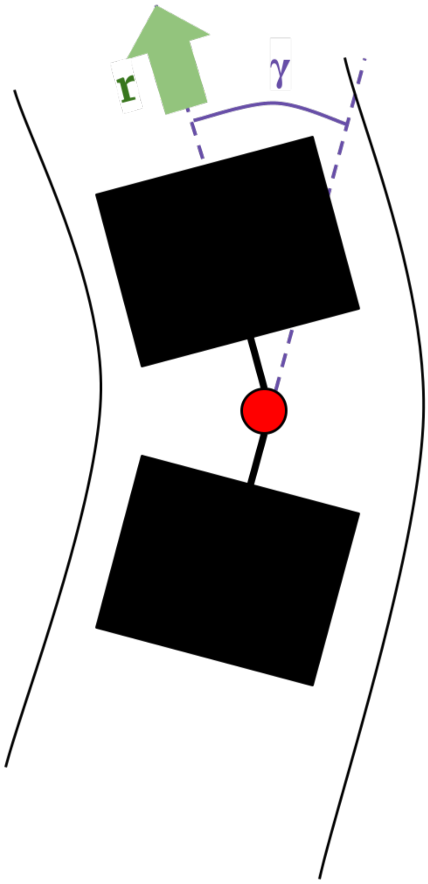

Standstill Auto-Tuning of ADRC for Electro-Hydraulic Articulated Steering

Master's thesis in Systems, Control and Mechatronics (MPSYS)

NIKLAS LEANDER LUCA TIMPE

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Standstill Auto-Tuning of ADRC for Electro-Hydraulic Articulated Steering

Niklas Leander Luca Timpe

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2026

Standstill Auto-Tuning of ADRC for Electro-Hydraulic Articulated Steering

Niklas Leander Luca Timpe

Supervisor: Jonas Sjöberg, Electrical Engineering
External Supervisor: Gustaf Blomberg
Examiner: Jonas Sjöberg

Master's Thesis 2026
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Sketch of an articulated steering vehicle following its reference angle, viewed from the top. The reference is shown as an arrow and the articulation angle as gamma (γ).

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Standstill Auto-Tuning of ADRC for Electro-Hydraulic Articulated Steering

NIKLAS LEANDER LUCA TIMPE
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Manual tuning of the steering controller for an autonomous articulated vehicle is time-consuming and scales poorly across vehicle variants. This thesis proposes an automated pipeline that identifies a small set of plant parameters and uses them to configure an error-based Active Disturbance Rejection Control (ADRC) law, without manual calibration. The steering plant is described by three blocks, an input deadband, a second- order linear time-invariant (LTI) core, and a mechanical output saturation. The pipeline runs three steps in sequence at vehicle standstill. First, the input deadband width is identified offline using a breakpoint grid search, and then statically compensated. Second, a step is applied to the plant and a nonlinear least-squares estimator fits the gain K and time constant τ of the LTI core to the recorded response. Third, an empirical tuning rule built on top of bandwidth parameterisation and half-gain tuning turns K and τ into the ADRC controller and Extended State Observer gains, after which the observer estimates and cancels unmodelled dynamics online during operation. The pipeline is verified on three test environments, a set of second-order plants with known parameters, an OpenModelica multi-body simulation model of the vehicle, and the real articulated vehicle as the final end-to-end check.

Within the simulation environment, nine vehicle configurations span differences in chassis mass, cylinder bore, ground friction, valve flow capacity, and linkage geometry. Across these configurations the auto-tuned controllers achieved low tracking error, maintained control smoothness under measurement noise, and absorbed the shift in plant gain produced by payload and ground-friction changes away from the standstill operating point at which the parameters were identified.

On the real vehicle the identification stages ran end to end and returned a deadband and gain comparable to the simulated family, but the closed loop did not track the reference sweep. The plant model omits a transport delay, assumed negligible, which leaves the auto-tuned bandwidth too high for the real plant, and a measured deadband hysteresis the model also omits adds to the gap. Lowering the bandwidth helps but does not close it, and accounting for these effects is left as future work.

Keywords: active disturbance rejection control, ADRC, electro-hydraulic steering, articulated vehicle, automatic tuning, extended state observer, bandwidth parameterization, autonomous driving.

Acknowledgements

I would like to sincerely thank Gustaf Blomberg and Anneli Pettersson for their support and the trust they placed in me throughout this project. I truly appreciate that you gave me the freedom to work autonomously, while always being ready to lend a guiding hand whenever I ran into difficult situations. Working on this thesis and collaborating with both of you has been a lot of fun.

Niklas Leander Luca Timpe, Gothenburg, September 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ADRC	Active Disturbance Rejection Control
DOF	Degree of Freedom
ESO	Extended State Observer
HFR	High-Frequency energy Ratio
LTI	Linear Time-Invariant
MSL	Modelica Standard Library
NLS	Nonlinear Least Squares
NRMSE	Normalised Root-Mean-Square Error
OCT	Open-Center Tandem
RMSE	Root-Mean-Square Error
SISO	Single-Input Single-Output
ZOH	Zero-Order Hold

Nomenclature

Below are the symbols used throughout this thesis, grouped by role.

Signals

$u(t)$	Control input (valve solenoid current)
$y(t)$	Plant output
$r(t)$	Reference signal
$e(t) = r(t) - y(t)$	Tracking error
$d(t)$	External disturbance input
γ	Articulation angle
$\dot{\gamma}$	Articulation angular velocity

Plant Parameters

K	Plant gain
τ	Dominant plant time constant
b_0	ADRC critical gain (nominal plant input gain)
$\Delta b = b - b_0$	Plant input-gain mismatch
b_d	Disturbance input gain

Controller and Observer

N	ADRC model order (1 or 2); reused below for the NLS sample count
k_1, k_2	Continuous-time state-feedback controller gains
$k_{\mathrm{d},1}, k_{\mathrm{d},2}$	Discrete-time controller gains (vector $\boldsymbol{k}_\mathrm{d}$)
l_1, l_2, l_3	Continuous-time extended state observer gains

$l_{\mathrm{d},1}, l_{\mathrm{d},2}, l_{\mathrm{d},3}$	Discrete-time observer gains (vector $\boldsymbol{l}_\mathrm{d}$)
ω_CL	Closed-loop bandwidth
ω_O	Observer bandwidth
k_ESO	Observer-to-controller bandwidth ratio, $\omega_\mathrm{O} = k_\mathrm{ESO}\,\omega_\mathrm{CL}$
$f(t)$	Output-domain total disturbance
$f^*(t)$	Error-domain total disturbance
$\hat{x}_1, \hat{x}_2, \hat{x}_3$	ESO state estimates (error, error-rate, total disturbance)
u_lim	Saturated control signal fed back to the observer
$u_\mathrm{min}, u_\mathrm{max}$	Lower and upper actuator saturation limits (distinct from the step amplitude u_max in Plant Identification below)

Deadband Identification and Compensation

d	True deadband half-width
$\hat{d}$	Identified deadband half-width
g	Gradient-zone half-width of the practical inverse
V_max	Absolute valve-current limit
u_comp	Compensator input (normalised controller output)
u_valve	Valve current after the inverse map
u_LTI	Useful actuation signal entering the LTI plant
α	Ramp-excitation slope used for deadband identification
$t_d = d/\alpha$	Time at which the ramp crosses the deadband boundary
N_base	Number of leading ramp samples used to estimate μ_0, σ_0 (threshold detector)
μ_0, σ_0	Mean and standard deviation of the baseline noise band (threshold detector)
k_σ	Threshold multiplier on σ_0 in the threshold detector
k	Segmented-regression breakpoint index
c	Constant level fitted to the dead-region segment
a_0, a_1, a_2	Polynomial coefficients fitted to the active-region segment

Plant Identification (NLS Step Experiment)

u_max	Step excitation amplitude (distinct from the actuator saturation limit u_max above)

$\varphi(t_i)$	Sampled angle measurements
$J(K, \tau)$	Least-squares cost
$(\hat{K}, \hat{\tau})$	NLS parameter estimates
N	Number of identification samples (distinct from the ADRC model order N above)

Discrete-Time Implementation

T_s	Controller sample time
k	Discrete time index
z	z-transform variable
$z_\mathrm{CL} = e^{-\omega_\mathrm{CL} T_\mathrm{s}}$	Closed-loop pole in the z-domain
$z_\mathrm{ESO} = e^{-k_\mathrm{ESO} \omega_\mathrm{CL} T_\mathrm{s}}$	Observer pole in the z-domain
$\boldsymbol{A}_\mathrm{d}, \boldsymbol{b}_\mathrm{d}$	ZOH-discretised plant matrices
$\boldsymbol{A}_\mathrm{ESO}, \boldsymbol{b}_\mathrm{ESO}$	Discrete current-observer matrices

Performance Metrics and Excitation

NRMSE_k	Normalised RMSE on the k-th reference half-period
NRMSE	Mean NRMSE over all half-periods of the sweep
HFR	High-frequency energy ratio of the control signal
M	Number of reference half-periods in the sweep
W_k	Reference half-period window $[t_k, t_{k+1}]$
$U(f)$	Discrete Fourier transform of u over the run
f_c	High-frequency cutoff used in HFR
f_Nyq	Nyquist frequency at the controller sample rate
$f_\mathrm{ref,max}$	Maximum reference sweep frequency
σ	Measurement-noise standard deviation

Vehicle Set and Geometry

$\mathcal{V}$	Set of vehicles targeted by the methodology
$v \in \mathcal{V}$	Individual vehicle satisfying every requirement of $\mathcal{V}$
$V_0, \ldots, V_8$	Nine simulation vehicles used as validation instances
$[\gamma_\mathrm{min}, \gamma_\mathrm{max}]$	Mechanical articulation range

Simulation Model

x, y	Global position of the rear-body centre of mass
ϕ_r	Rear-body orientation in the global frame
v_x, v_y	Body-frame translational velocities
ω	Body-frame yaw rate
m_f, m_r	Front- and rear-body masses
Q_p	Pump volumetric flow
$Q_{v,\mathrm{nom}}$	Valve nominal flow rating
D_c	Cylinder bore diameter
F_x, F_y, T_z	Planar friction forces and yaw torque applied at each body's CoM
$\mu_x, \mu_y, \mu_{\mathrm{rot}}$	Longitudinal, lateral, and rotational friction coefficients
v_0, w_0	Translational and rotational tanh-regularisation speeds
r_{eff}	Effective lever arm for the yaw friction torque
g	Gravitational acceleration

Contents

List of Acronyms **ix**

Nomenclature **xi**

List of Figures **xix**

List of Tables **xxiii**

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Contributions . . . 3
- 1.3 Thesis Outline . . . 4

2 Active Disturbance Rejection Control **7**
- 2.1 Plant Model and Total Disturbance . . . 7
- 2.2 Error-Based ADRC . . . 8
  - 2.2.1 Error Dynamics . . . 8
  - 2.2.2 Extended State Observer . . . 9
  - 2.2.3 Control Law . . . 10
- 2.3 Bandwidth Parameterization . . . 11
  - 2.3.1 Controller Tuning . . . 11
  - 2.3.2 Observer Tuning . . . 11
  - 2.3.3 Half-Gain Tuning . . . 11
- 2.4 Practical Discrete-Time Error-Based ADRC . . . 12
  - 2.4.1 Controller . . . 12
  - 2.4.2 Observer . . . 13
  - 2.4.3 Controller Output Limitation and Anti-Windup . . . 14

3 Methodology for Auto-Tuning Pipeline **15**
- 3.1 System Description and Vehicle Set . . . 15
  - 3.1.1 Architectural requirements . . . 15
  - 3.1.2 Modelling assumptions . . . 16
- 3.2 Plant Model for Controller Design . . . 16
  - 3.2.1 Expected Closed-Loop Structure . . . 16
  - 3.2.2 Constant Deadband Approximation . . . 17
  - 3.2.3 Second-Order LTI Core . . . 18

3.2.4 Output Saturation . . . 19
3.3 Test Environments . . . 19
3.3.1 Synthetic system . . . 19
3.3.2 Simulation model . . . 19
3.3.3 Real vehicle . . . 20
3.4 Deadband Identification . . . 20
3.4.1 Deadband Compensation . . . 20
3.4.2 Identifying the half-width . . . 23
3.4.2.1 Experiment . . . 23
3.4.2.2 Identification algorithm . . . 23
3.4.2.3 Algorithm comparison . . . 24
3.5 Controller Tuning . . . 26
3.5.1 Analytic Tuning of b_0 . . . 26
3.5.2 Empirical Tuning of ω_{CL} and k_{ESO} . . . 27
3.5.2.1 Reference Signal . . . 27
3.5.2.2 Metrics . . . 28
3.5.2.3 Parameter Sweeps . . . 29
3.6 Auto-Tuning and Plant Identification . . . 34
3.6.1 Plant Identification . . . 34
3.6.2 Simulation Model Verification . . . 35
3.7 Deliberate Simulation Model Mismatch Verification . . . 37
3.7.1 Mismatch setup . . . 37
3.7.2 Results . . . 38

4 Real-Vehicle Deployment **41**
4.1 Test conditions . . . 41
4.2 Deadband identification . . . 41
4.3 Plant identification . . . 42
4.4 Closed-loop sweep test . . . 43
4.5 Diagnosis . . . 44
4.5.1 A hidden transport delay . . . 44
4.5.2 Reproducing the failure on a synthetic plant . . . 45
4.5.3 Detuning the controller on the vehicle . . . 46
4.5.4 Deadband hysteresis and the sensor . . . 47

5 Discussion and Conclusion **49**
5.1 Achievements and limitations . . . 49
5.2 Why the real-vehicle deployment fell short . . . 49
5.3 What the simulation evidence establishes . . . 50
5.4 Future work . . . 51
5.5 Conclusion . . . 51

Bibliography **53**

A Simulation Model Details **I**
A.1 Model Simplifications . . . I
A.2 Validation Instances . . . II

A.3 Multi-Body Dynamics IV
A.3.1 Coordinate System and Degrees of Freedom IV
A.3.2 Body Geometry and Inertial Properties IV
A.3.3 Joints and Kinematic Constraints IV
A.3.4 Friction Model IV
A.4 Hydraulic System V
A.4.1 Hydraulic System Dynamics V
A.4.2 Force Transmission to Vehicle Bodies VI
A.5 External Control System VI

B Deadband Identification Algorithm **VII**
B.1 Exact ramp response VII
B.2 Grid-search objective VII
B.3 Alternative variants VIII

C Controller-Tuning Sweep Details **IX**
C.1 NRMSE threshold IX
C.2 HFR threshold X
C.3 First-order tracking XII

List of Figures

1.1 Schematic of an articulated vehicle. The articulation angle γ between the front and rear bodies sets the direction of travel and is the controlled output throughout this thesis, which should track the commanded reference r. The angle is actuated by a pair of hydraulic cylinders driven through an open-center tandem proportional directional valve. 2

1.2 Methodology and verification flow of the auto-tuning pipeline, read top to bottom along the left column, where each box is a stage developed in Chapter 3. Each coloured branch to the right is the develop-and-verify loop for that stage, run across the test environments. Straight arrows give the flow direction, curved arrows are iteration loops back to an earlier step, and a filled black circle marks the end of a stage's development loop. 5

2.1 Plant model assumed by a second-order ADRC, equation (2.3): a double integrator driven through the input gain b_0, with all unmodelled dynamics and disturbances lumped into the single total disturbance f. Adapted from [11]. 8

2.2 Complete error-based ADRC, equations (2.10) and (2.12). The red region is the Extended State Observer, which estimates the error states and the total disturbance, and the surrounding green region is the ADRC that builds the control signal from those estimates and rejects the disturbance. Adapted from [11]. 10

3.1 The simplified steering plant of this thesis, in closed loop with the error-based ADRC controller of Chapter 2. The control signal $u(t)$ is given by (2.13) and the output is the articulation angle $\gamma(t) = y(t)$. This is the model the rest of the methodology is designed against. . . 17

3.2 Flow to ports A and B as a function of spool stroke for several pump flow rates, for the test valve of [1]. The colored overlays added by the author indicate the three behavioral regions identified in the source paper: recirculation only (blue), transition (red), and full opening (green). The blue region is the one approximated as a constant deadband in this work. Adapted from [1]. 18

3.3 The deadband compensator inserted ahead of the plant model of Figure 3.1. The error-based ADRC, drawn in Figure 3.1, now produces the normalized command u_{comp}, the compensator converts it into the valve current u_{valve} that jumps over the constant deadband (Section 3.2.2), and the useful actuation u_{LTI} enters the LTI core. The output saturation of Figure 3.1 is omitted here to focus on the deadband. 21
3.4 The practical compensator of (3.5). The linear in-window gradient replaces the discontinuous jump of the ideal inverse and leaves a residual deadband of half-width g. 23
3.5 Results of the A-pos deadband identification on the nine Modelica vehicle configurations, with the identified half-width $\hat{d}$ marked on each. The identification was run with measurement noise of $\sigma = 0.002\,\text{rad}$. . 26
3.6 Sine-sweep reference used for all closed-loop tuning experiments. Its amplitude is reduced as the frequency rises, so the peak steering rate is highest at the slow start of the sweep and lowest at the fast end. . 27
3.7 Noise-free sweep of the 2nd-order ADRC over the closed-loop bandwidth, one curve per plant time constant τ in the synthetic family. Each curve is the mean over $k_{\text{ESO}} \in [3, 5]$ and the shaded band is ± 1 standard deviation across k_{ESO}. 29
3.8 Sweep of the 2nd-order ADRC over ω_{CL} with measurement noise and half-gain tuning, $k_{\text{ESO}} \in [3, 5]$, and $\tau \in [0.2, 2.0]\,\text{s}$. The coloured crosses mark the ω_{CL} at which each τ first reaches the tracking anchor NRMSE $= 0.5$, the circles the ω_{CL} at which it first reaches the smoothness anchor HFR $= 0.01$ (Appendix C). 31
3.9 The $(\tau, \omega_{\text{CL}})$ operating points where the HFR first reaches 0.01, with the fitted second-order polynomial (3.15) that becomes the 2nd-order tuning rule. 32
3.10 Sweep of the 1st-order ADRC over ω_{CL} with measurement noise and half-gain tuning on the fast plants ($\tau \leq 0.6\,\text{s}$), $k_{\text{ESO}} \in [3, 5]$. The shaded band is ± 1 standard deviation across k_{ESO}. 33
3.11 Relative error of the nonlinear least-squares estimator (3.18) against measurement noise, for several plant gains K and time constants τ. . 35
3.12 Closed-loop tracking response of the auto-tuned controllers on the nine simulation vehicles of Table A.3. 37
3.13 Closed-loop tracking response of the mismatched simulations of Table 3.5, overlaid for all nine vehicles. 39

4.1 Deadband identification on the real vehicle. Raw ramp-excitation data, starting from 220 mA with overlayed identification. Regression fit returning $\hat{d} = 298.5\,\text{mA}$. 42
4.2 Real-vehicle step response (full window). The NLS fit (3.16) yields $\hat{K} = 0.145$ and $\hat{\tau} = 0.288\,\text{s}$. 43
4.3 Closed-loop sweep on the real vehicle with the rule-based tuning ($\omega_{\text{CL}} = 2\,\text{rad/s}$, $k_{\text{ESO}} = 5$). The valve current saturates at 550 mA. . . 43

4.4 Zoom of the real-vehicle step response around the step instant. No angle motion is observed for roughly 0.17 s after the input changes. The red marked area is therefore the deadtime and the blue marked area the actual lag. 44
4.5 Synthetic plant with the identified gain $\hat{K}$, the decomposed lag $\tau_{\mathrm{lag}} = 0.118\,\mathrm{s}$, and an added 0.17 s transport delay, at the deployed tuning $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$, $k_{\mathrm{ESO}} = 5$. This is a simulation, not a vehicle test. . . . 45
4.6 The same synthetic plant as in Figure 4.5, at the detuned $\omega_{\mathrm{CL}} = 0.3\,\mathrm{rad/s}$, $k_{\mathrm{ESO}} = 3$. Lowering the bandwidth removes the stutter and the saturation. 46
4.7 Closed-loop sweep on the real vehicle after manual detuning to $\omega_{\mathrm{CL}} = 0.3\,\mathrm{rad/s}$ and $k_{\mathrm{ESO}} = 3$. 46
4.8 Targeted deadband test on the real vehicle, read after accounting for the transport delay (red). On the way up the angle starts to move at a higher current than the identified deadband (new deadband). On the way down it keeps moving until a much lower current, a directional hysteresis. A small angle bump appears when the vehicle stops. . . . 47

A.1 The OpenModelica simulation model, organised into its mechanical, hydraulic, control, and friction subsystems. I
A.2 Normalized geometric leverage for one piston of the steering linkage as a function of articulation angle γ. The leverage is maximal at $\gamma = 0$ and decreases toward the mechanical limits. VI

C.1 Noise-free tracking of the 2nd-order ADRC at $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$ and $k_{\mathrm{ESO}} = 3$ on the fast plants, showing the near-flat response that is the signature of an oversized b_0. X
C.2 Closed-loop response of the 2nd-order ADRC at $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$ and $k_{\mathrm{ESO}} = 3$ with measurement noise and half-gain tuning, for $\tau \in [0.2, 2.0]\,\mathrm{s}$. The chatter on the control signal grows with τ. XI
C.3 Zoomed linear-scale version of Figure 3.8 around the region where the NRMSE crosses and the HFR circles sit, with k_{ESO} fixed at 3. XII
C.4 Closed-loop response of the 1st-order ADRC at $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$ and $k_{\mathrm{ESO}} = 5$ with measurement noise and half-gain tuning, on the fast plants. XIII

List of Tables

3.1 Benchmark of the three candidate deadband-identification algorithms, used to select the one carried into the auto-tuning pipeline (Section 3.6). 25

3.2 Tuning rule produced by the parameter sweeps, with half-gain tuning applied in both regimes. 33

3.3 Auto-tuning pipeline output on the nine simulation vehicles of Table A.3. The deadband half-width $\hat{d}$ is identified by A-pos (Section 3.4.2.2). The plant parameters $\hat{K}$ and $\hat{\tau}$ are produced by the nonlinear least-squares estimator of Section 3.6.1. The ADRC order N, critical gain b_0, and closed-loop bandwidth ω_{CL} follow from the tuning rule of Section 3.5.2. Half-gain tuning and $k_{\mathrm{ESO}} = 5$ are applied throughout, as prescribed by the rule for $\hat{\tau} \leq 0.5\,\mathrm{s}$. 36

3.4 Closed-loop tracking metrics for the auto-tuned controllers on the nine simulation vehicles. 36

3.5 Closed-loop tracking metrics on the nine mismatched simulation vehicles. The mass/friction pattern $\pm M \pm F$ indicates whether mass and friction are scaled above (+) or below (−) the nominal values of Table A.1. 38

A.1 Baseline simulation model parameters. Per-vehicle scalings (Table A.3) are applied multiplicatively to these values. III

A.2 Geometry settings used in the validation set. The front and rear body lengths are measured along the longitudinal axis of each body. The rear articulation gap is the distance between the rear body and the articulation joint. III

A.3 The nine simulated vehicles used in this thesis. V0 is the nominal configuration; V1–V5 each shift one parameter axis; V6–V8 shift several axes at once. These are sampled probes into the simulation parameter space, not bounds on the vehicle set $\mathcal{V}$. Numeric entries are multiplicative scalings of the baseline values in Table A.1; geometry is one of three named settings. III

1

Introduction

1.1 Background

Autonomous driving requires a vehicle's motion to follow a requested path. If autonomous driving is applied to vehicles that are not always the same, but differ in configuration or even in model, the motion controller has to be adapted for each new vehicle. This also holds for articulated vehicles, which steer not by turning wheels but by rotating a front and a rear body segment relative to each other about a central joint. The direction of travel is set by the resulting articulation angle γ between the front and rear bodies, illustrated in Figure 1.1.

