focused on SI and DI dynamics.

When deciding which of these settings should be used, it is desired to have a set of HO model parameters that actually approximates the real human controller. As these were not provided in the work of Sheffler et al. (2019), this combination of CE dynamics and preview time was not selected. To decide between the remaining types of dynamics, it was first argued that it would be desirable to estimate at least one of the time constants of the pre-filter subsystem (see Section 6.4). This means that either gain- or double integrator dynamics should be used. From these two, DI dynamics are preferred, as they are encountered more frequently in real-life. Furthermore, the human remnant has a larger effect, which means that a worst case scenario would be tested when performing a human-in-the-loop experiment. However, it is recommended to revisit this topic once the performance of SSID with respect to realistic human remnant levels is clarified. Nevertheless, it the DI dynamics settings as chosen by van der El et al. (2016) are currently selected for the analysis going forward.

Table 8.1: Different options for the tracking tasks to select as a reference condition. For each task, both the general settings and the evaluation with respect to specific trade-off criteria are displayed.

	Reference	(van der El et al., 2016)			(Sheffler et al., 2019)
	Dynamics type	G	SI	DI	quasi-DI
Settings	CE dynamics	0.8	$\frac{1.5}{s}$	$\frac{5}{s^2}$	$\frac{3.2(s+2.2)}{(s+1.6)(s2+3.6s+4)}$
	Preview time [s]	1	1	1	1
Evaluation	HO model parameters available	+	+	+	-
	# of unknown parameters	6	5	6	6
	Human remnant level $\frac{\sigma_{u_n}}{\sigma_u}$	0.1	0.35	0.6	0.6
	Realistic control case	+/-	+	+	+

8.1.2. Refining the set of independent variables

Besides selecting settings for the tracking task, the specific values for the independent variables need to be selected. These independent variables are the parameters as discussed in Section 5.2, such as the level of noise, the disturbance function or the HO model parameters. Overall, it is important to keep all settings constant when evaluating the effect of an independent variable.

Regarding the noise levels, it is desired to test the performance of SSID regarding various noise levels to identify how this performance deteriorates with increasing noise levels. When selecting the actual noise levels, it is important to select them in such a way that they can be compared to human remnant levels found in actual human-in-the-loop experiments, as explained in Section 7.1.3.

For the disturbance function, it is at first sight important that the multi-loop Fourier Coefficients (FC) methods and SSID require a different number of forcing functions. A first step in the analysis would be to simply omit the disturbance function in the simulations for SSID. However, this means that both methods will have a different amount of data points to perform the cost function minimization. While this could be assumed when evaluating the effects of different noise intensities in the first step, it can be desired to evaluate the effects of this assumption in a second step. When this means that the forcing

functions require modifications, attention should be paid to forcing function design and aspects such as the signal bandwidth, the crest factors and whether the forcing function still appears randomly.
Finally, regarding the HO model parameters, variations are applied more easily. With respect to the set of basic parameters that is used, the sensitivities to relative parameter changes can be evaluated in terms of the change in the resulting frequency response function (FRF) and in terms of the resulting identified parameters.

Overall, it is important to clearly define how the independent variables are evaluated. While this subsection provided initial suggestions on how to do this, more elaborate analysis and argumentation is required before performing the actual tests.

8.2. Modifications to the simulation setup

Once the scope is refined, the selected models and tracking tasks can be implemented properly in the simulations. Although a preliminary simulation has been setup, it was concluded in Section 7.4 that some modifications and extensions are required. The main modification involves the environment of the simulations, which will be changed from a Simulink time domain setup to a MATLAB frequency domain simulation. It is expected that this will significantly speed up the simulations. Another modification, or rather extension, consists of increasing the interconnections between the different simulation elements. This will provide an improved overview and a reduced chance of errors. Lastly, the model structure for the simulations should be evaluated. As the SSID is applied to the pre-filter model structure, it might be desirable to generate the data using the same model structure for consistency. However, as this is not necessary, it can be decided to keep the simulation structure identically. After all, the simulation and identification structure do not have to be the same.