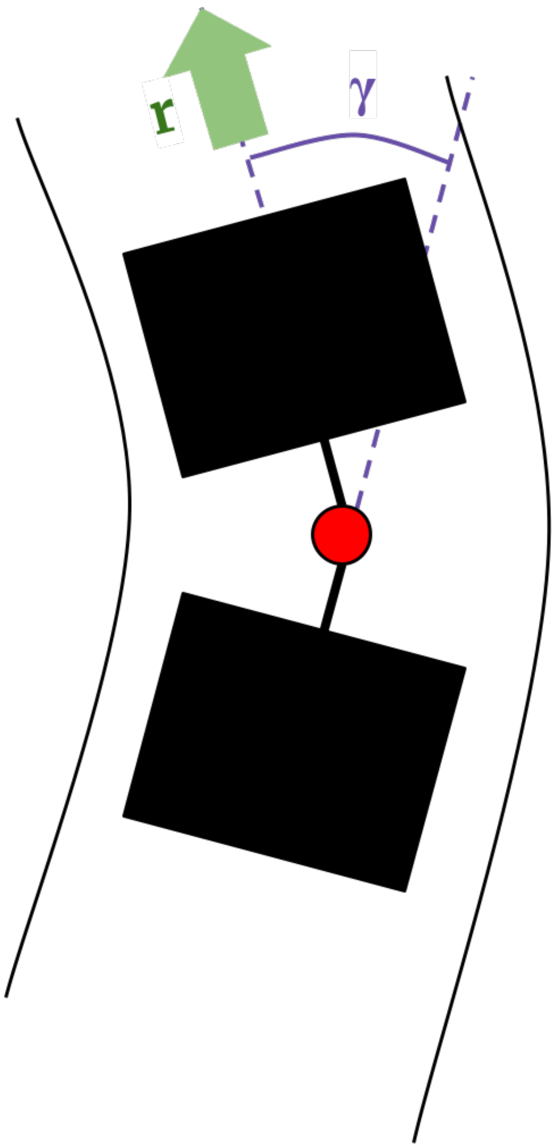

Figure 1.1: Schematic of an articulated vehicle. The articulation angle γ between the front and rear bodies sets the direction of travel and is the controlled output throughout this thesis, which should track the commanded reference r. The angle is actuated by a pair of hydraulic cylinders driven through an open-center tandem proportional directional valve.

Among these vehicles, a subset has electro-hydraulic actuation with an open-center tandem proportional directional valve (from now on the OCT valve), and it is this subset that this thesis targets. A valve solenoid current commands a hydraulic flow into a pair of cylinders, and the flow integrates into a change in the articulation angle γ. A motion controller sets the valve current based on the tracking error between the reference r and the articulation angle γ in a closed-loop, so that γ follows r. Its behavior is set by a small number of gains, which fix how strongly and quickly it reacts to a tracking error. Finding gains that give fast, stable tracking is called tuning. Vehicles within this subset are not identical, and their differences change the steering plant, so a controller that performs well on one vehicle might introduce oscillations or even instabilities, when trying to follow the reference, on another. Deploying a well-performing controller has typically required manual tuning and test drives by an experienced engineer per physical vehicle. Even for a single vehicle, this plant is not entirely fixed either. Ground friction and the load acting on the steering linkage differ between standing still and driving, and this can shift the plant's effective gain and time constant.

Electro-hydraulic actuation of steering platforms has been studied for tractors and similar machines, and the resulting steering plants are commonly described by a low-order linear model from input current to output angle [20, 14]. A known physical feature of the proportional directional valves used in such systems is a deadband around the zero point of the valve solenoid current command. This deadband occurs at the input stage, before the linear model, meaning that valve commands within this zone around zero produce no meaningful hydraulic flow to the cylinders and therefore no change in articulation angle, until the command exceeds the deadband threshold and the cylinder pair begins to move [1]. Based on that literature this thesis describes the steering plant by three blocks in series, an input deadband, a second-order linear time-invariant core, and a mechanical output saturation. This three-block view is the modeling premise the rest of the thesis builds on, and it is developed in detail in Section 3.2.

1.2 Contributions

The thesis investigated whether the manual calibration step of the motion controller could be replaced by an automated pipeline that identifies a small set of plant parameters from short standstill experiments to use them to configure an error-based Active Disturbance Rejection Control (ADRC) [7, 6, 11] law, for a defined family of vehicles that share the overall architecture of one vehicle provided by the company where this thesis was conducted.

The thesis delivers four contributions.

- An automated three-step tuning pipeline that runs at vehicle standstill. The pipeline first identifies and compensates the input deadband offline, then applies a standstill step and fits the gain K and time constant τ of a second-order LTI plant to the recorded response, and finally uses an empirical tuning rule built on top of bandwidth parameterisation and half-gain tuning [11, 10] to turn K and τ into the ADRC controller and ESO gains. The pipeline is described in Chapter 3.

- A set of three test environments of increasing complexity, a set of synthetic second-order plants with known parameters, an OpenModelica multi-body simulation model of the vehicle, and the real articulated vehicle as the final deployment (Section 3.3).

- Quantitative robustness evidence in the simulation environment across nine vehicle configurations that span chassis mass, cylinder bore, ground friction, valve flow capacity, and linkage geometry, with payload and ground-friction perturbations and measurement noise applied around the nominal operating point at which the parameters were identified. The robustness study and its metrics are reported in Section 3.7.

- An end-to-end deployment on the real vehicle and a diagnosis of why its closed-loop tracking falls short of the simulation results, identifying the unmodelled effects responsible as the open problem for future work (Chapter 4).

1.3 Thesis Outline

The remainder of the thesis is organised as follows. Chapter 2 introduces error-based ADRC, the Extended State Observer, and the bandwidth-parameterisation framework on which the empirical tuning rule is built. The methodology and verification flow of the resulting auto-tuning pipeline is shown in Figure 1.2. Chapters 3 to 4 follow this flow, developing each stage and verifying it across the test environments up to the real articulated vehicle. The thesis then closes with the discussion and conclusions in Chapter 5.

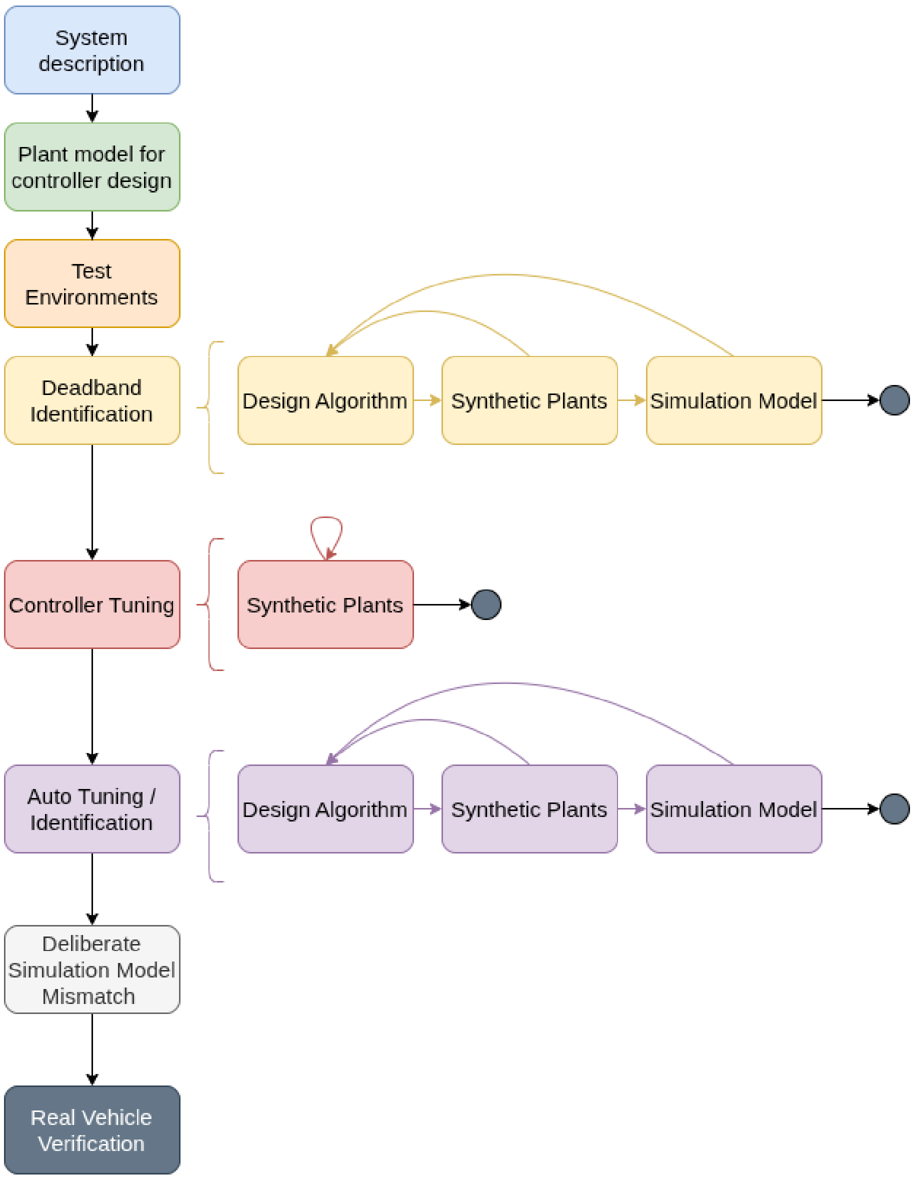

Figure 1.2: Methodology and verification flow of the auto-tuning pipeline, read top to bottom along the left column, where each box is a stage developed in Chapter 3. Each coloured branch to the right is the develop-and-verify loop for that stage, run across the test environments. Straight arrows give the flow direction, curved arrows are iteration loops back to an earlier step, and a filled black circle marks the end of a stage's development loop.

2

Active Disturbance Rejection Control

This chapter develops the Active Disturbance Rejection Control (ADRC) formulation. It is then used in the auto-tuning pipeline in Chapter 3 to handle the gap between the standstill operating point used for identification and the driving operating point where the vehicle actually has to perform.

Section 2.1 introduces the plant model and the total disturbance, Section 2.2 derives the error-based variant used in this thesis, and Section 2.3 reduces the resulting five gains to two design bandwidths.

2.1 Plant Model and Total Disturbance

ADRC was first proposed by Han [7] and later put into a practical linear form by Gao [6]. The key idea is to treat all uncertainties, that is unmodelled dynamics, parameter variations, nonlinearities, and external disturbances, as a single quantity called the *total disturbance*, estimate it in real time using an extended state observer (ESO), and cancel it in the control law.

Consider a second-order, single-input single-output (SISO) plant described by the differential equation [11, 6]

$$\ddot{y}(t) + a_1\,\dot{y}(t) + a_0\,y(t) = b\,u(t) + b_{\mathrm{d}}\,d(t), \tag{2.1}$$

where y is the output, u is the control input, d is an external disturbance, and $a_0, a_1, b, b_{\mathrm{d}}$ are plant and disturbance model parameters.

The key modelling step of ADRC is to rearrange (2.1) so that all unknown terms are collected into a single time-varying quantity $f(t)$, called the *total disturbance* [6]. Introducing the nominal gain b_0, we write

$$\ddot{y}(t) = \underbrace{-a_1\dot{y}(t) - a_0 y(t) + \Delta b \cdot u(t) + b_{\mathrm{d}}\,d(t)}_{f(t)} + b_0\,u(t), \tag{2.2}$$

where $\Delta b = b - b_0$ is the gain mismatch. Equation (2.2) reduces to

$$\ddot{y}(t) = f(t) + b_0\,u(t). \tag{2.3}$$

This is the plant model used by a 2nd-order ADRC: a double integrator with input gain b_0 and an additive disturbance $f(t)$. The parameter b_0 is the only plant parameter ADRC requires and is referred to as the *critical gain parameter* [11]. While this example used a second-order plant ($N = 2$), the same framework extends to any plant order N by augmenting the state vector accordingly [11]. Because b_0 and the plant order N have to approximately match the real plant that the ADRC controls, both have to be extracted from it.

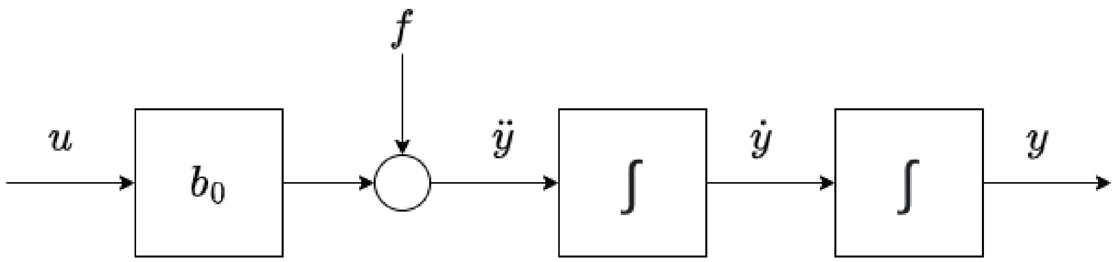

Figure 2.1: Plant model assumed by a second-order ADRC, equation (2.3): a double integrator driven through the input gain b_0, with all unmodelled dynamics and disturbances lumped into the single total disturbance f. Adapted from [11].

The compact plant model (2.3), shown in Figure 2.1, can be used as the basis for two ADRC variants. In the output-based variant [11] the observer and controller use the plant output $y(t)$ and its estimated derivatives, and the control law also needs the reference derivatives $r^{(N)}(t)$ to track non-constant references. If those derivatives are not available, they have to be dropped from the control law, which degrades tracking performance for continuously changing references, a common situation in motion control applications. This thesis uses the *error-based* variant, which formulates the observer and controller in terms of the tracking error $e(t)$ and therefore does not need those derivatives. The compact form (2.3) is used by the ESO in Section 2.2.2, the error-based reformulation provides the control law in Section 2.2.3, and b_0 is the single plant parameter that the auto-tuning pipeline in Chapter 3 has to supply from standstill identification.

2.2 Error-Based ADRC

This section derives the error-based variant of ADRC that the auto-tuning pipeline of Chapter 3 tunes. It first writes the dynamics in terms of the tracking error, then adds the Extended State Observer and the control law.

2.2.1 Error Dynamics

Let $r(t)$ be the reference to be tracked. The tracking error is then

$$e(t) = r(t) - y(t). \tag{2.4}$$

Differentiating twice and substituting the plant model (2.3) gives:

$$\ddot{e}(t) = \ddot{r}(t) - \ddot{y}(t) = \underbrace{\ddot{r}(t) - f(t)}_{f^*(t)} - b_0\, u(t), \tag{2.5}$$

where $f^*(t)$ is the *error-domain total disturbance* [11]. It contains all the same unknown terms as $f(t)$ plus the unknown second derivative of the reference $\ddot{r}(t)$. Two differences from the output form (2.3) are important: the error dynamics describe the *error* rather than the output, and the sign attached to b_0 is negative [11].

2.2.2 Extended State Observer

The ESO estimates the tracking error, its derivative, and the error-domain total disturbance from the two signals that are measured online: the tracking error $e(t)$ from (2.4) and the applied control input $u(t)$. To make $f^*(t)$ available to the estimator, it is added to the state vector as an extra state. The augmented state vector becomes $\boldsymbol{x}(t) = (e,\, \dot{e},\, f^*)^\mathsf{T}$, giving the state-space model [11]

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\,\boldsymbol{x}(t) - \boldsymbol{b}\,u(t) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \dot{f}^*(t), \qquad e(t) = \boldsymbol{c}^\mathsf{T}\boldsymbol{x}(t), \tag{2.6}$$

with

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0 \\ b_0 \\ 0 \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{2.7}$$

The last row of (2.6) states $\dot{x}_3 = \dot{f}^*(t)$, confirming that $x_3 = f^*(t)$. The $(0, 0, 1)^\mathsf{T}\dot{f}^*(t)$ term is required for an exact model, but $\dot{f}^*$ is not measurable. It is therefore omitted from the observer design, which Herbst and Madonski in [11] refer to as the *virtual input*. Omitting it treats $\hat{x}_3$ as a constant total disturbance estimate.

A so called Luenberger observer for (2.6), with the virtual input omitted, gives the ESO [11]:

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{A}\,\hat{\boldsymbol{x}}(t) - \boldsymbol{b}\,u(t) + \boldsymbol{l}\left(e(t) - \boldsymbol{c}^\mathsf{T}\hat{\boldsymbol{x}}(t)\right), \tag{2.8}$$

where $\boldsymbol{l} = (l_1,\, l_2,\, l_3)^\mathsf{T}$ is the observer gain vector. In component form:

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 + l_1\,(e - \hat{x}_1), \\ \dot{\hat{x}}_2 &= \hat{x}_3 - b_0\,u + l_2\,(e - \hat{x}_1), \\ \dot{\hat{x}}_3 &= l_3\,(e - \hat{x}_1). \end{aligned} \tag{2.9}$$

With the virtual input dropped, the third row of (2.9) reduces to $\dot{\hat{x}}_3 = l_3(e - \hat{x}_1)$, so the correction term $l_3(e - \hat{x}_1)$ acts as an integrator that drives $\hat{x}_3$ toward the true f^*. This gives ADRC zero steady-state error for constant disturbances and references [11].

2.2.3 Control Law

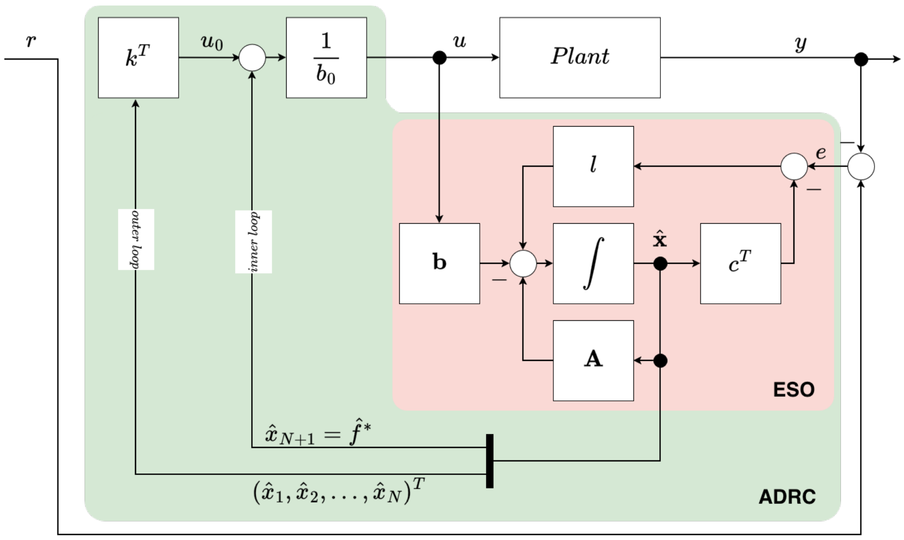

Figure 2.2: Complete error-based ADRC, equations (2.10) and (2.12). The red region is the Extended State Observer, which estimates the error states and the total disturbance, and the surrounding green region is the ADRC that builds the control signal from those estimates and rejects the disturbance. Adapted from [11].

The ESO (2.9) provides the estimated error states $\hat{x}_1$, $\hat{x}_2$ and the total disturbance estimate $\hat{x}_3$. From these, the control signal is built in two nested steps, shown in Figure 2.2 [11].

The *inner loop* in Figure 2.2 does the disturbance rejection and the plant gain inversion. Using the total disturbance estimate $\hat{f}^*(t) = \hat{x}_3(t)$ from the ESO (2.9), the control signal

$$u(t) = \frac{u_0(t) + \hat{f}^*(t)}{b_0} \tag{2.10}$$

cancels the effect of $f^*(t)$ and compensates the plant gain. Under perfect estimation ($\hat{f}^* = f^*$), substituting into (2.5) yields a unity-gain double integrator in the error:

$$\ddot{e}(t) \approx -u_0(t). \tag{2.11}$$

The *outer loop* controls the resulting double integrator using linear state feedback on the estimated error states:

$$u_0(t) = k_1\,\hat{x}_1(t) + k_2\,\hat{x}_2(t). \tag{2.12}$$

Combining the two loops, the overall control law is

$$u(t) = \frac{1}{b_0}\Big(k_1\,\hat{x}_1 + k_2\,\hat{x}_2 + \hat{x}_3\Big) = \frac{1}{b_0}\,(\boldsymbol{k}^\mathsf{T}\ \ 1)\,\hat{\boldsymbol{x}}(t), \tag{2.13}$$

where $\boldsymbol{k}^\mathsf{T} = (k_1\ \ k_2)$ and $\hat{\boldsymbol{x}} = (\hat{x}_1\ \ \hat{x}_2\ \ \hat{x}_3)^\mathsf{T}$.

2.3 Bandwidth Parameterization

Section 2.1 set the plant model and Section 2.2 set the controller and observer structure, but left the five gains, two for the controller and three for the observer, open. This section shows how bandwidth parameterization [5] reduces those five gains to two design parameters, one for the controller and one for the observer, which the empirical tuning rule of Section 3.5.2 later sets. The two subsections below cover the controller and the observer in turn.

2.3.1 Controller Tuning

The state-feedback controller acts on the normalised double integrator (2.11). Inserting the state-feedback law (2.12) into (2.11) gives the closed-loop characteristic polynomial

$$(\lambda + \omega_{\mathrm{CL}})^2 = \lambda^2 + k_2\lambda + k_1, \tag{2.14}$$

where both poles are placed at $-\omega_{\mathrm{CL}}$. Expanding the left-hand side gives

$$k_1 = \omega_{\mathrm{CL}}^2, \qquad k_2 = 2\,\omega_{\mathrm{CL}}. \tag{2.15}$$

2.3.2 Observer Tuning

The observer error dynamics are described by the characteristic polynomial of $(\boldsymbol{A} - \boldsymbol{l}\,\boldsymbol{c}^{\mathsf{T}})$. Placing all three observer poles at $-\omega_{\mathrm{O}}$ yields [5, 11]

$$(\lambda + \omega_{\mathrm{O}})^3 = \lambda^3 + l_1\lambda^2 + l_2\lambda + l_3, \tag{2.16}$$

giving

$$l_1 = 3\,\omega_{\mathrm{O}}, \qquad l_2 = 3\,\omega_{\mathrm{O}}^2, \qquad l_3 = \omega_{\mathrm{O}}^3. \tag{2.17}$$

The observer bandwidth is expressed relative to the closed-loop bandwidth using a factor k_{ESO} [9, 11]:

$$\omega_{\mathrm{O}} = k_{\mathrm{ESO}} \cdot \omega_{\mathrm{CL}}, \qquad k_{\mathrm{ESO}} \in [3, 10]. \tag{2.18}$$

Substituting into (2.17):

$$l_1 = 3\,k_{\mathrm{ESO}}\,\omega_{\mathrm{CL}}, \qquad l_2 = 3\,k_{\mathrm{ESO}}^2\,\omega_{\mathrm{CL}}^2, \qquad l_3 = k_{\mathrm{ESO}}^3\,\omega_{\mathrm{CL}}^3. \tag{2.19}$$

A larger k_{ESO} improves disturbance estimation speed but amplifies measurement noise, making the choice a trade-off between rejection bandwidth and noise sensitivity.

2.3.3 Half-Gain Tuning

Bandwidth parameterization produces feedback gains that grow with the desired bandwidth. In noisy applications this can make the control signal sensitive to measurement noise. Half-gain tuning addresses this by halving all feedback gains while keeping roughly the same closed-loop speed [10].

The method comes from the α-controller design, which finds gains by solving an algebraic Riccati equation such that the closed-loop system decays at rate α. When this approach is applied to ADRC with $\alpha = \omega_{\mathrm{CL}}$ for the controller and $\alpha = k_{\mathrm{ESO}} \cdot \omega_{\mathrm{CL}}$ for the observer, the resulting gains are exactly half the bandwidth-parameterization gains [10]:

$$k_i^{\mathrm{hg}} = \frac{1}{2}\, k_i, \qquad l_i^{\mathrm{hg}} = \frac{1}{2}\, l_i, \tag{2.20}$$

where k_i and l_i are from (2.15) and (2.19). There is therefore no need to solve the Riccati equation in practice, the half-gain values are simply the bandwidth-parameterization gains divided by two.

The trade-off is that the closed-loop poles become complex rather than purely real. For second-order ADRC the controller poles shift from $\lambda = -\omega_{\mathrm{CL}}$ to

$$\lambda_{1/2} = \left(-\frac{1}{2} \pm \frac{1}{2}\,\mathrm{j}\right) \omega_{\mathrm{CL}}, \tag{2.21}$$

giving a mildly underdamped step response instead of a critically damped one.

Half-gain tuning can be applied to the controller, the observer, or both independently. A practical starting point is to halve the observer gains only: this significantly reduces how much high-frequency noise reaches the control signal with almost no change to the tracking or disturbance rejection dynamics [10]. Although the derivation above is in continuous time, the same halving can be applied to the discrete-time gains from Section 2.4 after the z-domain pole placement step, and has been shown to produce the same noise reduction in practice [10].

2.4 Practical Discrete-Time Error-Based ADRC

This section presents the discrete-time implementation of error-based ADRC, the form run in every closed-loop test from the synthetic sweeps to the real vehicle (Chapter 4), along with the practical considerations needed for deployment on a real system.

2.4.1 Controller

Since the control law (2.13) has no dynamics, it transfers directly to discrete time by replacing t with the sampling instant k [11]:

$$u(k) = \frac{1}{b_0} \cdot \begin{pmatrix} \boldsymbol{k}_{\mathrm{d}}^{\mathsf{T}} & 1 \end{pmatrix} \cdot \hat{\boldsymbol{x}}(k), \qquad \boldsymbol{k}_{\mathrm{d}}^{\mathsf{T}} = \begin{pmatrix} k_{\mathrm{d},1} & k_{\mathrm{d},2} \end{pmatrix}. \tag{2.22}$$

The discrete gains $k_{\mathrm{d},1}, k_{\mathrm{d},2}$ play the same role as the continuous gains (2.15) but take different values. The continuous design places both poles at $-\omega_{\mathrm{CL}}$, so in discrete time the poles go to the matching z-domain location

$$z_{\mathrm{CL}} = e^{-\omega_{\mathrm{CL}} T_{\mathrm{s}}}. \tag{2.23}$$

Setting the closed-loop characteristic polynomial of the discretized double integrator (2.11) equal to $(z - z_{\mathrm{CL}})^2$,

$$z^2 - \left(2 - \tfrac{1}{2} k_{\mathrm{d},1} T_{\mathrm{s}}^2 - k_{\mathrm{d},2} T_{\mathrm{s}}\right) z + \left(1 - k_{\mathrm{d},2} T_{\mathrm{s}} + \tfrac{1}{2} k_{\mathrm{d},1} T_{\mathrm{s}}^2\right) = (z - z_{\mathrm{CL}})^2, \tag{2.24}$$

and solving for the gains gives [11]

$$k_{\mathrm{d},1} = \frac{(1-z_{\mathrm{CL}})^2}{T_{\mathrm{s}}^2}, \qquad k_{\mathrm{d},2} = \frac{4-(1+z_{\mathrm{CL}})^2}{2\,T_{\mathrm{s}}}. \tag{2.25}$$

2.4.2 Observer

Unlike the control law, the ESO is a dynamic system, so digital implementation needs a discrete-time version of it. Applying zero-order hold (ZOH) discretization to the augmented system first gives the discrete-time Luenberger observer in predictive form,

$$\hat{\boldsymbol{x}}(k+1|k) = \boldsymbol{A}_{\mathrm{d}} \cdot \hat{\boldsymbol{x}}(k|k-1) - \boldsymbol{b}_{\mathrm{d}} \cdot u(k) + \boldsymbol{l}_{\mathrm{p}} \cdot \Big(e(k) - \boldsymbol{c}^{\mathsf{T}} \cdot \hat{\boldsymbol{x}}(k|k-1)\Big) \tag{2.26}$$

with

$$\boldsymbol{A}_{\mathrm{d}} = e^{\boldsymbol{A}T_{\mathrm{s}}}, \qquad \boldsymbol{b}_{\mathrm{d}} = \int_0^{T_{\mathrm{s}}} e^{\boldsymbol{A}\tau} d\tau \boldsymbol{b}. \tag{2.27}$$

It is a *predictive observer* because the estimate $\hat{\boldsymbol{x}}(k|k-1)$ uses only measurements up to $k-1$, which adds one sample of delay. To use the current measurement $e(k)$ as well, the observer is recast into *current-observer* form, which splits each step into a prediction from past data and a correction with the present measurement [11],

$$\hat{\boldsymbol{x}}(k|k-1) = \boldsymbol{A}_{\mathrm{d}} \cdot \hat{\boldsymbol{x}}(k-1|k-1) - \boldsymbol{b}_{\mathrm{d}} \cdot u(k-1) \qquad \text{(prediction)} \tag{2.28}$$

$$\hat{\boldsymbol{x}}(k|k) = \hat{\boldsymbol{x}}(k|k-1) + \boldsymbol{l}_{\mathrm{d}} \cdot \Big(e(k) - \boldsymbol{c}_{\mathrm{d}}^{\mathsf{T}} \cdot \hat{\boldsymbol{x}}(k|k-1)\Big) \qquad \text{(correction)} \tag{2.29}$$

Combining the two steps gives

$$\hat{\boldsymbol{x}}(k|k) = \boldsymbol{A}_{\mathrm{ESO}}\, \hat{\boldsymbol{x}}(k-1|k-1) - \boldsymbol{b}_{\mathrm{ESO}}\, u(k-1) + \boldsymbol{l}_{\mathrm{d}}\, e(k), \tag{2.30}$$

where $\boldsymbol{l}_{\mathrm{d}} = (l_{\mathrm{d},1}, l_{\mathrm{d},2}, l_{\mathrm{d},3})^{\mathsf{T}}$ are the discrete observer gains and

$$\boldsymbol{A}_{\mathrm{ESO}} = \boldsymbol{A}_{\mathrm{d}} - \boldsymbol{l}_{\mathrm{d}} \cdot \boldsymbol{c}_{\mathrm{d}}^{\mathsf{T}} \cdot \boldsymbol{A}_{\mathrm{d}}, \qquad \boldsymbol{b}_{\mathrm{ESO}} = \boldsymbol{b}_{\mathrm{d}} - \boldsymbol{l}_{\mathrm{d}} \cdot \boldsymbol{c}_{\mathrm{d}}^{\mathsf{T}} \cdot \boldsymbol{b}_{\mathrm{d}}, \qquad \boldsymbol{c}_{\mathrm{d}}^{\mathsf{T}} = \boldsymbol{c}^{\mathsf{T}}. \tag{2.31}$$

From here on the corrected estimate $\hat{\boldsymbol{x}}(k|k)$ is written $\hat{\boldsymbol{x}}(k)$ for short, as in the control law (2.22).

The discrete gains $l_{\mathrm{d},1}, l_{\mathrm{d},2}, l_{\mathrm{d},3}$ play the same role as the continuous gains (2.19) but take different values. The continuous design places all three observer poles at $-\omega_{\mathrm{O}} = -k_{\mathrm{ESO}}\,\omega_{\mathrm{CL}}$, so in discrete time the poles go to the matching z-domain location

$$z_{\mathrm{ESO}} = e^{-k_{\mathrm{ESO}}\,\omega_{\mathrm{CL}}\,T_{\mathrm{s}}}. \tag{2.32}$$

Placing all three observer poles at z_{ESO} gives [11]

$$l_{\mathrm{d},1} = 1 - z_{\mathrm{ESO}}^3, \qquad l_{\mathrm{d},2} = \frac{3}{2\,T_{\mathrm{s}}}(1-z_{\mathrm{ESO}})^2(1+z_{\mathrm{ESO}}), \qquad l_{\mathrm{d},3} = \frac{(1-z_{\mathrm{ESO}})^3}{T_{\mathrm{s}}^2}. \tag{2.33}$$

Substituting into (2.31) gives the explicit matrices for second-order ADRC [11]:

$$\boldsymbol{A}_{\mathrm{ESO}} = \begin{pmatrix} 1-l_{\mathrm{d},1} & T_{\mathrm{s}}(1-l_{\mathrm{d},1}) & \frac{1}{2}T_{\mathrm{s}}^2(1-l_{\mathrm{d},1}) \\ -l_{\mathrm{d},2} & 1-l_{\mathrm{d},2}T_{\mathrm{s}} & T_{\mathrm{s}} - \frac{1}{2}l_{\mathrm{d},2}T_{\mathrm{s}}^2 \\ -l_{\mathrm{d},3} & -l_{\mathrm{d},3}T_{\mathrm{s}} & 1-\frac{1}{2}l_{\mathrm{d},3}T_{\mathrm{s}}^2 \end{pmatrix}, \tag{2.34}$$

$$\boldsymbol{b}_{\mathrm{ESO}} = \begin{pmatrix} \frac{1}{2} b_0 T_{\mathrm{s}}^2 (1 - l_{\mathrm{d},1}) \\ b_0 T_{\mathrm{s}} - \frac{1}{2} l_{\mathrm{d},2} b_0 T_{\mathrm{s}}^2 \\ -\frac{1}{2} l_{\mathrm{d},3} b_0 T_{\mathrm{s}}^2 \end{pmatrix}. \tag{2.35}$$

Equations (2.30) to (2.35) complete the discrete observer. Together with the discrete control law (2.22), they form the digital ADRC that runs in every closed-loop test of this thesis, with all five gains set by the three parameters ω_{CL}, k_{ESO} and b_0 that the tuning pipeline of Chapter 3 supplies.

2.4.3 Controller Output Limitation and Anti-Windup

The control law (2.22) computes the control signal $u(k)$ without accounting for the actuator's finite output limits. When the requested signal exceeds those limits, the actuator output stays at its maximum or minimum value, so further changes in the computed signal $u(k)$ no longer reach the plant and feedback action is lost for as long as the limit is hit. Because ADRC contains integral action through its total disturbance estimation, the integrator state continues to accumulate error during saturation, leading to *integrator windup* [11].

ADRC provides a simple and effective anti-windup mechanism: the observer is fed the *actual* (saturated) control signal u_{lim} instead of the computed u. The magnitude-limited control signal is

$$u_{\mathrm{lim}}(k) = \mathrm{sat}_{u_{\mathrm{min}}}^{u_{\mathrm{max}}}\Big(u(k)\Big) = \begin{cases} u_{\mathrm{min}}, & u(k) < u_{\mathrm{min}}, \\ u(k), & u_{\mathrm{min}} \le u(k) \le u_{\mathrm{max}}, \\ u_{\mathrm{max}}, & u(k) > u_{\mathrm{max}}. \end{cases} \tag{2.36}$$

By feeding u_{lim} back to the observer, the observer's internal model remains consistent with the actual signal applied to the plant. This is the only measure required to prevent windup in ADRC [11].

3

Methodology for Auto-Tuning Pipeline

This chapter develops the auto-tuning pipeline named as the central contribution in Section 1.2. It defines the vehicle set the method applies to, the plant model the controller is designed against, the test environments used for verification, and the pipeline steps that turn standstill measurements into a working ADRC controller. The sections below follow the order of the pipeline overview in Figure 1.2. The deployment of the integrated pipeline on the real vehicle is reported separately in Chapter 4.

3.1 System Description and Vehicle Set

This section is the *System description* stage, marked in blue in Figure 1.2. It sets out the architectural features that define the vehicle set $\mathcal{V}$ and the modelling assumptions the controller design relies on, both feeding the plant model of Section 3.2 and the pipeline built on it.

3.1.1 Architectural requirements

A vehicle v belongs to the vehicle set $\mathcal{V}$ when it satisfies all of the following, each of which can be checked by inspecting the machine:

1. **Dual-body articulated chassis.** A front and a rear unit coupled by a single central revolute joint with a finite mechanical range $\gamma \in [\gamma_{\min}, \gamma_{\max}]$, where γ is the articulation angle shown in Figure 1.1.
2. **Tracked locomotion.** The methodology targets tracked machines. Wheeled articulated vehicles are not considered.
3. **Electro-hydraulic articulation actuation.** Hydraulic cylinders driven by an open-center tandem proportional directional valve fed by a constant-flow supply.
4. **Single input, single output.** The control input is the valve solenoid current. The measured output is the articulation angle γ.

3.1.2 Modelling assumptions

For a vehicle in $\mathcal{V}$, the controller is designed against the second-order plant model of Section 3.2. That model is an accurate description of the vehicle under the following assumptions:

5. **Type-1 plant.** The open-loop response from valve current to articulation angle has exactly one integrator and is otherwise stable.

6. **Second-order dominant dynamics.** The plant is well approximated by the second-order model developed in Section 3.2, which holds when the control bandwidth stays below the valve resonance.

7. **Negligible transport delay.** Any transport delay θ in the actuator chain is assumed negligibly small, or at least small enough for the control speeds needed for the slow steering of these vehicles, and is not estimated by the pipeline. Characterising the largest θ for which this holds is left to future work.

8. **Available sample time.** A controller sample time of $T_\mathrm{s} \leq 10\,\mathrm{ms}$ is available. This thesis validates the solution only at a sample time of $10\,\mathrm{ms}$.

3.2 Plant Model for Controller Design

With the vehicle set established, this section defines the plant model assumption the controller is designed against. It is the *Plant model for controller design* stage marked in green in Figure 1.2 that the deadband identification, plant identification, and tuning stages (Sections 3.4 to 3.6) build on. It first sets up the closed loop and the three-block view of the steering plant, with the input deadband (Subsection 3.2.2), the second-order LTI core (Subsection 3.2.3), and the output saturation (Subsection 3.2.4), developed in turn.

3.2.1 Expected Closed-Loop Structure

The error-based ADRC controller of Chapter 2 closes a loop around the steering plant derived in this section, shown in Figure 3.1. In the generic notation of Chapter 2, $y(t)$ becomes the articulation angle $\gamma(t)$. The controller acts on the tracking error $e(t) = r(t) - \gamma(t)$ between the commanded reference $r(t)$ and the measured articulation angle $\gamma(t)$, both shown in Figure 1.1. The steering plant is split into three blocks in series, an input deadband, a second-order LTI core, and an output saturation.

Figure 3.1 assumes the plant behaves as the analytical model developed in this section. Some test environments (Section 3.3) deviate from this model, and the ESO absorbs any such mismatch as part of the total disturbance $f(t)$ (Chapter 2).

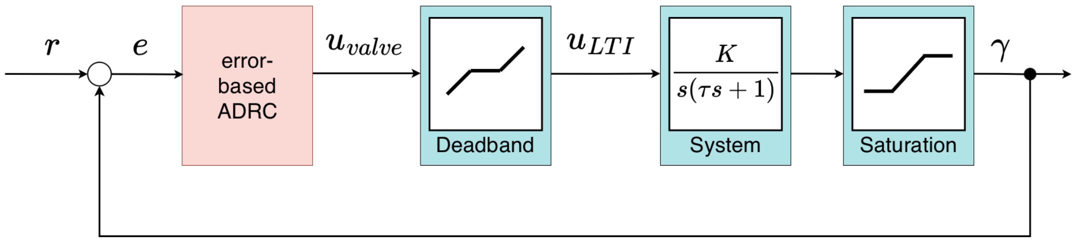

Figure 3.1: The simplified steering plant of this thesis, in closed loop with the error-based ADRC controller of Chapter 2. The control signal $u(t)$ is given by (2.13) and the output is the articulation angle $\gamma(t) = y(t)$. This is the model the rest of the methodology is designed against.

3.2.2 Constant Deadband Approximation

The input deadband, the first blue block in Figure 3.1, comes from the OCT valve introduced in Section 1.1. One reason for the deadband is the minimum current the solenoid needs before the spool begins to move. The dominant reason is internal recirculation. With the spool near neutral, the pump flow returns to the tank through an open internal path while the work ports A and B to the cylinder stay closed. As the spool moves off center the work ports begin to open, but the return path stays open too, so the flow keeps taking the easier route to the tank and only a negligible amount reaches the cylinder. The actuator stays stationary even though the spool has moved. This is the recirculation zone (blue in Figure 3.2), and it is the dominant cause of the valve's functional deadband [1, 17].

Only as the spool travels further does the return path close, the flow reach the cylinder, and the actuator begin to move (the transition and full-opening regions, red and green). The recirculation zone alone is the part this thesis treats as a deadband.

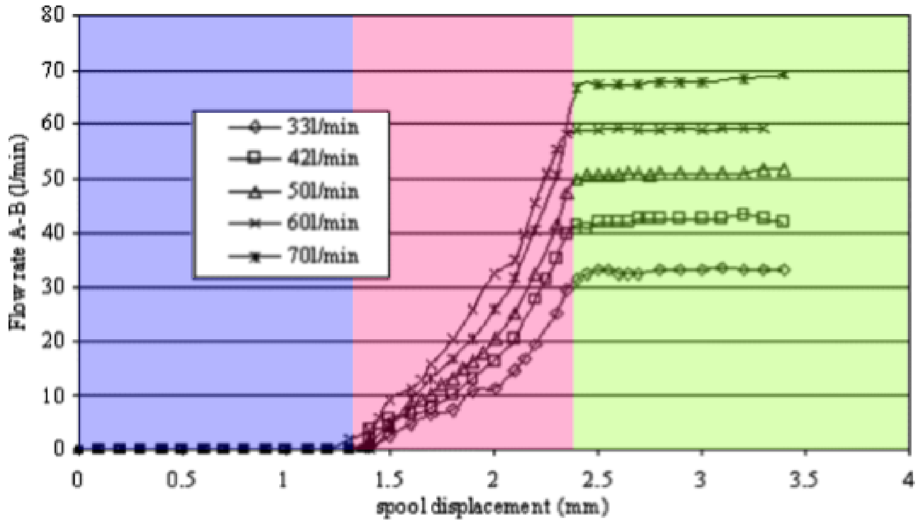

Figure 3.2: Flow to ports A and B as a function of spool stroke for several pump flow rates, for the test valve of [1]. The colored overlays added by the author indicate the three behavioral regions identified in the source paper: recirculation only (blue), transition (red), and full opening (green). The blue region is the one approximated as a constant deadband in this work. Adapted from [1].

The width of the recirculation zone depends on the load [17]. Under heavy inertia or high-pressure loads, the spool has to travel further before the supply pressure rises enough to overcome the load, which extends the effective deadband. This thesis nonetheless approximates the recirculation zone as a single constant deadband, identified at one nominal operating condition, and assumes the resulting error is absorbed into the total disturbance by the ESO (Chapter 2). Its half-width is identified and compensated in Section 3.4.

3.2.3 Second-Order LTI Core

The middle blue block of Figure 3.1 maps the effective valve opening $u_{\mathrm{LTI}}(t)$ past the deadband (Section 3.2.2) to the articulation angle $\gamma(t)$. This thesis models it as a second-order Type-1 plant

$$G(s) = \frac{K}{s\,(\tau s + 1)}, \tag{3.1}$$

with gain K and lag time constant τ. This is the plant dynamics the controller is designed and tuned against.

This low-order form is the standard model for electro-hydraulic steering actuators, with the control voltage as input and the steered angle as output [20, 14]. Yin et al. reduce a higher-order model to this form because the valve dynamics are much faster than the steering dynamics, the second-order-dominance assumption of the vehicle set (Section 3.1).

Both references also include a transport-delay term $e^{-\theta s}$, which Yin et al. compensate with a Smith predictor. This thesis instead omits it, as its own simplification rather than one from the literature, mentioned as the negligible-transport-delay assumption of Section 3.1. Chapter 4 returns to it as the main cause of the tracking shortfall on the real vehicle.

The parameters K and τ are estimated by the auto-tuning identification of Section 3.6.

3.2.4 Output Saturation

The third blue block in Figure 3.1 is the mechanical saturation at the output. The articulation angle cannot exceed its physical range $[\gamma_{\min}, \gamma_{\max}]$, defined by the vehicle geometry. The saturation limits are supplied by the user rather than identified, and the block is handled outside the controller by keeping the reference signal within them.

3.3 Test Environments

This section describes the three test environments of increasing complexity introduced in Chapter 1, marked in orange in Figure 1.2 and used to develop and verify every later pipeline stage. Section 3.3.1 covers the synthetic plants, Section 3.3.2 the simulation model, and Section 3.3.3 the real vehicle.

3.3.1 Synthetic system

The first test environment (Figure 1.2) is a set of synthetic plants, the simplest of the three. Their parameters are known exactly and they have no hidden dynamics, so a failure here points to the algorithm rather than to imperfect knowledge of the plant.

The plants are exactly the second-order Type-1 LTI core of the plant model (Section 3.2, (3.1)):

$$G(s) = \frac{K}{s\,(\tau s + 1)}. \tag{3.2}$$

The time constant τ is swept from 0.01 to 2.0 s, spanning fast actuators up to slow ones. The gain K is fixed at 1.0 for the closed-loop tracking sweeps but is itself swept for the plant identification tests. The lower bound of 0.01 s is set by the controller sample time $T_s = 10\,\text{ms}$, since any shorter time constant falls below one sample period and cannot be resolved.