Depending on the results obtained and the (updated) predicted relevance with which this is done, the effect of different CE dynamics or preview times might be evaluated. However, this is something that receives a low priority.

8.3. Running simulations and performing analysis

The next step in answering the research questions involves running the simulations and analyzing the results. First, the performance of SSID regarding the noise levels will be evaluated, followed by the forcing functions. To run the simulations, the exact noise levels and disturbance functions for the tests need to be defined. During the tests, the candidate pool densities also play an important role, as was seen in Chapter 4. However, considering the fact that the model parameters that need to be estimated are intuitive, the process of defining suitable ranges is simplified. Furthermore, the performance metrics still need to be determined before running the simulations. Often applied statistics for measuring the performance in manual control simulation include the variance accounted for (VAF) and the coherence function (Zaal et al., 2009; Nieuwenhuizen et al., 2008).

After the initial tests to investigate the noise levels and the disturbance signals, the sensitivity of both the multi-loop FC method and SSID to the different HO model parameters is investigated. For this, it might already be valuable to graphically observe the sensitivities of the resulting closed loop (SSID) and subsystem (multi-loop FC) FRFs, to see how they are affected by changes of the HO parameters. The method proposed by Pasma et al. (2017) appears suitable for this analysis.

8.4. Comparing the methods with experimental data

The majority of the proposed research project uses simulations. The advantage of such simulations are the simplicity with which different settings can be tested and the fact that the estimated parameters are known. However, even though experimentally validated models are used, they will still differ compared to a real human controller for several reasons. One of those is for example the non-linear components of manual control, which are simply included in the human remnant. For example, Mulder et al. (2019) describe how the position of the target signal in pursuit displays influences the control strategy. Another reason is found in the previously discussed assumption of modelling the human as

a two-channel system (see Section 2.1.2). Such assumptions differ from reality and are thus not captured by the models. Therefore, a human-in-the-loop experiment can be performed to check how the method performs when applied in real-life.

The aim of the experiment will be to identify the model parameters using SSID and compare these to the results obtained with the multi-loop FC method. The techniques can be applied to results of a human-in-the-loop experiment with identical settings for all parameters except for the disturbance function. Where the multi-loop FC method requires two forcing functions (the target- and the disturbance function), SSID only requires a one. Therefore, two tasks will be evaluated: one with the selected tracking task settings (see Section 8.1.1) and two forcing functions and one with all the basic settings and only one forcing function. The simulations should already provide an indication of the difference between applying SSID and multi-loop FC identification, which can be confirmed by the experiment.

The experiment setup is then similar to that of the simulations to allow comparison between results. The same type of forcing functions will be used and the same type of tracking task is used. However, the simulation results with respect to different noise levels might provide a reason to select a different tracking task for the experiment. For example, when the performance of SSID is bad at high noise levels, it might be preferred to perform a tracking task with lower levels of the human remnant, such as gain- or single-integrator controlled element dynamics.
Independent of the selected dynamics, it can be decided to use the settings from a similar experiment such as performed by van der El et al. (2016, 2018b,a) or both (if the simulation settings use those experiment settings as well). If settings from the mentioned experiments are taken, the data could also be used to apply the multi-loop FC technique. Then, the experiment only needs to be performed for the case where only a single disturbance function is present. However, this might introduce a confound in the experiment, as different operators will perform the new task. Especially because the effort of doing the small test twice is not a problem, it is better to do them both. This should be carefully evaluated when setting up the experiment.

Overall, the human-in-the-loop experiment is thus fairly simple. A tracking task, similar or even identical to that of an already performed experiment will be selected and the only independent variable is the disturbance function. Then, the parameters identified using the two identification techniques can be compared. The differences found are expected to match the differences found in the simulations.