During testing, white measurement noise with standard deviation $\sigma = 0.002\,\text{rad}$ is added at the output, and for the deadband identification routine (Section 3.4) a deadband is added in front of the plant input.

3.3.2 Simulation model

The second test environment (Figure 1.2) is a simulation model of the articulated vehicle, built in OpenModelica. This multi-body and hydraulic physics model adds the coupled dynamics the synthetic plants leave out. It verifies the auto-tuning pipeline derived on the synthetic plants and serves as the main development environment, since access to the real vehicle is very limited (Chapter 4). It is run on nine instances, V_0 to V_8. The instance V_0 mirrors the real vehicle, and V_1 to V_8 are samples of the vehicle set $\mathcal{V}$ chosen to stress-test the methodology. The design description, the model simplifications, and the exact parameters of the nine vehicles are given in Appendix A.

3.3.3 Real vehicle

The third test environment (Figure 1.2) is the real articulated vehicle, the unit the full pipeline is deployed on in Chapter 4. It is one specific member of $\mathcal{V}$, satisfying the architectural requirements of Section 3.1 by construction, and the unit that the simulation instance V_0 is parameterised to match. It differs from the other two environments in one decisive way. The modelling assumptions of Section 3.1 hold by construction in the synthetic and simulation environments, but on real hardware they may not. The real vehicle is therefore the only environment that can show whether those assumptions survive in practice. Here the deadband compensator, the controller, and the auto-tuning routine run together as one integrated pipeline for the final verification.

3.4 Deadband Identification

This section is the *Deadband Identification* stage, marked in yellow in Figure 1.2. It handles the input deadband block of the plant model, the recirculation deadband of the OCT valve introduced in Section 3.2.2 and drawn as the first block of Figure 3.1. Section 3.4.1 compensates it with a static inverse placed before the control loop, and Section 3.4.2 identifies the half-width $\hat{d}$ that the inverse needs.

3.4.1 Deadband Compensation

The controller (Chapter 2) is designed for the linear time-invariant (LTI) core of the three-block plant model (Section 3.2). Research into applying ADRC to systems with valve deadzones [19, 8] suggests that relying solely on the disturbance observer to suppress such nonlinearities can lead to decreased performance. While those authors propose complex online adaptive methods, this thesis takes a simpler, modular approach. The deadband is identified offline and compensated with a static inverse before the control loop. By separating the two, the ADRC can focus on the linear part of the plant. This keeps the control design simple and makes it easier to adapt in the future.

The controller output is normalized to $[-1, 1]$ so it stays independent of the valve input current, which differs between vehicles. The compensator sits between the controller and the physical valve, as shown in Figure 3.3. On the controller side the working signal is $u_{\text{comp}} \in [-1, 1]$, the desired *useful actuation* applied to the plant. On the valve side the working signal is the actual current $u_{\text{valve}} \in [-V_{\max}, V_{\max}]$ that drives the solenoid, where $V_{\max}$ is the absolute valve-current limit. Inside the deadband no useful flow reaches the cylinder (Section 3.2.2), so the current must jump over the deadband region before any actuation is produced. Outside the deadband the controller's range must be scaled into the linear current range so that $u_{\text{comp}} = \pm 1$ produces $u_{\text{valve}} = \pm V_{\text{max}}$. The result of these two operations is the useful actuation signal $u_{\text{LTI}} \in [-1, 1]$ entering the LTI plant. The goal of the compensator is $u_{\text{LTI}} = u_{\text{comp}}$, so that the ADRC can act as if the deadband were not there.

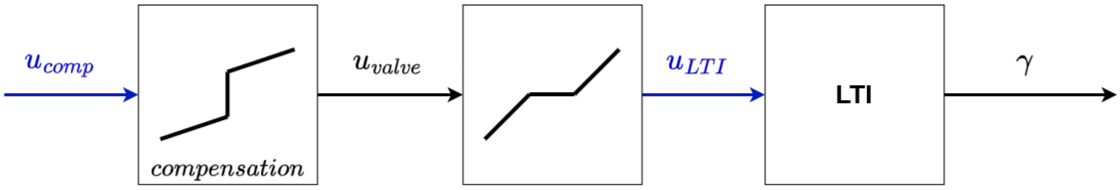

Figure 3.3: The deadband compensator inserted ahead of the plant model of Figure 3.1. The error-based ADRC, drawn in Figure 3.1, now produces the normalized command u_{comp}, the compensator converts it into the valve current u_{valve} that jumps over the constant deadband (Section 3.2.2), and the useful actuation u_{LTI} enters the LTI core. The output saturation of Figure 3.1 is omitted here to focus on the deadband.

Deadband Model

With a symmetric deadband of half-width d [18], the useful actuation signal entering the LTI plant is

$$u_{\text{LTI}} = \begin{cases} \dfrac{u_{\text{valve}} + d}{V_{\text{max}} - d}, & u_{\text{valve}} < -d \\ 0, & |u_{\text{valve}}| \leq d \\ \dfrac{u_{\text{valve}} - d}{V_{\text{max}} - d}, & u_{\text{valve}} > d \end{cases} \tag{3.3}$$

Inside the deadband the useful actuation is zero, because no flow reaches the cylinder, regardless of whether the spool has moved. Outside the deadband the relation between current and useful actuation is linear, with a slope chosen so that $u_{\text{valve}} = \pm V_{\text{max}}$ produces $u_{\text{LTI}} = \pm 1$.

Ideal Inverse

The ideal inverse of (3.3) produces the current that achieves the commanded actuation:

$$u_{\text{valve}} = u_{\text{comp}} \cdot (V_{\text{max}} - d) + d \cdot \text{sgn}(u_{\text{comp}}). \tag{3.4}$$

The first term scales the controller's range $[-1, 1]$ into the linear current range $[-(V_{\text{max}} - d), V_{\text{max}} - d]$. The second term jumps over the deadband. Substituting (3.4) into (3.3) gives $u_{\text{LTI}} = u_{\text{comp}}$, which is the desired cancellation. However, the inverse has a jump discontinuity at $u_{\text{comp}} = 0$, where the current jumps by $2d$ whenever the controller output crosses zero. If the control signal oscillates near zero, as it will during steady-state tracking, this discontinuity drives the valve at high frequency and corresponds to an effectively infinite bandwidth demand.

Practical Compensation

The ideal inverse needs one adjustment to be usable in practice. Equation (3.4) is the standard dead-zone inverse [15]. To remove its discontinuity at zero, it is made piecewise-linear within a window of half-width g around zero (Figure 3.4), following a gradient of slope $\hat{d}/g$ from the origin to the deadband edge $\hat{d}$ at $u_{\mathrm{comp}} = \pm g$:

$$u_{\mathrm{valve}} = \begin{cases} \hat{d} + (u_{\mathrm{comp}} - g)\,\dfrac{V_{\max} - \hat{d}}{1 - g}, & u_{\mathrm{comp}} > g \\ \dfrac{\hat{d}}{g}\,u_{\mathrm{comp}}, & |u_{\mathrm{comp}}| \leq g \\ -\hat{d} + (u_{\mathrm{comp}} + g)\,\dfrac{V_{\max} - \hat{d}}{1 - g}, & u_{\mathrm{comp}} < -g \end{cases} \tag{3.5}$$

Outside the window the map runs linearly from the edge $\hat{d}$ to $\pm V_{\max}$ at $u_{\mathrm{comp}} = \pm 1$. The compensator is therefore continuous, at the price of a residual deadband of half-width g in the command that the controller has to handle.

The window half-width g controls the width of the smoothing region around zero and determines the size of the residual deadband. As $g \to 0$ the gradient steepens toward the discontinuous ideal inverse, shrinking the residual deadband but reviving the jump it is supposed to remove. As g grows the gradient softens but the residual deadband widens, until at $g = \hat{d}/V_{\max}$ the map reduces to $u_{\mathrm{valve}} = V_{\max}\,u_{\mathrm{comp}}$ and no compensation remains. A usable g lies between these extremes. This thesis fixes $g = 0.1$ for the whole vehicle set, so the gradient occupies the band $|u_{\mathrm{comp}}| \leq 0.1$. Ideally g would depend on the deadband size and on how fast the actuator can move, since that limits how steep the gradient can be, but finding such a rule is left to future work. The simulation tracking result in Section 3.6.2 verifies that $g = 0.1$ is adequate.

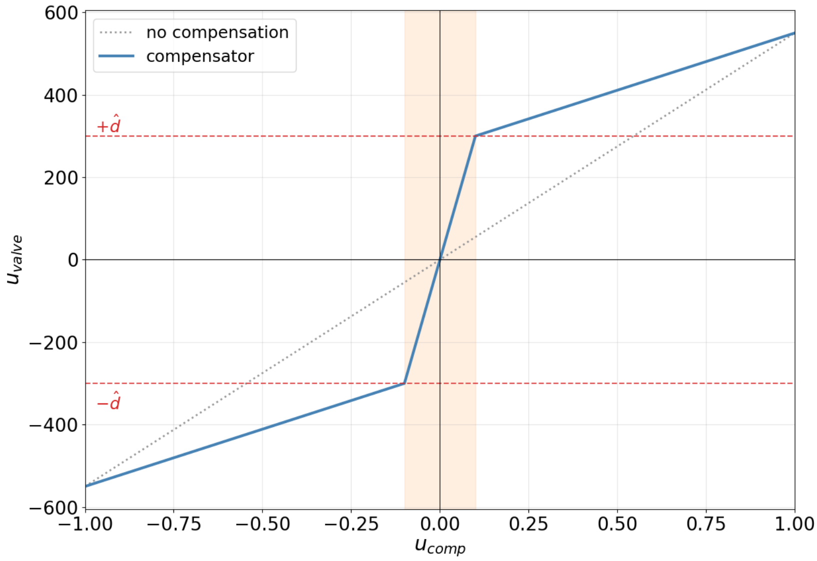

Figure 3.4: The practical compensator of (3.5). The linear in-window gradient replaces the discontinuous jump of the ideal inverse and leaves a residual deadband of half-width g.

3.4.2 Identifying the half-width

The half-width $\hat{d}$ of the deadband model (3.3), which the compensator (3.5) needs, is identified by the algorithm developed below and validated against two alternatives (Appendix B). The experiment is described first, then the identification algorithm, and finally the comparison that selects it.

3.4.2.1 Experiment

The three algorithms are compared on the seven 2nd-order Type-1 synthetic plants of Section 3.3.1. An artificial deadband of half-width $d = 0.5$ is added in front of each plant, a slow input ramp $u_{\text{valve}}(t) = \alpha\, t$ with slope $\alpha = 0.002\,\text{s}^{-1}$ is applied, and the recorded $(u_{\text{valve}}, \gamma)$ pairs are passed to the algorithm under test. Every plant is run at three noise levels, $\sigma \in \{0.001, 0.01, 0.1\}$ rad (low / medium / high), to test how sensitive each method is to measurement noise. The identified $\hat{d}$ feeds the compensator of (3.5).

3.4.2.2 Identification algorithm

The deadband half-width d is the input value at which the actuator begins to respond. In the ramp data the measured angle γ stays flat while the input is below d, then starts to rise once the input passes d. Finding d means finding the single input value where the data changes from flat to rising. This is a segmented regression problem, and that changeover point is called the breakpoint.

Lerman [12] finds the breakpoint with a grid search. A grid of candidate breakpoints is placed along the input axis, and each candidate is tried in turn. For a given candidate, the samples on its left and the samples on its right are fitted separately by least squares, and the total error of the two fits is recorded. The candidate with the smallest total error is taken as the breakpoint. The reason this works is that at the correct breakpoint each side fits data of its own shape and the error is small, whereas a wrong split forces one side to fit data of the wrong shape, which raises the error.

Lerman fits a straight line on each side. The contribution here is to replace those generic segments with the shapes the plant dictates, a constant in the dead region, where the actuator does not move, and a degree-2 polynomial in the active region. The polynomial is quadratic because the Type-1 plant $G(s) = K/[s(\tau s+1)]$ (Section 3.2.3) integrates the input ramp into a quadratic, as derived in Appendix B. Writing $s_i = u_{\text{valve},i} - d_k$ for the input measured past a candidate breakpoint d_k at sample index k, the model fitted at each candidate is

$$\hat{\gamma}_i(k, \mathbf{a}, c) = \begin{cases} c, & i \leq k \\ a_0 + a_1 s_i + a_2 s_i^2, & i > k \end{cases} \tag{3.6}$$

A constant fits the flat left window and a quadratic the rising right window, each by ordinary least squares. The estimate $\hat{d}$ is the candidate split that minimises the combined root-mean-square error of the two segments, with the full least-squares objective given in Appendix B.

The three benchmarked algorithms

A-pos applies the fit of (3.6) directly to the raw measured angle, with no differentiation and no pre-filter. Appendix B describes two alternatives that were tested against this algorithm: A-vel, a slight adaptation that differentiates the angle first, and a threshold detector, a simple baseline that triggers when the angle leaves the noise band.

3.4.2.3 Algorithm comparison

This is the final step of the deadband identification, selecting which of the three algorithms feeds the auto-tuning pipeline (Section 3.6). Table 3.1 reports the bias, standard deviation, and RMSE of the identification error $\hat{d} - d$ for each algorithm at each noise level.

Table 3.1: Benchmark of the three candidate deadband-identification algorithms, used to select the one carried into the auto-tuning pipeline (Section 3.6).

Noise	Algorithm	Bias	Std	RMSE
Low	A-vel	+0.0010	0.0015	0.0017
	A-pos	+0.0010	0.0014	0.0016
	Threshold	+0.0027	0.0007	0.0028
Medium	A-vel	−0.0006	0.0020	0.0019
	A-pos	+0.0010	0.0014	0.0016
	Threshold	+0.0070	0.0010	0.0071
High	A-vel	−0.0056	0.0000	0.0056
	A-pos	+0.0009	0.0014	0.0016
	Threshold	+0.0210	0.0010	0.0210

A-pos is the only method that maintains consistent accuracy as the noise grows, with its RMSE holding near 0.0016 at every level. It is therefore chosen for the auto-tuning pipeline (Section 3.6), on both accuracy and simplicity: it works directly on the raw measured angle with no differentiation or filtering and needs no additional tuneable parameters.

To test whether A-pos also works on the simulation models, it is run on all nine Modelica vehicle configurations. The true deadband is not known here, so the fit can only be judged visually, but each identified $\hat{d}$ sits at the point where the angle starts to move (Figure 3.5).

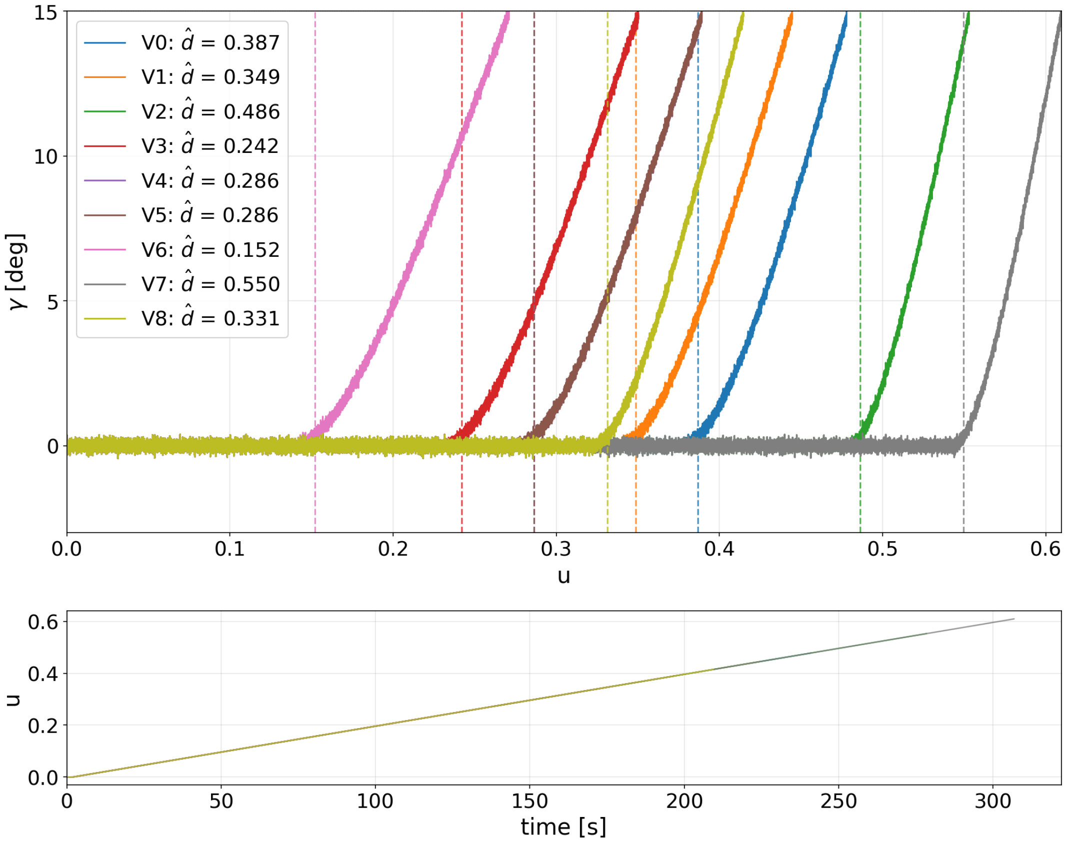

Figure 3.5: Results of the A-pos deadband identification on the nine Modelica vehicle configurations, with the identified half-width $\hat{d}$ marked on each. The identification was run with measurement noise of $\sigma = 0.002\,\text{rad}$.

3.5 Controller Tuning

This section is the *Controller Tuning* stage, marked in red in Figure 1.2. It turns the plant gain K and lag τ of the plant model (3.1) into ADRC settings. The rule is developed on the synthetic test environment, where K and τ are known by construction (Section 3.3.1). The input gain b_0 is found analytically (Section 3.5.1) and the remaining settings empirically (Section 3.5.2), and the resulting rule is applied in the auto-tuning stage (Section 3.6).

3.5.1 Analytic Tuning of b_0

The input gain b_0 is the only ADRC setting fixed analytically from the plant model. The rest are tuned empirically in Section 3.5.2. For the deadband-compensated second-order Type-1 plant (3.1), b_0 is derived from the gain K and lag τ. The exact expression depends on the ADRC model order N (Chapter 2).

For a 2nd-order ADRC ($N = 2$), writing (3.1) as the differential equation

$$\tau\,\ddot{y} + \dot{y} = K\,u \quad\Longleftrightarrow\quad \ddot{y} = -\frac{1}{\tau}\,\dot{y} + \frac{K}{\tau}\,u \tag{3.7}$$

and comparing with the ADRC plant form (2.3) absorbs the $-\dot{y}/\tau$ term into the total disturbance $f(t)$, leaving

$$b_0 = \frac{K}{\tau}. \tag{3.8}$$

A 1st-order ADRC ($N = 1$) models the output rate directly as $\dot{y} = f(t) + b_0\, u$, the counterpart of (2.3) for $N = 1$, and needs only the gain,

$$b_0 = K. \tag{3.9}$$

3.5.2 Empirical Tuning of ω_{CL} and k_{ESO}

Two ADRC settings have no analytic value, the closed-loop bandwidth ω_{CL} and the ratio k_{ESO} that sets the observer bandwidth relative to it (Chapter 2). This section fixes them empirically, by running the error-based ADRC over the synthetic plants for a grid of ω_{CL} and k_{ESO} values and scoring each run. The reference signal that drives the runs (Section 3.5.2.1) and the two metrics that score them (Section 3.5.2.2) are defined first, then the sweeps produce the rule (Section 3.5.2.3).

3.5.2.1 Reference Signal

The reference must span the steering angles and rates the controller will face. The company specifies a maximum steering angle of 0.45 rad and a maximum steering rate of 10°/s, which roughly matches field measurements of heavy articulated vehicles in the literature [4].

The reference frequency is swept from 0.02 Hz to 0.3 Hz to cover the operating range, from slow long curves where the articulation angle changes over about ten seconds to fast steering corrections with a period of a few seconds. The amplitude is reduced as the frequency rises, so the peak steering rate is highest at the slow end, reaching the $\approx 10°/\mathrm{s}$ bound at 0.02 Hz, and drops to $\approx 1.5°/\mathrm{s}$ at 0.3 Hz. The resulting sweep is shown in Figure 3.6.

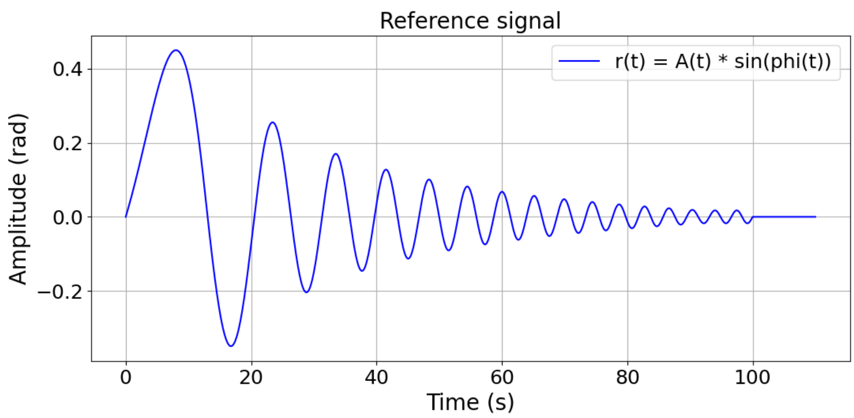

Figure 3.6: Sine-sweep reference used for all closed-loop tuning experiments. Its amplitude is reduced as the frequency rises, so the peak steering rate is highest at the slow start of the sweep and lowest at the fast end.

3.5.2.2 Metrics

Normalised root-mean-square error (NRMSE). Because the reference amplitude drops as the frequency rises, a single root-mean-square error taken over the full run would be dominated by the slow end where the reference is large. A tracking error of equal absolute size at the fast end would barely register, even though it represents a much larger fraction of the local amplitude. To weight the slow and fast parts of the sweep equally, the tracking error is therefore computed per reference half-period and normalised by the local amplitude.

Let $\{t_k\}_{k=0}^{M}$ denote the zero crossings of the reference $r(t)$ over the run, ordered in time. Each interval $W_k = [t_k, t_{k+1}]$ spans one half-period of the reference and contains a single peak. The normalised root-mean-square error on that interval is

$$\mathrm{NRMSE}_k = \frac{1}{\max_{t \in W_k} |r(t)|} \sqrt{\frac{1}{t_{k+1} - t_k} \int_{t_k}^{t_{k+1}} \Big(r(t) - y(t)\Big)^2 \, dt}. \tag{3.10}$$

Dividing by the local peak expresses the error as a fraction of the local amplitude. The scalar reported for a run is the mean over all half-periods,

$$\mathrm{NRMSE} = \frac{1}{M} \sum_{k=0}^{M-1} \mathrm{NRMSE}_k, \tag{3.11}$$

where M is the number of half-periods. The metric is frequency-dependent, penalising the same tracking lag in seconds more heavily at higher reference frequencies.

High-frequency energy ratio (HFR). Control effort spent on content that the reference cannot have demanded acts as a proxy for actuator wear and for the noise sensitivity of the loop. The reference sweep of Section 3.5.2.1 runs up to $f_{\mathrm{ref,max}} = 0.3\,\mathrm{Hz}$, so the legitimate command content of the control signal is also bounded by this frequency. Anything in the spectrum of u above a cutoff $f_c = 2\, f_{\mathrm{ref,max}} = 0.6\,\mathrm{Hz}$ is then either measurement-noise-driven chatter or harmonic content from a near bang-bang regime, and neither is part of the underlying command. The HFR reports the fraction of the total energy of u that sits above this cutoff,

$$\mathrm{HFR} = \frac{\int_{f_c}^{f_{\mathrm{Nyq}}} |U(f)|^2 \, df}{\int_0^{f_{\mathrm{Nyq}}} |U(f)|^2 \, df}, \tag{3.12}$$

where $U(f)$ is the discrete Fourier transform of u over the run and f_{Nyq} is the Nyquist frequency at the controller sample rate.

A value of $\mathrm{HFR} \approx 0$ means almost all control effort sits in the band the reference can ask for, so the control signal is close to a clean command. Larger values mean more effort goes above that band, whether from measurement-noise chatter or from the harmonics of a near bang-bang regime. The ratio is invariant to the plant gain, so it scores comparably across the plant family, and it is used to rank tunings against each other rather than as an absolute account of chatter energy.

3.5.2.3 Parameter Sweeps

Bandwidth parameterization (Section 2.3) leaves ω_{CL} and k_{ESO} to set. The common choice of ω_{CL} from the settling time of a step response [11, Eq. 3.21] does not fit here, since the error-based ADRC is sensitive to step inputs (Chapter 2) and the steering loop sees a slowly changing reference rather than a step. The two are chosen empirically, sweeping ω_{CL} on a log scale from 0.1 rad/s to 6 rad/s and k_{ESO} over $[3, 5]$, scored by the metrics of Section 3.5.2.2 on the reference sweep of Section 3.5.2.1. Two acceptance thresholds are used throughout, NRMSE ≤ 0.5 for tracking and HFR ≤ 0.01 for control smoothness. Both are calibration anchors read from the tracking responses rather than values taken from a specification, NRMSE $= 0.5$ marking where tracking is still visually acceptable and HFR $= 0.01$ where control smoothness is still acceptable. The responses they are read from are shown in Appendix C.

Noise-free baseline: the model-order split. A first sweep without sensor noise isolates how the metrics depend on the controller settings alone, using a 2nd-order ADRC on the synthetic plants of (3.1).

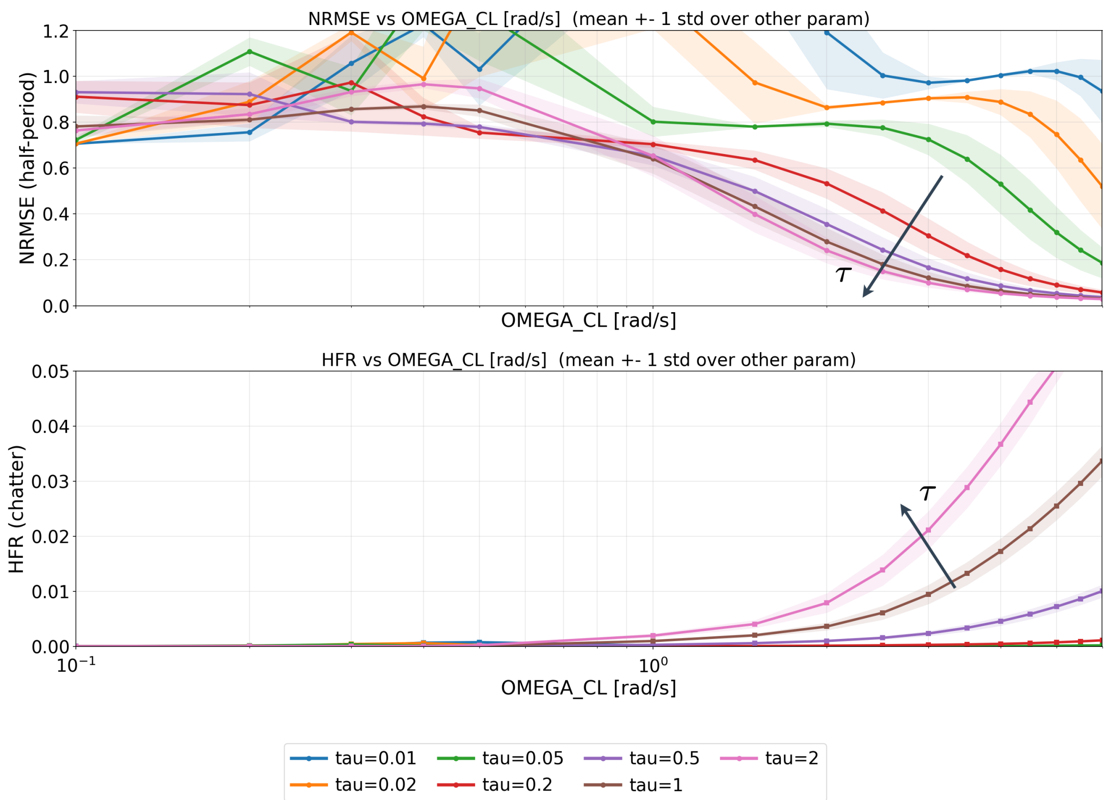

Figure 3.7: Noise-free sweep of the 2nd-order ADRC over the closed-loop bandwidth, one curve per plant time constant τ in the synthetic family. Each curve is the mean over $k_{\text{ESO}} \in [3, 5]$ and the shaded band is ± 1 standard deviation across k_{ESO}.

The sweep in Figure 3.7 splits the plants in two. The slower plants ($\tau \geq 0.5$ s) reach a low NRMSE already near $\omega_{\mathrm{CL}} = 0.5\,\mathrm{rad/s}$, while the faster plants ($\tau < 0.5\,\mathrm{s}$) only do so at much higher ω_{CL}, and on those the HFR stays near 0 throughout. The cause is the input gain. Substituting the bandwidth-parameterised gains (2.15) into the control law (2.13) gives

$$u(t) = \frac{1}{b_0}\Big(\omega_{\mathrm{CL}}^2 \hat{x}_1 + 2\,\omega_{\mathrm{CL}}\,\hat{x}_2 + \hat{x}_3\Big), \tag{3.13}$$

which divides by b_0. With $b_0 = K/\tau$ (3.8), a small τ inflates b_0, shrinks the effective feedback gain, and leaves the fast plants lagging until ω_{CL} is raised far enough to scale the gains back up. That high bandwidth then pairs with a noise-sensitive high observer bandwidth.

This motivates dropping to a 1st-order ADRC on the fast plants. It takes $b_0 = K$ (3.9), with no τ in the denominator, and its control law

$$u(t) = \frac{1}{b_0}\Big(\omega_{\mathrm{CL}} \hat{x}_1 + \hat{x}_2\Big), \tag{3.14}$$

drops one state estimate and a power of ω_{CL}, so it transmits less observer noise. The residual $-\dot{y}/\tau$ term of (3.7) is absorbed into the total disturbance the ESO already estimates, while the slower plants keep the 2nd-order ADRC. The split between the two is located by the noisy sweeps below.

Measurement noise: half-gain tuning. Adding white Gaussian noise at $\sigma = 0.002\,\mathrm{rad}$ to the measured angle drives the observer gains, which scale as ω_{O}, ω_{O}^2, and ω_{O}^3 (2.17), into chatter at the bandwidths needed for acceptable tracking. Half-gain tuning (Section 2.3.3) is therefore applied from here on. The 2nd-order ADRC is then swept over $\tau \in [0.2, 2.0]\,\mathrm{s}$ with k_{ESO} over $[3, 5]$.

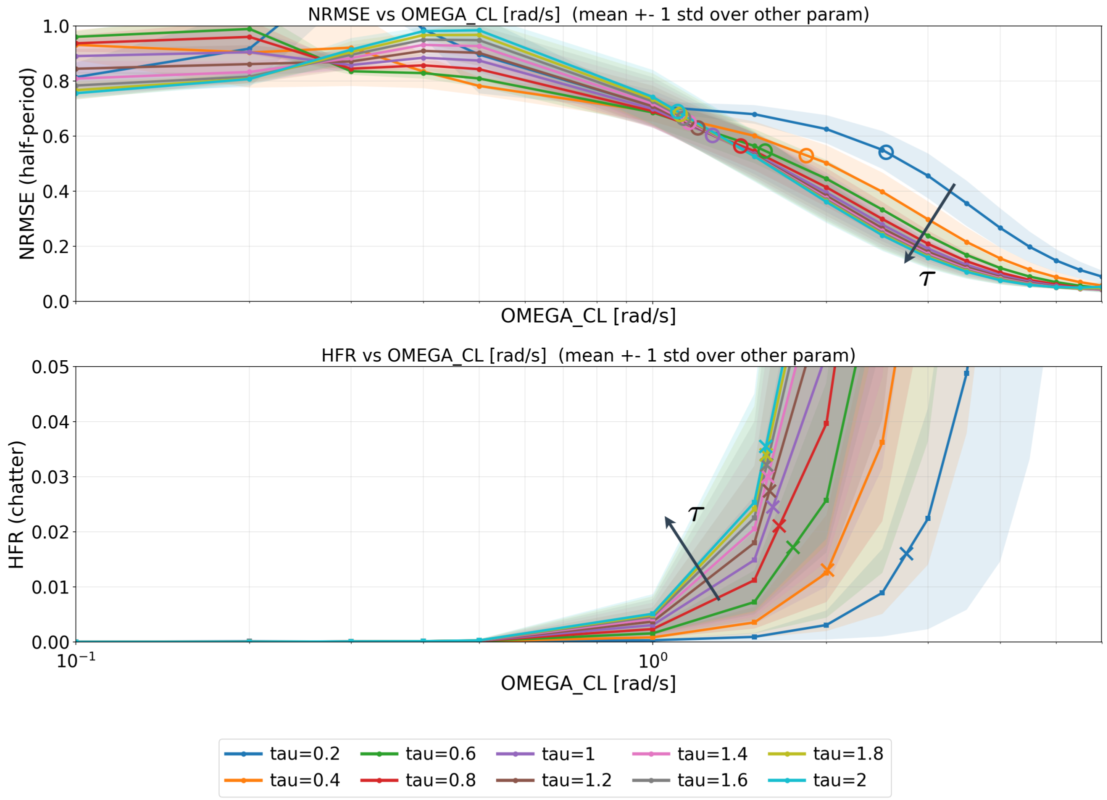

Figure 3.8: Sweep of the 2nd-order ADRC over ω_{CL} with measurement noise and half-gain tuning, $k_{\mathrm{ESO}} \in [3, 5]$, and $\tau \in [0.2, 2.0]$ s. The coloured crosses mark the ω_{CL} at which each τ first reaches the tracking anchor NRMSE $= 0.5$, the circles the ω_{CL} at which it first reaches the smoothness anchor HFR $= 0.01$ (Appendix C).

For each τ, the crosses in Figure 3.8 mark the ω_{CL} where tracking first reaches NRMSE $= 0.5$ and the circles where smoothness first reaches HFR $= 0.01$. On the slower plants the two cannot be met at the same ω_{CL}, so noise sensitivity is prioritised and the operating point for each τ is taken from the HFR $= 0.01$ circles at $k_{\mathrm{ESO}} = 3$ (Appendix C).

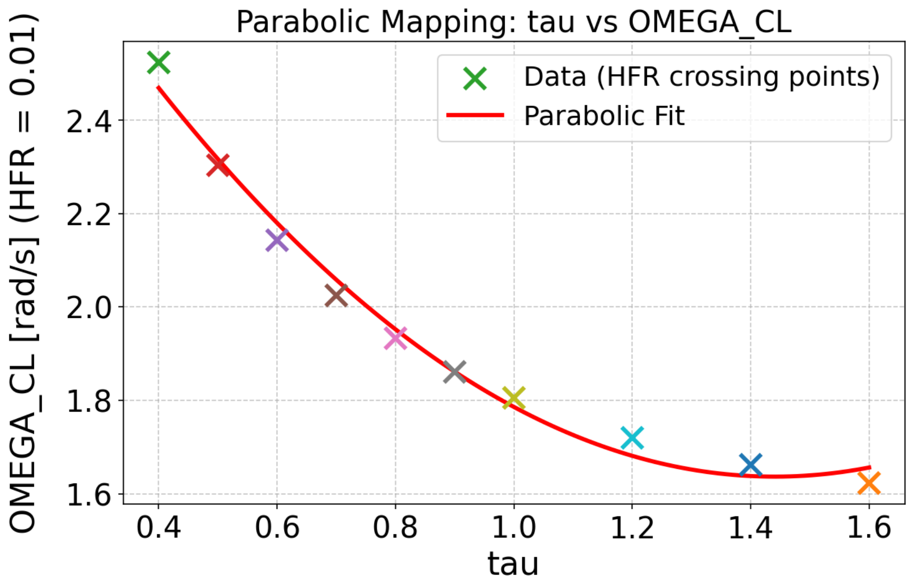

Figure 3.9: The $(\tau, \omega_{\mathrm{CL}})$ operating points where the HFR first reaches 0.01, with the fitted second-order polynomial (3.15) that becomes the 2nd-order tuning rule.

Fitting a second-order polynomial to those $(\tau, \omega_{\mathrm{CL}})$ points (Figure 3.9) gives the empirical 2nd-order rule

$$\omega_{\mathrm{CL}} = 0.77\,\tau^2 - 2.21\,\tau + 3.23, \tag{3.15}$$

with τ in seconds and ω_{CL} in rad/s. The coefficients are calibrated to this noise level, reference sweep, and HFR ceiling rather than being universal. How far the rule moves under other choices is discussed in Section 5.3.

The 1st-order rule (fast plants). The 1st-order ADRC is swept over the fast plants ($\tau \leq 0.6\,\mathrm{s}$) at the same noise level and with half-gain tuning.

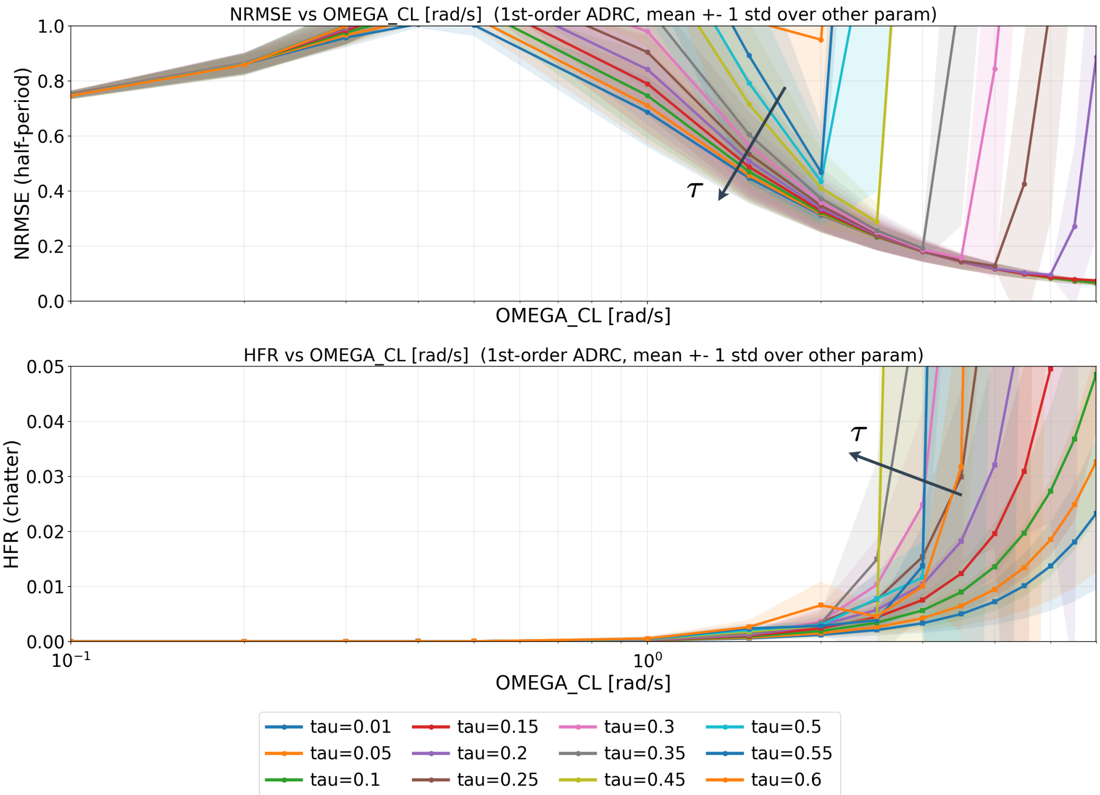

Figure 3.10: Sweep of the 1st-order ADRC over ω_{CL} with measurement noise and half-gain tuning on the fast plants ($\tau \leq 0.6\,\mathrm{s}$), $k_{\mathrm{ESO}} \in [3, 5]$. The shaded band is ± 1 standard deviation across k_{ESO}.

At $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$ every fast plant sits below both thresholds (Figure 3.10), including across $k_{\mathrm{ESO}} \in [3, 5]$, while raising the bandwidth further pushes the NRMSE or the HFR up on at least one plant. The fast-plant rule is therefore $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$, $k_{\mathrm{ESO}} = 5$, and half-gain tuning. The NRMSE of the $\tau = 0.5\,\mathrm{s}$ plant begins to rise sharply at this bandwidth, which places the order split at $\tau = 0.5\,\mathrm{s}$ (the 1st-order tracking responses are shown in Appendix C).

Resulting tuning rule. Table 3.2 collects the rule, applied in the auto-tuning stage (Section 3.6) once K and τ are identified from a standstill experiment.

Table 3.2: Tuning rule produced by the parameter sweeps, with half-gain tuning applied in both regimes.

Regime	ADRC order N	b_0	ω_{CL}	k_{ESO}
$\tau \leq 0.5\,\mathrm{s}$	1	K	$2\,\mathrm{rad/s}$	5
$\tau > 0.5\,\mathrm{s}$	2	K/τ	$0.77\,\tau^2 - 2.21\,\tau + 3.23$	3

3.6 Auto-Tuning and Plant Identification

This section is the *Auto Tuning / Identification* stage, marked in purple in Figure 1.2. It identifies the plant parameters the tuning rule needs, then runs the full auto-tuning pipeline on the simulation model. Section 3.6.1 develops the identification, and Section 3.6.2 reports the pipeline run on the nine vehicles, which is then stress-tested under model mismatch (Section 3.7) and deployed on the real vehicle (Chapter 4).

3.6.1 Plant Identification

The tuning rule of Section 3.5 takes the plant gain K and the dominant lag τ as input. On the synthetic family these values are prescribed by construction. On a real plant they have to be estimated. The estimator below extracts K and τ from a step experiment, and from them computes b_0 as the input to the tuning rule. It is first validated on the synthetic family of Section 3.3.1.

Step experiment The input steps from zero to a fixed amplitude $u_{\max}$ and then back to zero. Returning to zero stops the integrating output from growing further and keeps the articulation angle away from its mechanical limits. The step response of (3.1) during the active part of the step is

$$y(t;\, K, \tau) = K\, u_{\max}\Big[t - \tau\Big(1 - e^{-t/\tau}\Big)\Big]\,. \tag{3.16}$$

Nonlinear least-squares (NLS) estimator Recovering a low-order model from step-response data is a standard problem in process identification, not one this thesis has to solve from scratch. Nonlinear least-squares curve-fitting of the step-response model is one of its established solutions, which Cox et al. [2] compare against other methods for first- and second-order-plus-dead-time plants. Here the dead-time term is dropped under the negligible-delay assumption (Section 3.1), and the same least-squares fit is applied to the second-order Type-1 model. K and τ are recovered by fitting the step response (3.16) to the N sampled angle measurements $\varphi(t_i)$. The fit minimises the sum of squared residuals,

$$J(K, \tau) = \sum_{i=1}^{N}\Big[\varphi(t_i) - y(t_i;\, K, \tau)\Big]^2, \tag{3.17}$$

giving the estimate

$$(\hat{K}, \hat{\tau}) = \arg\min_{K,\,\tau}\; J(K, \tau). \tag{3.18}$$

Since τ enters (3.16) nonlinearly through the exponential, (3.18) has no closed form. The minimisation is solved iteratively with the Levenberg–Marquardt algorithm [13], invoked through SciPy's `least_squares` routine [16].

The iteration needs a starting guess. It is obtained by inspecting the asymptote of (3.16): at large t the response approaches a straight line whose slope is $K\, u_{\max}$ and whose time-axis offset is τ. Reading these two values from the data gives initial K_0 and τ_0 that are already close to the true parameters, and the iteration then converges within a few steps on every plant in the synthetic sweep.

Synthetic validation The estimator is validated on the synthetic family of Section 3.3.1, swept over both K and τ, with white Gaussian measurement noise added on top.

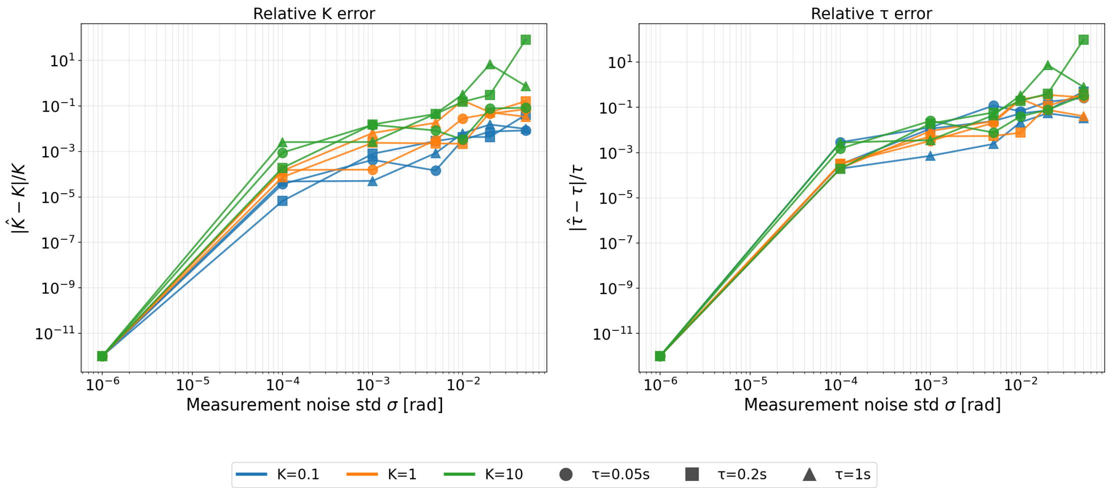

Figure 3.11: Relative error of the nonlinear least-squares estimator (3.18) against measurement noise, for several plant gains K and time constants τ.

Figure 3.11 shows both relative errors growing with the measurement-noise level σ. The K-error grows with the plant gain, because higher-gain plants reach the angle limit sooner and yield fewer data points for the fit, while the τ-error is largely insensitive to both K and τ.