8.5. Other analyses

Besides evaluating SSID using simulations and experiments, the analytical properties such as the bias of the method can also be evaluated, as shown by van Lunteren (1979) for single loop tracking tasks and later by van Paassen (1994) for multi-loop tracking tasks. As SSID and multi-loop FC identification use either the closed-loop or the subsystem FRFs, these results provide a direct indication of the analytical biases introduced in both methods. An analysis similar to the one performed by Nieuwenhuizen et al. (2008) could be performed when investigating this in more detail. However, considering the many other aspects that are to be investigated, this is currently decided to be outside of the scope of this research project.

9

Conclusions

In the quest of constructing a universal model that captures the different strategies that humans apply when controlling systems, one of the aspects that requires better understanding is the anticipatory feedforward control. This research focuses on a recently developed frequency domain identification method called subsystem identification (SSID) that can identify both feedback- and feedforward subsystems. The research objective therefore reads as:

Contribute to the set of tools used to identify feedback- and feedforward behaviour in manual control cybernetics by exploring the capabilities and limitations of Subsystem Identification.

With this goal in mind, a literature study was performed and a research project was setup. This first part, the literature study, focused on both the state-of-the-art of manual control that involves both feedback- and feedforward behaviour and this newly developed SSID technique. From the state-of-the-art, it was concluded that both pursuit- and preview tracking tasks and tracking tasks with predictable target signals allow humans to use both feedback- and feedforward control behaviour. In models, the human operator (HO) is then often modelled as a two-channel system that responds either to the error- and the tracking signal (ET) or to the system output- and the target signal (XT). Especially the pursuit- and preview tracking tasks are relevant as the human control behaviour is well understood by the intuitive and experimentally validated model developed by van der El et al. (2016). Furthermore, this type of tracking task works well in combination with frequency domain techniques such as the well-studied multi-loop Fourier Coefficients (FC) technique and SSID, the topic of this research project.
Subsystem identification is a technique that uses a candidate pool approach and the closed-loop frequency response function (FRF) to estimate both the feedback- and feedforward subsystems. The fact that it only requires a single forcing function to identify multiple subsystems makes it unique with respect to similar techniques. Over the years, different versions of SSID have been developed, for which the last one is most relevant as it is able to estimate both feedback- and feedforward subsystems, including time delays. Although initial versions of the technique have demonstrated that SSID is able to successfully identify the parameters if the candidate pool is sufficiently dense and noise levels are sufficiently low, this has not been done with respect to realistic noise models that describe the non-linear behaviour of human operators. Therefore, it is unclear to what extent the technique is able to properly identify human operator model parameters. Furthermore, the technique appears computationally expensive.

Based on this literature study, a research project was setup to evaluate how SSID performs compared to the current state-of-the-art. This performance is evaluated with respect to realistic noise levels (resembling the human remnant), forcing functions and variations in the human operator parameters, as these three points address the advantages and possible limitations of the technique. This will be evaluated by performing simulations and a small human-in-the-loop experiment. To provide a reference, the multi-loop FC technique is used, as this well-studied technique is closely related to SSID. Furthermore, the analysis will be performed for a pursuit- and preview tracking task and the preview model (van der El et al., 2016), for which SSID appears to be capable of identifying the parameters.

In order to perform the tests, an initial simulation has been setup. For the setup of this Simulink simulation different parts such as forcing function design, HO model implementation, remnant modelling and multi-loop FC identification implementation were constructed and verified. With this, all main simulation components have been constructed. Furthermore, a preliminary plan has been made for an experiment. The next steps of the research consist of modifying the simulation setup and answering the research questions by performing the simulations. Consequently, the human-in-the-loop experiment can be performed and evaluated.

Bibliography

Damveld, H. J., Beerens, G. C., van Paassen, M. M., and Mulder, M. (2010). Design of Forcing Functions for the Identification of Human Control Behavior. *Journal of Guidance, Control, and Dynamics*, 33(4):1064–1081.

Drop, F. M. (2016). *Control-Theoretic Models of Feedforward in Manual Control*. PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering.