At the noise level of practical interest, $\sigma = 10^{-3}$ rad, both errors stay around 10^{-3} for the low-gain plants and around 10^{-1} for the high-gain plants. The estimator is therefore accurate enough at the noise levels expected on the simulation and real vehicle, and is used as the front end of the tuning pipeline from here.

3.6.2 Simulation Model Verification

On the simulation model the plant parameters K and τ are not known in advance, so the full auto-tuning pipeline has to be run end to end. The chain of deadband identification, plant identification, and controller tuning rule is applied to the nine validation vehicles of Table A.3 (Appendix A). The resulting deadband half-width $\hat{d}$, plant parameters $\hat{K}$ and $\hat{\tau}$, and ADRC settings N, b_0, and ω_{CL} are listed in Table 3.3.

Table 3.3: Auto-tuning pipeline output on the nine simulation vehicles of Table A.3. The deadband half-width $\hat{d}$ is identified by A-pos (Section 3.4.2.2). The plant parameters $\hat{K}$ and $\hat{\tau}$ are produced by the nonlinear least-squares estimator of Section 3.6.1. The ADRC order N, critical gain b_0, and closed-loop bandwidth ω_{CL} follow from the tuning rule of Section 3.5.2. Half-gain tuning and $k_{\mathrm{ESO}} = 5$ are applied throughout, as prescribed by the rule for $\hat{\tau} \leq 0.5\,\mathrm{s}$.

ID	$\hat{d}$	$\hat{K}$	$\hat{\tau}$ [s]	N	b_0	ω_{CL} [rad/s]
V0	0.3868	0.1478	0.0734	1	0.1478	2.000
V1	0.3488	0.1477	0.0751	1	0.1477	2.000
V2	0.4863	0.2762	0.0395	1	0.2762	2.000
V3	0.2422	0.1477	0.0751	1	0.1477	2.000
V4	0.2863	0.1478	0.0734	1	0.1478	2.000
V5	0.2863	0.1478	0.0726	1	0.1478	2.000
V6	0.1522	0.1478	0.0731	1	0.1478	2.000
V7	0.5496	0.2764	0.0405	1	0.2764	2.000
V8	0.3312	0.2765	0.0362	1	0.2765	2.000

The closed loop is then driven by the sine sweep reference of Section 3.5.2.1 on all nine vehicles, with the angle sensor carrying the same white Gaussian noise level as in the synthetic sweep of Section 3.5.2. The resulting NRMSE and HFR are listed in Table 3.4.

Table 3.4: Closed-loop tracking metrics for the auto-tuned controllers on the nine simulation vehicles.

ID	NRMSE	HFR
V0	0.18985	0.0034
V1	0.18864	0.0028
V2	0.26840	0.0156
V3	0.18735	0.0020
V4	0.18777	0.0023
V5	0.18408	0.0022
V6	0.18890	0.0018
V7	0.28508	0.0195
V8	0.24213	0.0081

Figure 3.12 overlays the tracking responses of all nine vehicles, confirming the NRMSE numbers in Table 3.4.

V7 and V8 show some chatter at the start of the sweep that disappears after one period. Of those two, only V7's chatter is large enough to push its HFR above the threshold of 0.01 defined in Section 3.5.2. V2 also exceeds the threshold despite no visible chatter.

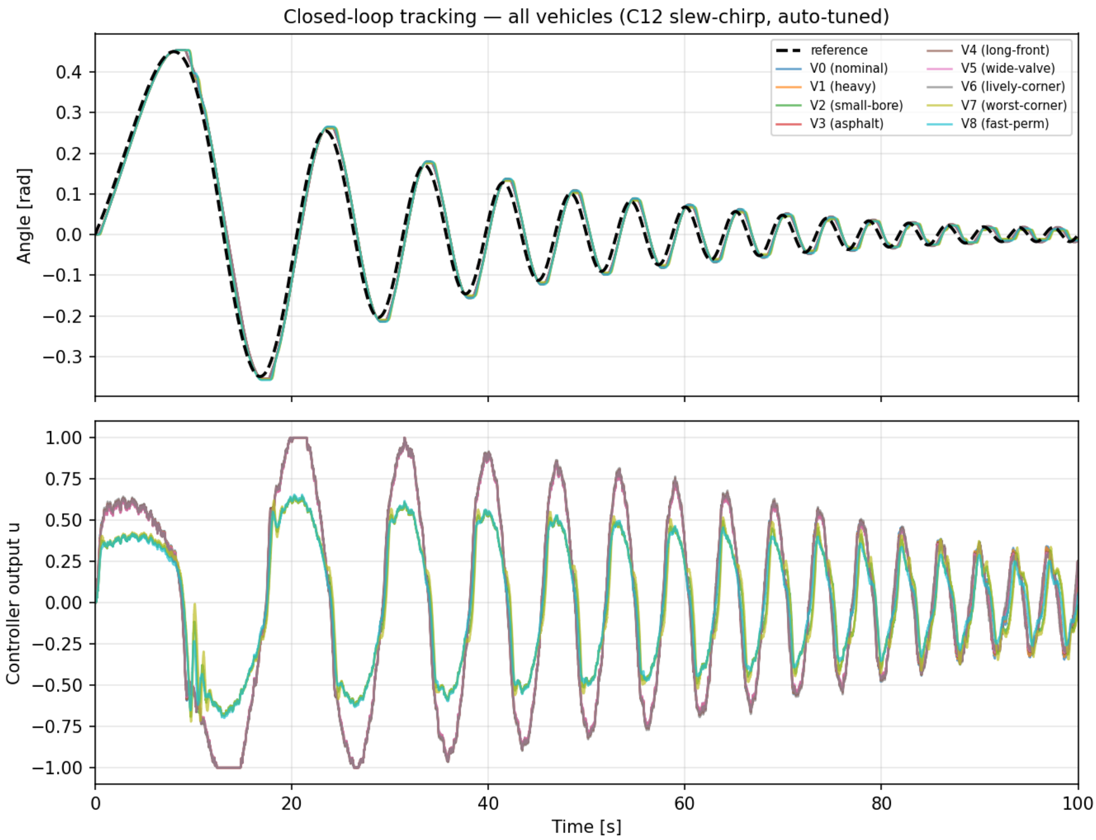

Figure 3.12: Closed-loop tracking response of the auto-tuned controllers on the nine simulation vehicles of Table A.3.

The auto-tuned controllers therefore meet the design target in the simulation environment. Their robustness to the standstill-to-driving operating-point gap is verified by the model-mismatch study in Section 3.7, and the integrated pipeline is then deployed on the real vehicle in Chapter 4.

3.7 Deliberate Simulation Model Mismatch Verification

This section is the *Deliberate Simulation Model Mismatch* stage, marked in light grey in Figure 1.2. It tests how the auto-tuned controllers hold up when the plant is moved away from the operating point used for identification, giving the closest in-simulation evidence for the robustness claim made in Chapter 5.

3.7.1 Mismatch setup

The plant is identified at standstill, so K and τ describe it at that one operating point. In operation the effective gain and lag shift as payload and ground friction change, and the ESO is meant to absorb that shift as part of the total disturbance

$f(t)$ of (2.3). This section tests that property on the plant family by perturbing the simulation model after the controller has been tuned.

The controllers from Section 3.6.2 are kept fixed. The mass and friction of the simulation model are then changed away from the nominal values of Table A.1 that were used during identification, creating a mismatch between the model the controller was tuned on and the model it now runs against. Each of the nine vehicles V0–V8 runs with one mass scaling and one friction scaling. $+M$ raises the front and rear body masses by 20 % and $-M$ lowers them by 20 %. $+F$ raises the longitudinal, lateral, and rotational friction coefficients by 20 % and $-F$ lowers them by 20 %.

All four sign combinations $\pm M \pm F$ appear at least once across the vehicle set. The same sine sweep reference (Section 3.5.2.1) and noise level as in Section 3.6.2 are applied.

3.7.2 Results

Table 3.5: Closed-loop tracking metrics on the nine mismatched simulation vehicles. The mass/friction pattern $\pm M \pm F$ indicates whether mass and friction are scaled above (+) or below (−) the nominal values of Table A.1.

ID	Mismatch	NRMSE	HFR
V0	+M+F	0.24092 (+0.05107)	0.0060 (+0.0026)
V1	-M+F	0.18198 (−0.00666)	0.0027 (−0.0001)
V2	+M-F	0.26259 (−0.00581)	0.0184 (+0.0028)
V3	-M-F	0.11671 (−0.07064)	0.0029 (+0.0009)
V4	+M-F	0.18165 (−0.00612)	0.0022 (−0.0001)
V5	-M+F	0.17673 (−0.00735)	0.0021 (−0.0001)
V6	-M-F	0.11855 (−0.07035)	0.0041 (+0.0023)
V7	+M+F	0.35626 (+0.07118)	0.0223 (+0.0028)
V8	-M-F	0.14823 (−0.09390)	0.0082 (+0.0001)

Table 3.5 shows that the mismatch has only a small effect on the vehicles where mass and friction were perturbed with opposite signs. On the extreme cases the effect is more visible: when both perturbations are positive ($+M + F$) the NRMSE and HFR both rise, and when both are negative ($-M - F$) the NRMSE drops by roughly the same absolute amount while the HFR still rises. The overlaid tracking responses are shown in Figure 3.13.

The NRMSE drop under $-M-F$ and its rise under $+M+F$ in Table 3.5 both follow from the role of b_0 in the control law (3.14). At lower mass and friction the same valve current drives the angle of the vehicle faster after perturbation than with its configuration during identification. The identified plant gain $b_0 = K$ is therefore too small for this operating condition. The control law (3.14) scales the control output with $1/b_0$, so an undersized b_0 increases the control output in the same way as raising ω_{CL}. Because $\hat{\tau}$ is very small the controllers run as 1st-order ADRCs (Section 3.5.2), and according to the 1st-order sweep in Figure 3.10 raising ω_{CL} further lowers the NRMSE on these plants. A larger $\hat{\tau}$ would have risked crossing into the sharp HFR rise, but the small $\hat{\tau}$ leaves room before that happens. Conversely, at higher mass and friction the same valve current drives the angle more slowly after perturbation than with its configuration during identification. The identified plant gain $b_0 = K$ is therefore too large for this operating condition, so b_0 is oversized and reduces the control output in the same way as lowering ω_{CL}. The NRMSE therefore rises as it is shown in the same sweep. Mass and friction mismatch is therefore equivalent to shifting the tuning left or right on the ω_{CL} axis of the 1st-order sweep.

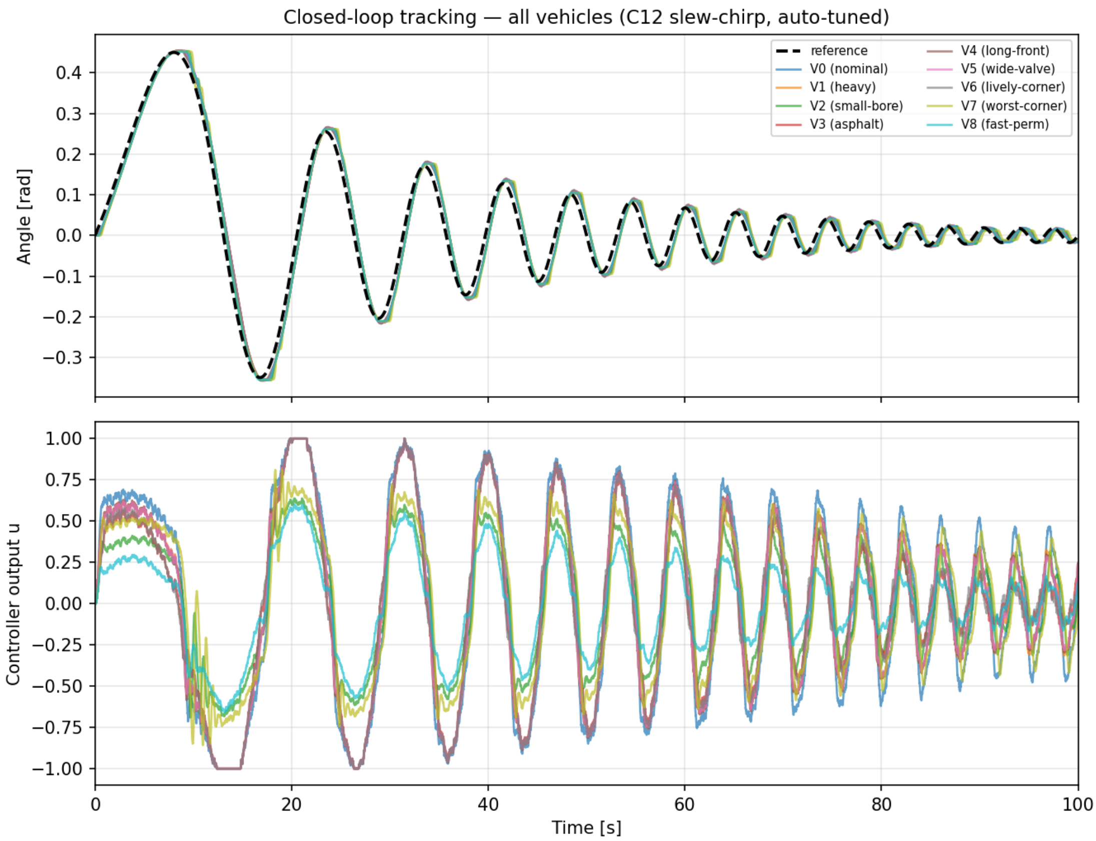

Figure 3.13: Closed-loop tracking response of the mismatched simulations of Table 3.5, overlaid for all nine vehicles.

The study therefore shows that the standstill-tuned ADRC absorbs a shift in plant gain, the form a change in payload or ground friction takes. A change in operating point that does not act as a gain shift is outside what it can show, a scope limit discussed in Section 5.3. The integrated pipeline is then deployed on the real vehicle in Chapter 4.

4

Real-Vehicle Deployment

This chapter is the *Real Vehicle Verification* stage, marked in dark grey in Figure 1.2. It applies the integrated pipeline to the real articulated vehicle, the third environment of Section 3.3.3. It runs the same three stages used on the simulation model, deadband identification, standstill step identification of K and τ, and bandwidth-parameterised tuning of an error-based ADRC, and drives the closed loop with the same sine sweep reference (Section 3.5.2.1). The sections that follow report each stage and then diagnose and discuss the closed-loop result, whose open problems are taken up in Chapter 5.

4.1 Test conditions

Access to the vehicle is limited. The hydraulic pump used in these tests is a temporary stand-in, fitted because the production pump was not yet available. It cannot be operated for long durations, which shortens every individual test, and it is powered by a non-constant voltage source whose output drops while the pump runs. The constant-flow assumption of the simulation model (Appendix A) therefore holds only approximately during these tests, although the production pump is expected to deliver the constant flow the assumption requires. The most informative data was collected at the beginning of each test, when the supply voltage was at its highest.

The angle sensor used during these tests is also a temporary unit, not the production sensor the vehicle will use in operation. It is more sensitive than the production sensor, and that sensitivity has two visible consequences. It responds to small body tilts of the vehicle, so the measured signal contains motion that is not articulation. After the vehicle stops it also shows a small settling wiggle to one side before coming to rest. Both effects are kept in mind when reading the traces in this chapter.

These conditions are stated once here so that the rest of the chapter can focus on what the data shows.

4.2 Deadband identification

The ramp-excitation algorithm of Section 3.4.2.2 is applied to the real vehicle. To stay within the available test window the ramp is started from 220 mA rather than zero, which skips the part of the input range where no motion is expected anyway.

A visible increase in the angle measurement appears at roughly 300 mA, and the algorithm returns a deadband half-width of $\hat{d} = 298.5\,\mathrm{mA}$, in agreement with the visual break (Figure 4.1).

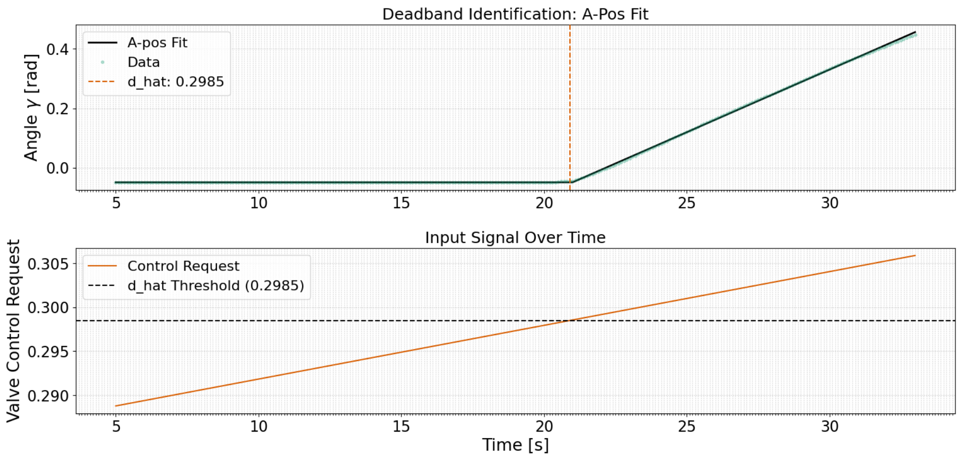

Figure 4.1: Deadband identification on the real vehicle. Raw ramp-excitation data, starting from 220 mA with overlayed identification. Regression fit returning $\hat{d} = 298.5\,\mathrm{mA}$.

The simulation vehicles drive a normalised input scaled to $[-1, 1]$, so the real $\hat{d}$ has to be divided by the 550 mA saturation current ($V_{\max}$) before it can be compared. This gives $\hat{d} \approx 0.54$, which sits within the range of normalised deadband half-widths identified across the nine simulation vehicles in Table 3.3, near its upper end. The real plant is therefore comparable to the simulated family on this parameter. The identified $\hat{d}$ is used as the deadband half-width of the practical inverse (Section 3.4.1) for every closed-loop test that follows.

4.3 Plant identification

A step from zero to the standard amplitude $u_{\max}$ is applied. The nonlinear least-squares estimator of Section 3.6.1 is run on the resulting angle trace and returns $\hat{K} = 0.145$ and $\hat{\tau} = 0.288\,\mathrm{s}$ (Figure 4.2).

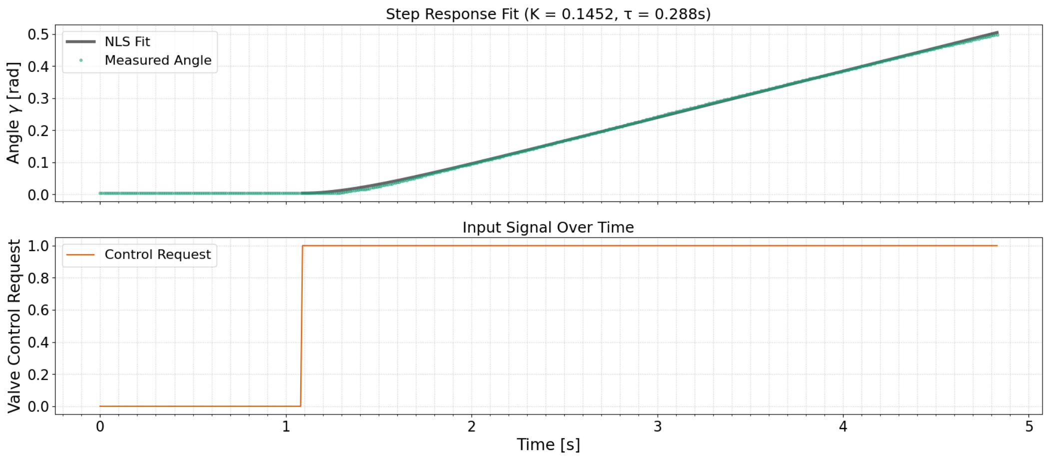

Figure 4.2: Real-vehicle step response (full window). The NLS fit (3.16) yields $\hat{K} = 0.145$ and $\hat{\tau} = 0.288\,\mathrm{s}$.

The gain $\hat{K} = 0.145$ is comparable to the lower-gain cluster of the simulation vehicles, whose identified gains sit around $\hat{K} \approx 0.145$ in Table 3.3. The lag $\hat{\tau} = 0.288\,\mathrm{s}$, by contrast, is several times larger than that of any simulated vehicle, whose lags all fall below $0.08\,\mathrm{s}$. Since $\hat{\tau} \leq 0.5\,\mathrm{s}$, the tuning rule of Section 3.5.2 selects a 1st-order ADRC with $b_0 = \hat{K}$, $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$, and $k_{\mathrm{ESO}} = 5$.

4.4 Closed-loop sweep test

The auto-tuned ADRC (1st-order, $b_0 = \hat{K}$, $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$, $k_{\mathrm{ESO}} = 5$) is deployed on the vehicle and driven by the sine sweep reference of Section 3.5.2.1. The closed-loop response is shown in Figure 4.3. The lower trace is the valve current request sent to the solenoid, that is, the signal at the output of the deadband compensator and not the raw controller output u. The large jumps visible at every zero crossing are therefore the deadband step inserted by the practical inverse of Section 3.4.1, not controller chatter.

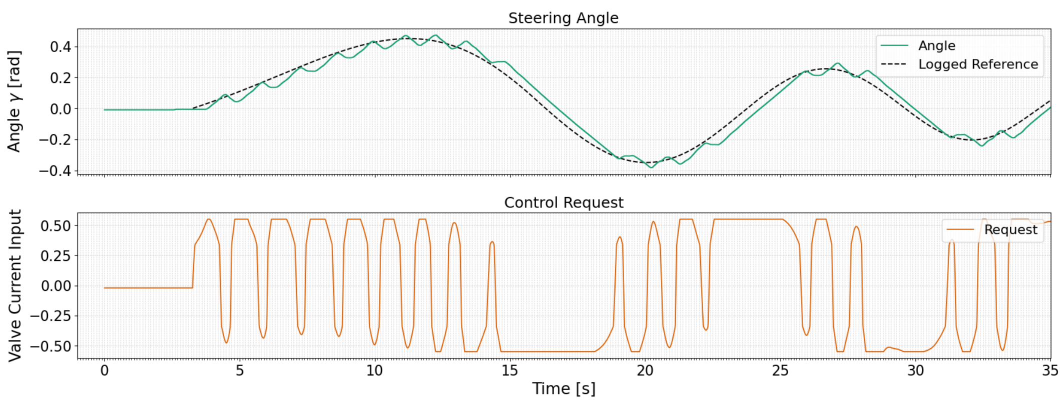

Figure 4.3: Closed-loop sweep on the real vehicle with the rule-based tuning ($\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$, $k_{\mathrm{ESO}} = 5$). The valve current saturates at $550\,\mathrm{mA}$.

Two effects dominate the response. The angle does not follow the reference smoothly: it climbs up to the reference and then drops back down, producing a stuttered steering motion. The valve current swings to its saturation limits in both directions on most reference half-periods. The test is cut short before the full sweep completes, because the pump can no longer supply enough flow under the dropping supply voltage (Section 4.1).