Drop, F. M., Pool, D. M., Damveld, H. J., van Paassen, M. M., and Mulder, M. (2013). Identification of the Feedforward Component in Manual Control With Predictable Target Signals. *IEEE Transactions on Cybernetics*, 43(6):1936–1949.

Drop, F. M., Pool, D. M., Mulder, M., and Bülthoff, H. H. (2016). Constraints in Identification of Multi-Loop Feedforward Human Control Models. In *Proceedings of the 13th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, Kyoto, Japan*.

Drop, F. M., Pool, D. M., van Paassen, M. M., Mulder, M., and Bülthoff, H. H. (2018). Objective Model Selection for Identifying the Human Feedforward Response in Manual Control. *IEEE Transactions on Cybernetics*, 48(1):2–15.

Drop, F. M., Pool, D. M., van Paassen, M. M., Mulder, M., and Bülthoff, H. H. (2019). Effects of Target Signal Shape and System Dynamics on Feedforward in Manual Control. *IEEE Transactions on Cybernetics*, 49(3):768–780.

Duarte, R., Pool, D. M., van Paassen, M. M., and Mulder, M. (2017). Experimental Scheduling Functions for Global LPV Human Controller Modeling. *IFAC-PapersOnLine*, 50(1):15853 – 15858. 20th IFAC World Congress.

Hess, R. A. (1965). The Human Operator as and Element in a Control System with Time Varying Dynamics. Master's thesis, University of California, Davis.

Hess, R. A. (1981). Pursuit Tracking and Higher Levels of Skill Development in the Human Pilot. *IEEE Transactions on Systems, Man, and Cybernetics*, 11:262–273.

Hess, R. A. and Modjtahedzadeh, A. (1989). A Preview Control Model of Driver Steering Behavior. In *Proceedings of the IEEE International Conference on System, Man and Cybernetics, Cambridge, MA, USA*, volume 2, page 504–509.

Ito, K. and Ito, M. (1975). Tracking Behavior of Human Operators in Preview Control Systems. *Electrical Engineering in Japan (English translation of Denki Gakkai Ronbunshi)*, 95(1):120–127.

Koushkbaghi, S., Wang, S., Seigler, M. T., and Hoagg, J. B. (2018). On the Control Strategies that Humans use to Interact with Linear Systems with Output nonlinearities. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3732–3737.

Krendel, E. S. and McRuer, D. T. (1960). A Servomechanics Approach to Skill Development. *Journal of the Franklin Institute*, 269(1):24–42.

Laurense, V. A., Pool, D. M., Damveld, H. J., van Paassen, M. M., and Mulder, M. (2015). Effects of Controlled Element Dynamics on Human Feedforward Behavior in Ramp-Tracking Tasks. *IEEE Transactions on Cybernetics*, 45(2):253–265.

Levison, W. H., Baron, S., and Kleinman, D. L. (1969). A Model for Human Controller Remnant. *IEEE Transactions on Man-Machine Systems*, 10(4):101–108.

Ljung, L. (1999). *System Identification Theory for the User*. Prentice Hall, Inc., second edition.

Matveeva, F., Mousavi, S. A. S., Zhang, X., Seigler, T. M., and Hoagg, J. B. (2016). On the Effects of Changing Reference Command as Humans Learn to Control Dynamic Systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1211–1216.

McRuer, D. T., Allen, R. W., Weir, D. H., and Klein, R. H. (1977). New Results in Driver Steering Control Models. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 19(4):381–397.

McRuer, D. T., Graham, D., Krendel, E. S., and Reisener, W. J. (1965). Human Pilot Dynamics in Compensatory Systems: Theory, Models, and Experiments with Controlled Element and Forcing Function Variations. Technical Report AFFDL-TR-65-15, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base (OH).

McRuer, D. T., Hofmann, L. G., Jex, H. R., Moore, G. P., Phatak, A. V., Weir, D. H., and Wolkovitch, J. (1968). New Approaches to Human-Pilot/Vehicle Dynamic Analysis. Technical Report AFFDL-TR-67-150, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base (OH).

McRuer, D. T. and Jex, H. R. (1967). A Review of Quasi-Linear Pilot Models. *IEEE Transactions on Human Factors in Electronics*, HFE-8(3):231–249.

Mulder, M., Pool, D. M., Abbink, D. A., Boer, E. R., Zaal, P. M. T., Drop, F. M., van der El, K., and van Paassen, M. M. (2018). Manual Control Cybernetics: State-of-the-Art and Current Trends. *IEEE Transactions on Human-Machine Systems*, 48(5):468–485.

Mulder, M., Pool, D. M., van der El, K., Drop, F. M., and van Paassen, M. M. (2019). Manual Control with Pursuit Displays: New Insights, New Models, New Issues. In *Proceedings of the 14th IFAC Symposium on Analysis Design and Evaluation of Human Machine Systems Tallinn, Estonia*, pages 139–144.

Nieuwenhuizen, F. M., Zaal, P. M. T., Mulder, M., van Paassen, M. M., and Mulder, J. A. (2008). Modeling Human Multichannel Perception and Control Using Linear Time-Invariant Models. *Journal of Guidance, Control, and Dynamics*, 31(4):999–1013.

Pasma, J. H., Boonstra, T. A., van Kordelaar, J., Spyropoulou, V. V., and Schouten, A. C. (2017). A Sensitivity Analysis of an Inverted Pendulum Balance Control Model. *Frontiers in Computational Neuroscience*, 11:99.

Pew, R. W., Duffendack, J. C., and Fensch, L. K. (1967). Sine-Wave Tracking Revisited. *IEEE Transactions on Human Factors in Electronics*, HFE-8(2):130–134.

Popovici, A., Zaal, P. M. T., and Pool, D. M. (2017). Dual Extended Kalman Filter for the Identification of Time-Varying Human Manual Control Behavior. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference, Denver (CO)*, number AIAA-2017-3666.

Rasmussen, J. (1983). Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(3):257–266.

Shadmehr, R. and Mussa-Ivaldi, F. A. (1994). Adaptive Representation of Dynamics During Learning of a Motor task. *Journal of Neuroscience*, 14(5):3208–3224.

Sheffler, A. J. S., Mousavi, S. A. S., Hellström, E., Jankovic, M., Santillo, M. A., Seigler, T. M., and , H. J. B. (2019). Effects of Reference-Command Preview as Humans Learn to Control Dynamic Systems. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 4199–4204.

Sheridan, T. B. (1966). Three Models of Preview Control. *IEEE Transactions on Human Factors in Electronics*, 7(2):91–102.

Stapleford, R. L., McRuer, D. T., and Magdaleno, R. E. (1967). Pilot Describing Function Measurements in a Multiloop Task. *IEEE Transactions on Human Factors in Electronics*, 8(2):113–125.

van der El, K. (2018). *How Humans Use Preview Information in Manual Control*. PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering.

van der El, K., Padmos, S., Pool, D. M., van Paassen, M. M., and Mulder, M. (2018a). Effects of Preview Time in Manual Tracking Tasks. *IEEE Transactions on Human-Machine Systems*, 48(5):486–495.

van der El, K., Pool, D. M., Damveld, H. J., van Paassen, M. M., and Mulder, M. (2016). An Empirical Human Controller Model for Preview Tracking Tasks. *IEEE Transactions on Cybernetics*, 46(11):2609–2621.

van der El, K., Pool, D. M., and Mulder, M. (2019). Analysis of Human Remnant in Pursuit and Preview Tracking Tasks. In *Proceedings of the 14th IFAC Symposium on Analysis Design and Evaluation of Human Machine Systems Tallinn, Estonia*, pages 145–150.

van der El, K., Pool, D. M., van Paassen, M. M., and Mulder, M. (2018b). Effects of Preview on Human Control Behavior in Tracking Tasks With Various Controlled Elements. *IEEE Transactions on Cybernetics*, 48(4):1242–1252.