4.5 Diagnosis

The auto-tuned controller did not achieve the design target on the real vehicle. This section investigates why. The step response shows a transport delay that is not in the plant model. Adding this delay to a synthetic plant reproduces the failure, which confirms the delay is the cause. Lowering the bandwidth on the vehicle removes the delay-driven problems but a residual remains. The residual is traced to deadband hysteresis and sensor tilt sensitivity.

4.5.1 A hidden transport delay

A zoom of the step response around the step instant (Figure 4.4) shows that the angle does not move for roughly 0.17 s after the input changes. This is a transport delay. The NLS model (3.16) has no delay term, so the estimator absorbed the delay into the fitted lag $\hat{\tau}$, which is why $\hat{\tau} = 0.288\,\mathrm{s}$ is larger than expected. Splitting $\hat{\tau}$ into delay and true lag gives roughly 0.17 s of delay and 0.118 s of true lag. The true lag is close to the simulated family. The large fitted lag was therefore mostly delay, and the delay is the part the plant model of Section 3.2.3 does not include.

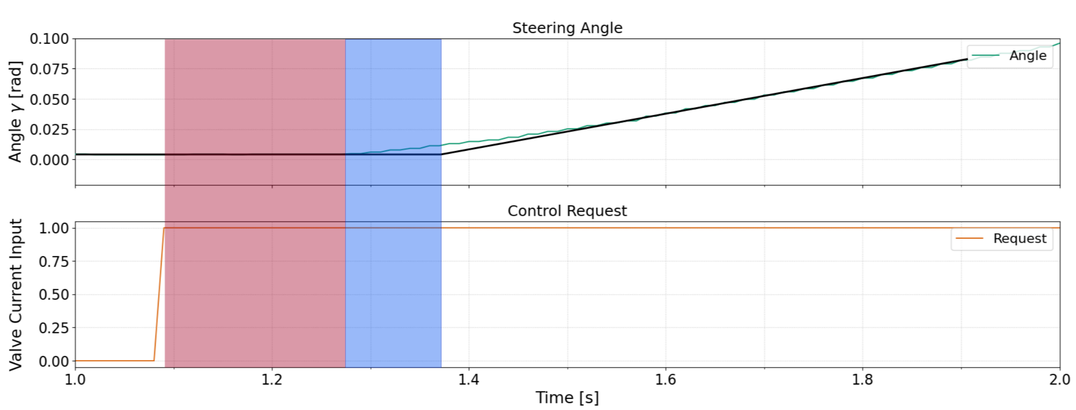

Figure 4.4: Zoom of the real-vehicle step response around the step instant. No angle motion is observed for roughly 0.17 s after the input changes. The red marked area is therefore the deadtime and the blue marked area the actual lag.

The negligible-delay assumption of Section 3.2.3 is violated here. At $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$ the controller acts on an angle measurement that is roughly 0.17 s old. Its correction arrives after the plant has already moved on, and at this bandwidth it overcorrects and drives the valve current to saturation.

4.5.2 Reproducing the failure on a synthetic plant

A synthetic plant is built with the identified gain $\hat{K} = 0.145$, the decomposed lag $\tau_{\text{lag}} = 0.118\,\text{s}$, and an added transport delay of $0.17\,\text{s}$. The delay is the only feature added beyond the synthetic sweeps of Section 3.5. The controller from the real-vehicle test is applied, tuned from the inflated $\hat{\tau} = 0.288\,\text{s}$ to $\omega_{\text{CL}} = 2\,\text{rad/s}$ and $k_{\text{ESO}} = 5$, and the same sine sweep is run.

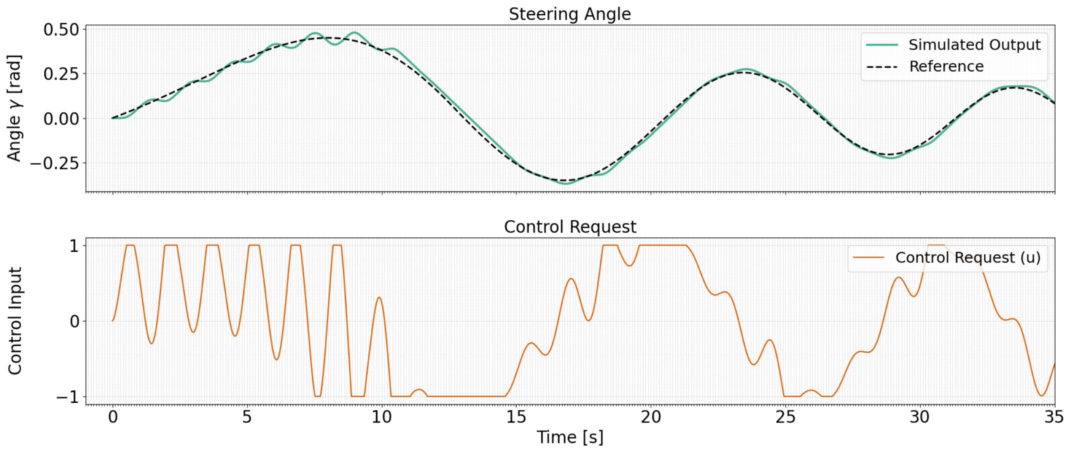

Figure 4.5: Synthetic plant with the identified gain $\hat{K}$, the decomposed lag $\tau_{\text{lag}} = 0.118\,\text{s}$, and an added $0.17\,\text{s}$ transport delay, at the deployed tuning $\omega_{\text{CL}} = 2\,\text{rad/s}$, $k_{\text{ESO}} = 5$. This is a simulation, not a vehicle test.

At the deployed tuning the synthetic plant reproduces the stutter and the saturating control signal seen on the vehicle in Figure 4.3. The transport delay is therefore sufficient, on its own and in simulation, to produce the kind of failure observed on the vehicle.

If the delay is the cause of the failure, lowering the bandwidth should reduce its influence. The bandwidth is therefore lowered to $\omega_{\text{CL}} = 0.3\,\text{rad/s}$ and $k_{\text{ESO}} = 3$.

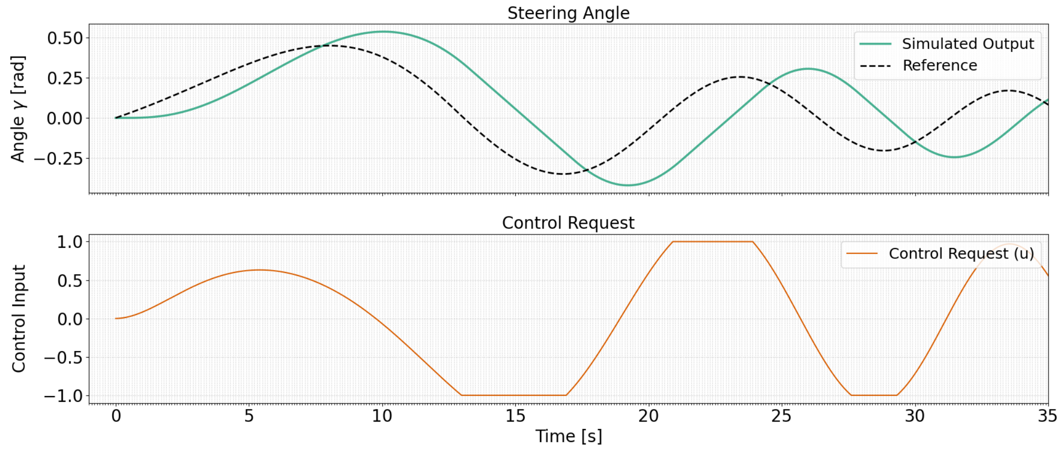

Figure 4.6: The same synthetic plant as in Figure 4.5, at the detuned $\omega_{\text{CL}} = 0.3\,\text{rad/s}$, $k_{\text{ESO}} = 3$. Lowering the bandwidth removes the stutter and the saturation.

At the lower bandwidth the stutter and the saturation disappear and the response settles into a clean track with a tracking lag (Figure 4.6). Once the closed-loop time scale is long compared to the 0.17 s delay, the delay no longer drives the loop into saturation. In simulation, lowering the bandwidth removes the failure.

4.5.3 Detuning the controller on the vehicle

The same reduction is then tried on the vehicle. The tuning is lowered to $\omega_{\text{CL}} = 0.3\,\text{rad/s}$ and $k_{\text{ESO}} = 3$ and the sweep is repeated. The result is shown in Figure 4.7.

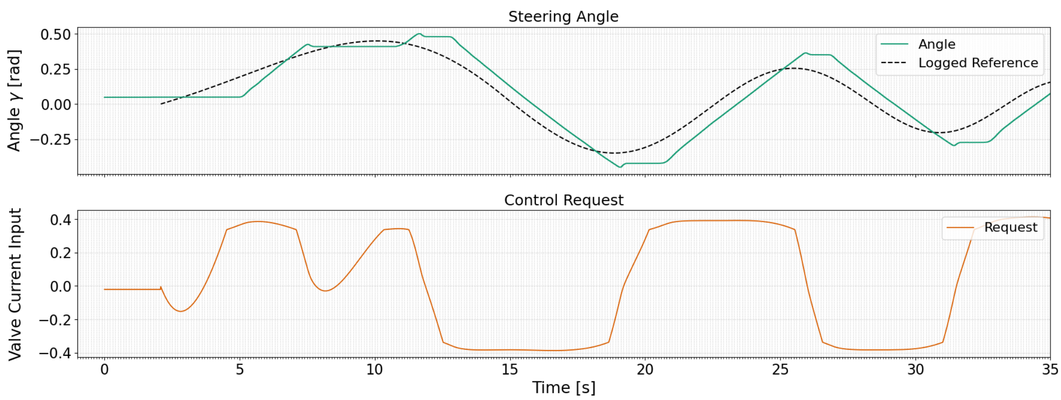

Figure 4.7: Closed-loop sweep on the real vehicle after manual detuning to $\omega_{\text{CL}} = 0.3\,\text{rad/s}$ and $k_{\text{ESO}} = 3$.

The aggressive stutter is gone. The controller no longer drives the angle back across the deadband on each reversal, so the large up-and-down swings of Figure 4.3 disappear and are replaced by a large tracking lag, which is the expected cost of the lower bandwidth. What remains are small peaks at the points where the control signal relaxes toward zero. On the synthetic plant lowering the bandwidth removed the failure entirely. On the vehicle it reduces the control stutter and the tracking error that appears as humps, but does not remove them. An effect is present on the vehicle that the delayed synthetic plant does not contain. The small peaks and this residual are examined next.

4.5.4 Deadband hysteresis and the sensor

Two candidates for the remaining tracking failure are probed directly on the vehicle. A targeted deadband test ramps the current up until the vehicle starts to move, holds, and then ramps the current back down until the vehicle stops. The result is shown in Figure 4.8.

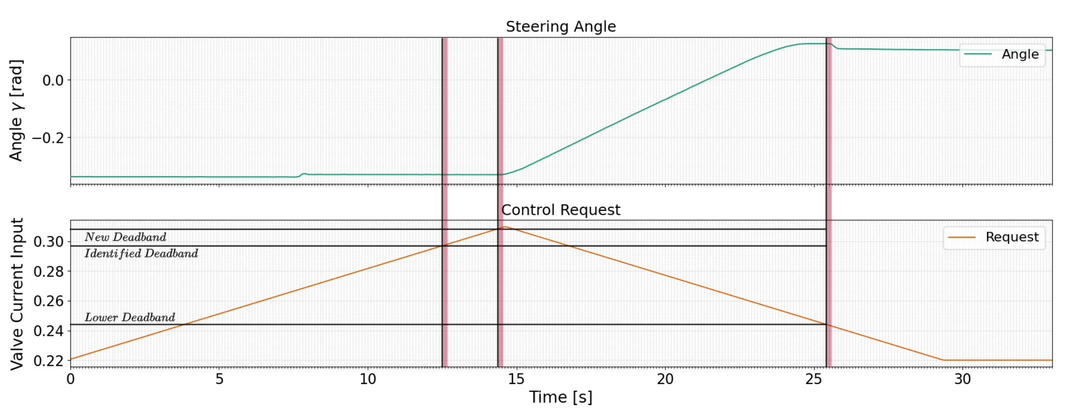

Figure 4.8: Targeted deadband test on the real vehicle, read after accounting for the transport delay (red). On the way up the angle starts to move at a higher current than the identified deadband (new deadband). On the way down it keeps moving until a much lower current, a directional hysteresis. A small angle bump appears when the vehicle stops.

On the way up the angle starts to move at a slightly higher current than the deadband found in Section 4.2, marked as the new deadband in Figure 4.8. The test starts from a non-zero articulation angle, where the linkage lever arm is smaller and the load on the valve is higher. Section 3.2.2 already noted that the recirculation-zone width of the OCT valve grows with load [17], which the plant model simplifies to a constant deadband and assumes the ESO removes. The slightly higher start current fits this load dependence.

On the way down the angle keeps moving until a much lower current, well below both the identified and the new deadband. This drop is much larger than the small rise on the way up. The current that starts the motion is higher than the current where it stops, which is hysteresis. The plant model assumes a single constant deadband (Section 3.2.2) and does not include this hysteresis, so it might be another source of the bumpy tracking left after lowering the bandwidth.

Finally, when the vehicle stops the measured angle bumps back a little. This is the sensitive sensor reacting to a small body tilt as the vehicle settles (Section 4.1), not articulation. The small peaks left in the lowered-bandwidth test of Figure 4.7 have the same shape and are likely the same effect. In the normal-bandwidth test of Figure 4.3 they could not be separated from the angle swinging back under the delay. Lowering the bandwidth removes that swing (Section 4.5.3) and leaves these peaks on their own.

5

Discussion and Conclusion

This chapter reviews what the auto-tuning pipeline achieves, why its deployment on the real vehicle did not match the simulation results, what the simulation evidence can and cannot establish, and what remains for the framework to be useful in practice.

5.1 Achievements and limitations

This thesis set out to replace the manual calibration of the steering controller with an automated pipeline (Section 1.2). The work delivers the groundwork for such a framework rather than a finished tool. Three pieces are in place. The three verification environments (Section 3.3) give a structured way to develop and check each component before it reaches the vehicle. The error-based ADRC and the rule that turns a gain and a lag into its parameters (Chapter 3) give a controller that can be configured from a short standstill experiment. On the simulation model the pipeline meets its target across the nine vehicle configurations and absorbs mass and friction mismatch (Sections 3.6.2 and 3.7).

On the real vehicle the deadband identification and the standstill plant identification run end to end and return expected values, but the closed loop does not track the reference sweep (Chapter 4). The identification part of the pipeline therefore transfers to hardware, while the controller it produces does not yet perform there.

5.2 Why the real-vehicle deployment fell short

The real vehicle is a member of $\mathcal{V}$, but it did not meet the negligible-delay assumption of that set (Section 3.1). The diagnosis of Section 4.5 traced the failure to effects the plant model leaves out. The dominant one is a transport delay.

The pipeline assumes the delay negligible (Section 3.1), so it appears nowhere in the design. It is absent from the plant model, the tuning sweeps, and the identification, which folds it into the fitted lag. On the real vehicle the delay is roughly $0.17\,\mathrm{s}$, and at the bandwidth the rule selects it is large enough to make the valve current saturate (Section 4.5.1).

The transport delay is already documented in the literature on which the plant model is based. Yin et al. [20] identify the same plant class with a delay that exceeds the lag itself and compensate it with a Smith predictor. Despite this, the pipeline is built on the assumption that the delay is negligible. That assumption proved incorrect on the real vehicle.

A further effect not captured by the model is deadband hysteresis (Section 4.5.4), which makes the current needed to start motion higher than the current where it stops. This is a probable candidate for the remaining overshoots in the tracking after lowering the bandwidth reduced the control chatter from the transport delay. The temporary sensor and pump are test setup limitations that were observed in the data but whose exact contribution could not be verified without further tests.

5.3 What the simulation evidence establishes

Every closed-loop result in this thesis, the tuning sweeps of Section 3.5.2, the simulation model verification of Section 3.6.2, and the mismatch study of Section 3.7, shares the same sine-sweep reference, which spans only a fixed frequency band. Behaviour outside that band is not established anywhere in this thesis.

The robustness study of Section 3.7 varies mass and friction, which changes the effective plant gain. The ESO absorbs this change, but that is all the study proves. It shows robustness to a gain shift within the tested range, and nothing beyond that.

The operating point could also differ from the identification point in other ways, such as a shift in τ or changes in the limits of the deadband. The study cannot speak to these. It also does not establish how much gain shift the ESO can absorb. It tests only the chosen perturbation range, and a different operating point could shift the gain further than this.

The tuning rule should be read with the same care. The polynomial $\omega_{\mathrm{CL}}(\tau)$ in (3.15), the order split at $\tau = 0.5\,\mathrm{s}$, and the acceptance ceilings NRMSE ≤ 0.5 and HFR ≤ 0.01 were all fixed at one noise level, one reference sweep, and one visual judgement. The nine configurations confirm the rule works at that setting, but not how it moves when the setting changes. A higher noise level would pull the polynomial down. A different reference band moves the cutoff f_c the HFR is measured against. Tighter or looser ceilings move the operating points the polynomial is fitted through. The coefficients are specific to the calibration in Section 3.5.2 and are not universal constants. The rule is best read as a starting point that gets the controller close enough to skip most manual iteration.

Characterising this sensitivity needs no further vehicle testing. The noise level, the reference profile, and the acceptance ceilings are all inputs to the same synthetic sweep that produced the rule. Re-running it at a few noise levels and reference bands, and re-reading the classification at ceilings around NRMSE $= 0.5$ and HFR $= 0.01$, would map how far the coefficients and the split move. This is left for future work.

5.4 Future work

The following extensions address the issues found in the diagnosis. The most important is to estimate the transport delay from the standstill step, where it is clearly visible, and to extend the tuning rule to use it. The current rule lowers the bandwidth only when τ is large. Estimating the delay alone would not change this, because the rule never looks at the delay. The rule has to be extended to lower the bandwidth when the delay is large as well. This is the model-light way to tolerate it.

Recovering the performance that a lower bandwidth gives up would require compensating the delay rather than tolerating it, which means a predictor and an explicit plant model. Whether this step is worth leaving the model-light approach for is a design decision for the company.

The measured deadband hysteresis should be folded into the deadband compensation, so that the start and stop currents are treated separately. The closed-loop experiment should then be repeated with the production pump and the production sensor. These remove the test-setup limitations and would confirm whether the remaining tracking error is caused by the hysteresis alone.

Finally, the standstill-to-driving operating point should be validated directly on the vehicle, which the real-vehicle tests in this thesis did not cover.

5.5 Conclusion

The groundwork for an automated tuning framework is in place, but it is not yet a finished tool. In simulation the pipeline configures a working controller across a range of vehicles. The real-vehicle deployment showed where the approach reaches its limit. The transport delay, which the model-light pipeline assumes away, proved to be the main obstacle. The missing piece is an extension that detects and mitigates these effects, starting with estimating the delay and tuning to it. With such an extension the framework would be a candidate for the manual-tuning replacement the company is after. As it stands, the work is a method that works in simulation and an analysis of the gap between simulation and real hardware.

Bibliography

[1] R. Amirante, G. Del Vescovo, and A. Lippolis. Flow forces analysis of an open center hydraulic directional control valve sliding spool. *Energy Conversion and Management*, 47(1):114–131, 2006.

[2] Chris Cox, John Tindle, and Kevin Burn. A comparison of software-based approaches to identifying fopdt and sopdt model parameters from process step response data. *Applied Mathematical Modelling*, 40(1):100–114, 2016.

[3] Peter Fritzson, Adrian Pop, Karim Abdelhak, Adeel Ashgar, Bernhard Bachmann, Willi Braun, Daniel Bouskela, Robert Braun, Lena Buffoni, Francesco Casella, Rodrigo Castro, Rüdiger Franke, Dag Fritzson, Mahder Gebremedhin, Andreas Heuermann, Bernt Lie, Alachew Mengist, Lars Mikelsons, Kannan Moudgalya, Lennart Ochel, Arunkumar Palanisamy, Vitalij Ruge, Wladimir Schamai, Martin Sjölund, Bernhard Thiele, John Tinnerholm, and Per Östlund. The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development. *Modeling, Identification and Control*, 41(4):241–295, 2020.

[4] Lulu Gao, Yueqi Dong, and Jixing Zhao. Dynamic modeling and characteristic analysis of articulated steering vehicles. *Applied Sciences*, 13(8), 2023.

[5] Zhiqiang Gao. Scaling and parameterization based controller tuning. In *Proceedings of the American Control Conference*, volume 6, pages 4989– 4996, 07 2003.

[6] Zhiqiang Gao. Active disturbance rejection control: A paradigm shift in feedback control system design. In *Amer. Control Conf.*, page 7 pp., 07 2006.

[7] Jingqing Han. From pid to active disturbance rejection control. *IEEE Transactions on Industrial Electronics*, 56(3):900 – 906, 04 2009.

[8] Kanghui He, Chaoyang Dong, and Qing Wang. Active disturbance rejection adaptive control for uncertain nonlinear systems with unknown time-varying dead-zone input. *Asian Journal of Control*, 24(3):1209–1222, 2022.

[9] Gernot Herbst. A simulative study on active disturbance rejection control (adrc) as a control tool for practitioners. *Electronics*, 2(3):246–279, 2013.

[10] Gernot Herbst, Arne-Jens Hempel, Thomas Göhrt, and Stefan Streif. Half-gain tuning for active disturbance rejection control. *IFAC-PapersOnLine*, 53(2):1319–1324, 2020.

[11] Gernot Herbst and Rafal Madonski. *Active Disturbance Rejection Control: From Principles to Practice*. Springer Nature Switzerland, Cham, Switzerland, 01 2025.

[12] P. M. Lerman. Fitting segmented regression models by grid search. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 29(1):77–84, 03 1980.

[13] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, 2 edition, 2006.

[14] Hongchu Qiu, Qin Zhang, John Reid, and Duqiang Wu. System identification of an electrohydraulic steering system. *SAE Transactions*, 108:361–367, 1999.

[15] Gang Tao and P.V. Kokotović. Adaptive control of plants with unknown output dead-zones. *IFAC Proceedings Volumes*, 26(2, Part 1):73–76, 1993. 12th Triennial World Congress of the International Federation of Automatic control. Volume 1 Theory, Sydney, Australia, 18-23 July.

[16] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[17] Haibo Xie, Wu Zhu, Jiujiu Lv, Liang Hu, and Huayong Yang. Slewing valve based on bypass pressure-compensation principle for dynamic control of large inertia load. *Advances in Mechanical Engineering*, 10(8):1687814018794342, 2018.

[18] Bing Xu, Qi Su, Junhui Zhang, and Zhenyu Lu. A dead-band model and its online detection for the pilot stage of a two-stage directional flow control valve. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 230(4):639–654, 2016.

[19] Fengbo Yang, Hongping Zhou, and Wenxiang Deng. Active disturbance rejection adaptive control for hydraulic lifting systems with valve dead-zone. *Electronics*, 11(11):1788, June 2022.