van der El, K., Pool, D. M., van Paassen, M. M., and Mulder, M. (2020). Effects of Target Trajectory Bandwidth on Manual Control Behavior in Pursuit and Preview Tracking. *IEEE Transactions on Human-Machine Systems*, 50(1).

van Grootheest, A., Pool, D. M., van Paassen, M. M., and Mulder, M. (2018). Identification of Time-Varying Manual Control Adaptations with Recursive ARX Models. In *2018 AIAA Modeling and Simulation Technologies Conference*.

van Lunteren, A. (1979). *Identification of Human Operator Describing Function Models with One or Two Inputs in Closed Loop Systems*. PhD thesis, Delft University of Technology, Faculty of Mechanical Engineering.

van Paassen, M. M. (1994). *Biophysics in Aircraft Control, A Model of the Neuromuscular System of the Pilot's Arm*. PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering.

van Paassen, M. M. and Mulder, M. (1998). Identification of Human Operator Control Behaviour in Multiple-Loop Tracking Tasks. In *Proceedings of the Seventh IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, Kyoto Japan*, pages 515–520. Pergamon.

van Wingerden, J.-W. and Verhaegen, M. (2009). Subspace Identification of Bilinear and LPV Systems for open- and Closed-loop data. *Automatica*, 45(2):372 – 381.

Vos, M. C., Pool, D. M., Damveld, H. J., van Paassen, M. M., and Mulder, M. (2014). Identification of Multimodal Control Behavior in Pursuit Tracking Tasks. In *Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics, San Diego (CA)*, pages 69–74.

Wasicko, R. J., McRuer, D. T., and Magdaleno, R. E. (1966). Human Pilot Dynamic Response in Single-loop Systems with Compensatory and Pursuit Displays. Technical Report AFFDL-TR-66-137, Air Force Flight Dynamics Laboratory.

Wiener, N. (1965). *Cybernetics, Second Edition: Or the Control and Communication in the Animal and the Machine*. The MIT Press.

Wolpert, D. M., Miall, R. C., and Kawato, M. (1998). Internal Models in the Cerebellum. *Trends in Cognitive Sciences*, 2(9):338 – 347.

Yamagami, M., Howell, D., Roth, E., and Burden, S. A. (2019). Contributions of Feedforward and Feedback Control in a Manual Trajectory-Tracking task. In *Proceedings of the 2nd IFAC Conference on Cyber-Physical & Human-Systems, Miami (FL)*, volume 51 of *IFAC-PapersOnLine*, pages 61–66.

Yamashita, T. (1989). Precognitive Behavior in Tracking of Targets with 2 sine Waves. *Japanese Psychological Research*, 31(1):20–28.

Yamashita, T. (1990). Effects of Sine Wave Combinations on the Development of Precognitive Mode in Pursuit Tracking. *The Quarterly Journal of Experimental Psychology*, 42A(4):791–810.

Yu, B., Gillespie, B. R., Freudenberg, J. S., and Cook, J. A. (2014). Human Control Strategies in Pursuit Tracking with a Disturbance Input. In *53rd IEEE Conference on Decision and Control*, pages 3795–3800.

Zaal, P. M. T., Pool, D. M., Chu, Q. P., van Paassen, M. M., Mulder, M., and Mulder, J. A. (2009). Modeling Human Multimodal Perception and Control Using Genetic Maximum Likelihood Estimation. *Journal of Guidance, Control, and Dynamics*, 32(4):1089–1099.

Zhang, X. and Hoagg, J. B. (2016a). Frequency-Domain Subsystem Identification with Application to Modeling Human Control Behavior. *Systems & Control Letters*, 87.

Zhang, X. and Hoagg, J. B. (2016b). Subsystem Identification of Multivariable Feedback and Feedforward Systems". *Automatica*, 72:131 – 137.

Zhang, X., Wang, S., Hoagg, J. B., and Seigler, T. M. (2018). The Roles of Feedback and Feedforward as Humans Learn to Control Unknown Dynamic Systems. *IEEE Transactions on Cybernetics*, 48(2):543–555.