[20] Chengqiang Yin, Qun Sun, Jian Wu, Chengqiang Liu, and Jie Gao. Development of electrohydraulic steering control system for tractor automatic navigation. *Journal of Electrical and Computer Engineering*, 2018:1–7, 02 2018.

A

Simulation Model Details

This appendix describes the OpenModelica simulation model used to develop and verify the identification, control, and auto-tuning stages of Chapter 3. The model is implemented in Modelica and simulated in OpenModelica [3], using the Modelica Standard Library (MSL) 4.1.0 and the OpenHydraulics library 2.0.0. Its architecture is shown in Figure A.1, and the sections below follow that structure.

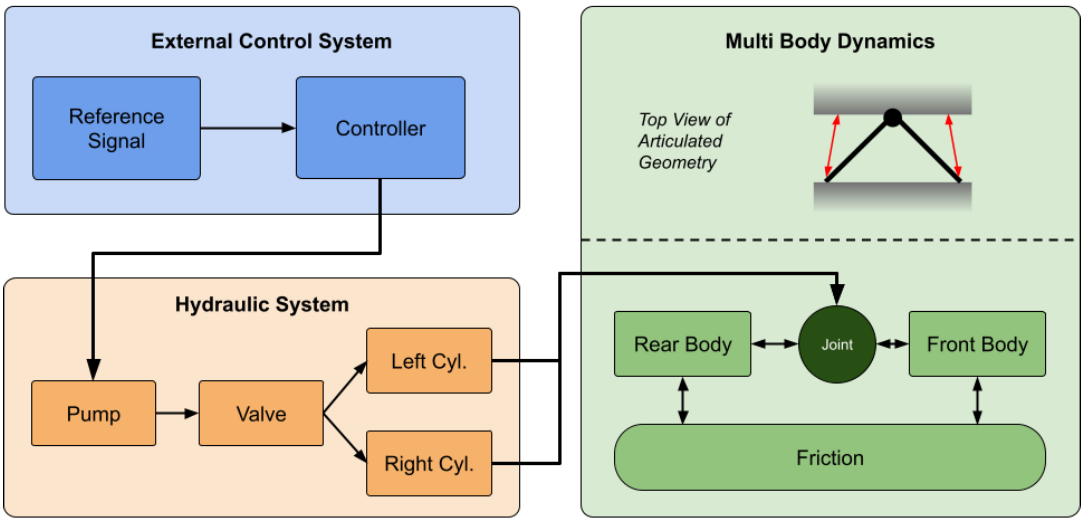

Figure A.1: The OpenModelica simulation model, organised into its mechanical, hydraulic, control, and friction subsystems.

The model is assembled from OpenHydraulics and Modelica Standard library components. Most parameters were set deliberately to represent the machine, but since data sheets were not available for most components, these values are rough estimates rather than numbers taken from specifications. The hydraulic line and fluid model is a further exception. It is included with its built-in library behavior to give the circuit more realistic pressure and flow dynamics, but it is not parameterized to the real vehicle, and its internal pressure-loss and fluid modeling, including the isothermal oil assumption, is left as the library provides it.

A.1 Model Simplifications

Several simplifications of the model are worth flagging for later interpretation.

First, motion is restricted to a horizontal plane. The ground is flat, the vehicle has no pitch or roll, and inclines or uneven terrain are not represented.

Second, the propulsion system is not modelled, and friction is treated as independent of the forward driving speed (not the steering speed) beyond the tanh regularization. As a result, aerodynamic drag, rolling resistance, and other speed-dependent effects are absent. There is also no stiction, so no elevated breakaway force has to be overcome when the vehicle starts from rest.

Third, no separate solenoid deadband is modelled. The spool moves for any nonzero current, so the minimum-current threshold described in Section 3.2.2 is not represented, and the only deadband in the model is the recirculation zone of the OCT valve.

Fourth, the hydraulic supply is assumed to be always on and to deliver a constant flow. The pump activation logic and the time it takes to build up flow are not modelled.

Fifth, no transport delay is modelled in the actuator chain, consistent with the negligible-delay assumption of Section 3.1.

Sixth, sensor effects are limited to additive white noise added when the simulation model is used as a test environment (Section 3.3.2). Sensor drift, bias, quantisation, and communication delays are not modelled. The only non-ideality on the controller path is the 100 Hz sample-and-hold described in Section A.5.

A.2 Validation Instances

To test the methodology in simulation, the model is run on a chosen set of nine parameter combinations. These nine are not the limits of $\mathcal{V}$. They are sample points in the parameter space, picked to test the methodology under a range of conditions. The set has one nominal vehicle, five settings that each change one parameter away from nominal, and three settings that change several at once. The five parameters are mass, cylinder bore diameter, ground friction, valve nominal flow, and linkage geometry. The supply pump is sized to saturate the valve, so its flow is not varied on its own. The cylinder bore is varied instead, since it sets the piston area and with it the articulation rate for a given valve flow. Mass, bore, friction, and valve flow are scaled from the baseline values in Table A.1. Geometry is picked from the three settings in Table A.2, which adjust the body proportions and the articulation gap. The nine vehicles, each with an ID and a short label, are listed in Table A.3. They are the parameter set used by the deadband identification (Section 3.4), the auto-tuning pipeline (Section 3.6), and the model mismatch verification (Section 3.7).

Table A.1: Baseline simulation model parameters. Per-vehicle scalings (Table A.3) are applied multiplicatively to these values.

Parameter	Symbol	Nominal value
Front body mass	m_f	2700 kg
Rear body mass	m_r	1600 kg
Pump flow	Q_p	15 l/min
Longitudinal friction	μ_x	0.15
Lateral friction	μ_y	0.20
Rotational friction	μ_{rot}	0.10
Valve nominal flow	$Q_{v,\mathrm{nom}}$	15 l/min
Cylinder bore diameter	D_c	63 mm

Table A.2: Geometry settings used in the validation set. The front and rear body lengths are measured along the longitudinal axis of each body. The rear articulation gap is the distance between the rear body and the articulation joint.

Setting	Front body length [m]	Rear body length [m]	Rear articulation gap [m]
short-sym	2.0	2.0	0.4
nominal	2.5	2.5	0.5
long-front	3.0	2.5	0.6

Table A.3: The nine simulated vehicles used in this thesis. V0 is the nominal configuration; V1–V5 each shift one parameter axis; V6–V8 shift several axes at once. These are sampled probes into the simulation parameter space, not bounds on the vehicle set $\mathcal{V}$. Numeric entries are multiplicative scalings of the baseline values in Table A.1; geometry is one of three named settings.

ID	Label	Mass	Bore D_c	Friction	Valve flow	Geometry
V0	nominal	1.0	1.0	1.0	1.0	nominal
V1	heavy	1.2	1.0	1.0	1.0	nominal
V2	small-bore	1.0	0.8	1.0	1.0	nominal
V3	asphalt	1.0	1.0	1.2	1.0	nominal
V4	long-front	1.0	1.0	1.0	1.0	long-front
V5	wide-valve	1.0	1.0	1.0	1.2	nominal
V6	lively-corner	0.8	1.0	1.2	1.0	short-sym
V7	worst-corner	1.2	0.8	0.8	0.8	long-front
V8	fast-perm	1.0	0.8	0.8	1.2	short-sym

A.3 Multi-Body Dynamics

A.3.1 Coordinate System and Degrees of Freedom

The model is based on a planar motion assumption, restricting the vehicle dynamics to motion in the horizontal plane. This results in a four degree-of-freedom (4-DOF) system:

- x, y: global position of the rear body center of mass
- ϕ_r: orientation angle of the rear body relative to the global frame
- γ: articulation angle between the front and rear bodies

The planar motion constraint does not allow any vertical translation, roll, and pitch motions, which is a simplification made for speed of development.

A.3.2 Body Geometry and Inertial Properties

Each body is modelled as a homogeneous `BodyBox` component from the Modelica MultiBody library, with the mass uniformly distributed throughout the volume. The moments of inertia are computed automatically from the geometry and the uniform-density assumption. The nominal masses are listed in Table A.1.

A.3.3 Joints and Kinematic Constraints

Two types of joints connect the bodies to the environment and to each other:

1. **Planar Joint (Ground to Rear Body):** A three-degree-of-freedom joint that constrains the rear body to move in the horizontal plane while allowing translation in x and y directions and rotation about the vertical z-axis. This joint is implemented using the `Joints.Planar` component.
2. **Revolute Joint (Articulation):** A single-degree-of-freedom hinge joint connecting the rear and front bodies, allowing relative rotation about the vertical z-axis. The articulation angle γ is defined as the relative rotation angle, with $\gamma = 0$ corresponding to the straight configuration.

A.3.4 Friction Model

Ground–vehicle interaction is modeled through planar friction forces and a yaw torque applied at each body's center of mass. This captures the aggregate effect of track–ground contact without resolving individual track links or contact patches.

Friction Forces and Torque
Friction is modeled as tanh-regularized Coulomb friction, chosen for numerical

stability and low computational cost:

$$F_x = -\mu_x mg\cos(\gamma)\tanh\left(\frac{v_x}{v_0}\right) \tag{A.1}$$

$$F_y = -\mu_y mg\cos(\gamma)\tanh\left(\frac{v_y}{v_0}\right) \tag{A.2}$$

$$T_z = -\mu_{\text{rot}} mgr_{\text{eff}}\cos(\gamma)\tanh\left(\frac{\omega}{w_0}\right) \tag{A.3}$$

where v_0 and w_0 are the translational and rotational tanh-regularisation speeds.

The coefficients differ by direction to reflect track behavior on the ground. Lateral friction exceeds longitudinal friction ($\mu_y > \mu_x$) so the vehicle resists sideways sliding more than forward motion, and μ_{rot} sets the resistance to yaw, with r_{eff} the effective lever arm at which the yaw friction torque acts. The important nominal values are listed in Table A.1.

Implementation
Each body carries a `TrackFrictionSubsystem` that reads v_x, v_y, and ω from `AbsoluteVelocity` and `AbsoluteAngularVelocity` sensors, evaluates the expressions above, and applies the result at the body's center of mass through `WorldForce` and `WorldTorque` actuators.

A.4 Hydraulic System

The articulation joint is actuated by a hydraulic circuit, which is the sole steering input to the vehicle. Because the steering response passes through the chain valve input $\rightarrow$ hydraulic flow $\rightarrow$ piston motion $\rightarrow$ articulation angle, the hydraulics contribute a significant share of the overall response dynamics. The circuit is described below at the level of detail needed to reproduce the model. Components are taken from the OpenHydraulics library.

A.4.1 Hydraulic System Dynamics

The hydraulic system introduces several dynamic effects:

1. **Supply:** The supply is modeled as a pump producing a constant volumetric flow, assumed always on so that the pump activation logic and flow build-up time are not modelled. A relief valve limits the maximum system pressure. The pump is sized to saturate the valve, so the valve sets the effective flow ceiling. The nominal flow is listed in Table A.1.

2. **Valve:** The valve is an open-center tandem proportional directional valve (OCT valve), driven by a control input in the range $[-1, 1]$. Its spool dynamics are modeled as a second-order transfer function parameterized by bandwidth and damping ratio. In the neutral spool position, the supply flow is recirculated to the tank. The nominal flow rating of the valve, which sets the overall flow capacity and is varied along the valve-flow axis of the validation set, is listed in Table A.1.

3. **Cylinder model:** A double-acting differential cylinder with internal leakage across the piston seal. Cylinder friction is omitted, and only numerical damping is included.

4. **Lines and fluid:** Hydraulic lines connecting the supply, valve, and cylinder are modeled using the OpenHydraulics line components, which include laminar and turbulent pressure losses. The working fluid is treated as a compressible oil with pressure-dependent density under an isothermal assumption.

A.4.2 Force Transmission to Vehicle Bodies

The hydraulic cylinders are connected to the vehicle bodies through `LineForceWithMass` components, which convert the hydraulic pressure difference into mechanical line forces acting between attachment points on the front and rear bodies. The cylinder mounting geometry is arranged in a converging V-configuration: the rear attachment points are spaced widely apart while the front attachment points are closer to the vehicle's longitudinal centerline. This produces a moment arm that converts cylinder forces into articulation torque, with an effective lever arm that varies with the articulation angle. The normalized geometric leverage for one side of the simulation instance is shown in Figure A.2.

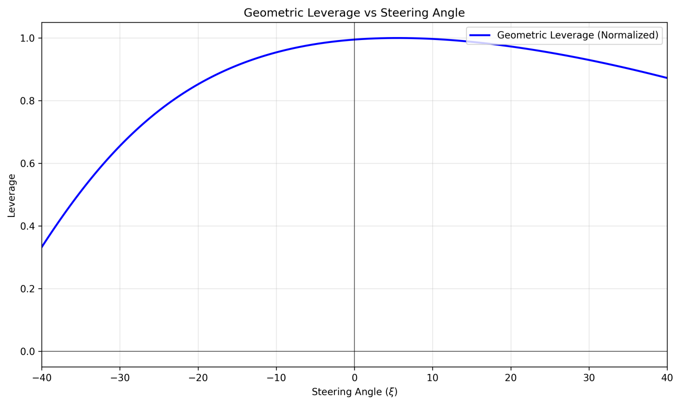

Figure A.2: Normalized geometric leverage for one piston of the steering linkage as a function of articulation angle γ. The leverage is maximal at $\gamma = 0$ and decreases toward the mechanical limits.

A.5 External Control System

The controller receives a single feedback signal: the articulation angle γ between the front and rear bodies, measured by a `RelativeAngles` sensor at the articulation joint. It is implemented in external C code and integrated into the Modelica model through external function interfaces, so that the same code runs in simulation and on the embedded target while preserving its internal state between calls. The controller operates at a fixed sample time of $T_s = 10\,\text{ms}$ (100 Hz): sensor signals are sampled using Modelica's `when sample(0, Ts)` construct, and the control output is held constant between sample instants (zero-order hold).

B

Deadband Identification Algorithm

This appendix gives the details of the breakpoint grid search used to identify the deadband half-width in Section 3.4.2.2: the exact ramp response that fixes the segment shapes, the least-squares objective the grid search minimises, and the two alternative variants compared in the benchmark of Section 3.4.2.3.

B.1 Exact ramp response

In the active region the plant is driven by the input past the deadband, $u_{\text{valve}}-d$, which ramps up over time. Because the Type-1 second-order plant $G(s) = K/[s(\tau s+1)]$ (Section 3.2.3) integrates its input, the input ramp drives the angle along a quadratic. Writing $t = u_{\text{valve}}/\alpha$, after the deadband edge $t_d = d/\alpha$ the response is

$$\gamma(t) = \frac{K\alpha}{2}(t-t_d)^2 - K\alpha\tau(t-t_d) + K\alpha\tau^2\Big(1 - e^{-(t-t_d)/\tau}\Big). \tag{B.1}$$

The first two terms are a quadratic in $t - t_d$, that is, in the input past the deadband $u_{\text{valve}} - d$. The last term is a transient left by the lag τ. It decays within a few τ, and because the ramp is slow the input barely advances over that time, so almost all of the data lies on the clean quadratic. The measured $(u_{\text{valve}}, \gamma)$ curve therefore has the shape the fit of (3.6) uses, a constant in the dead region and a degree-2 polynomial in the input past the deadband in the active region.

B.2 Grid-search objective

Let $\{(u_{\text{valve},i}, \gamma_i)\}_{i=1}^N$ be the recorded samples, indexed in order of rising input along the ramp. A candidate breakpoint at index k splits them into a left window, the samples $i \le k$ in the dead region where the angle is still flat, and a right window, the samples $i > k$ in the active region where the angle rises. The candidate deadband is the input at the split, $d_k = u_{\text{valve},k}$, and the right window is measured from it through $s_i = u_{\text{valve},i} - d_k$. The piecewise model fitted at each candidate is (3.6), with the two segments fitted independently. The left window gives the constant as its sample mean $c^*(k) = \frac{1}{k}\sum_{i\le k}\gamma_i$, and the right window gives a_0, a_1, a_2 by ordinary least squares on the quadratic in s_i. The root-mean-square error across both segments is

$$\mathrm{RMSE}(k) = \sqrt{\frac{1}{N}\left[\sum_{i\le k}\Big(\gamma_i - c^*(k)\Big)^2 + \sum_{i>k}\Big(\gamma_i - a_0^*(k) - a_1^*(k)s_i - a_2^*(k)s_i^2\Big)^2\right]}, \quad \text{(B.2)}$$

and the breakpoint estimate is the candidate that minimises (B.2),

$$\hat{k} = \arg\min_k \ \mathrm{RMSE}(k), \qquad \hat{d} = u_{\mathrm{valve},\hat{k}}. \quad \text{(B.3)}$$

In A-pos the position does not jump at the breakpoint, so the left constant is tied to the right intercept, $c = a_0$, in each iteration.

B.3 Alternative variants

Two variants are compared against A-pos in the benchmark of Section 3.4.2.3.

A-vel (velocity fit). A-vel applies the same fit to a velocity estimate $\dot{\gamma}(u_{\mathrm{valve}})$, where the active-region segment drops from degree-2 to degree-1. The velocity is obtained by differentiating the angle, which amplifies the noise and so has to be low-pass filtered.

Threshold detector. To cross-check the model-fit variants against a method that uses a completely different strategy, a simple threshold detector is included. The first N_{base} samples of the ramp lie inside the dead region, where the angle is not moving. Their mean μ_0 and standard deviation σ_0 define a noise band. The estimate is the first u_{valve} at which the angle leaves this band,

$$\hat{d} = \min\Big\{ u_{\mathrm{valve}}(t) \ \Big| \ |\gamma(t) - \mu_0| > k_\sigma \cdot \sigma_0 \Big\}, \quad \text{(B.4)}$$

where k_σ is the threshold multiplier on σ_0. The detector is intentionally simple and noise-dependent. As the measurement noise grows, σ_0 grows with it, and the band widens, delaying the trigger. Its role is not to compete with the model-fit variants but to give an independent estimate of the same quantity.

C

Controller-Tuning Sweep Details

This appendix supports the parameter sweeps of Section 3.5.2.3. It records how the two acceptance thresholds were anchored from the tracking responses and shows the per-plant responses behind the rule.

C.1 NRMSE threshold

In the noise-free baseline, the reference tracking at $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$ and $k_{\mathrm{ESO}} = 3$ (Figure C.1) shows the oversized-b_0 signature directly. On the fast plants the output barely moves and the control signal is almost flat, the controller no longer pushing the plant hard enough because the control law divides by the inflated b_0 (Section 3.5.2.3, (3.13)). The slower plants in the same figure are tracked at a visually acceptable level, with NRMSE around 0.5 or below. This sets the working tracking threshold NRMSE ≤ 0.5 used in Section 3.5.2.3.

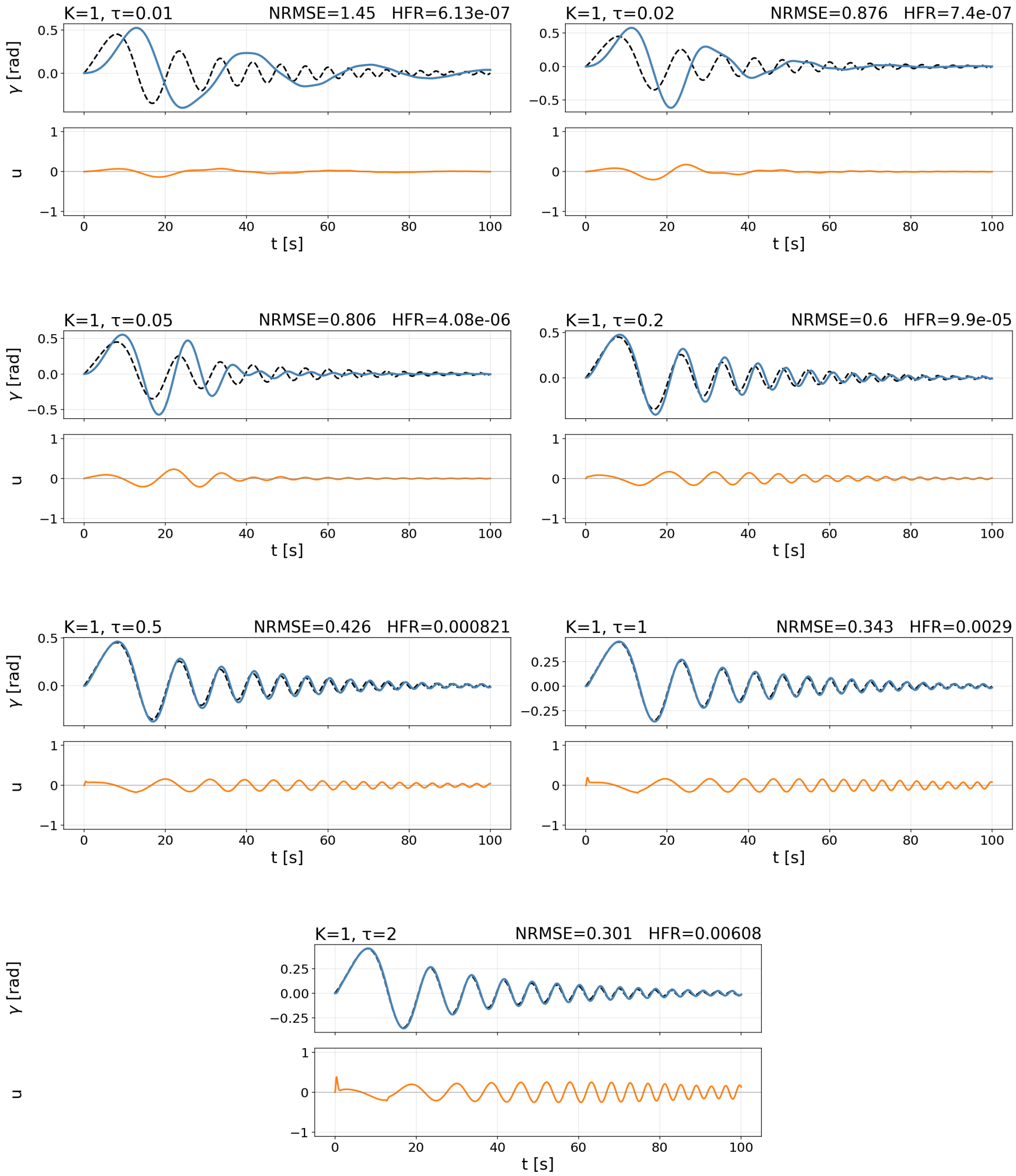

Figure C.1: Noise-free tracking of the 2nd-order ADRC at $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$ and $k_{\mathrm{ESO}} = 3$ on the fast plants, showing the near-flat response that is the signature of an oversized b_0.

C.2 HFR threshold

With measurement noise added, the control signals of the 2nd-order sweep (Figure C.2) set the smoothness threshold. Chatter stays bounded across all plants but grows with τ. The plant with $\tau = 0.8\,\mathrm{s}$ is taken as the upper acceptable limit: there the HFR reaches slightly above 0.01, the loop stays stable, and the control signal is still acceptable. The threshold is therefore set to HFR ≤ 0.01.

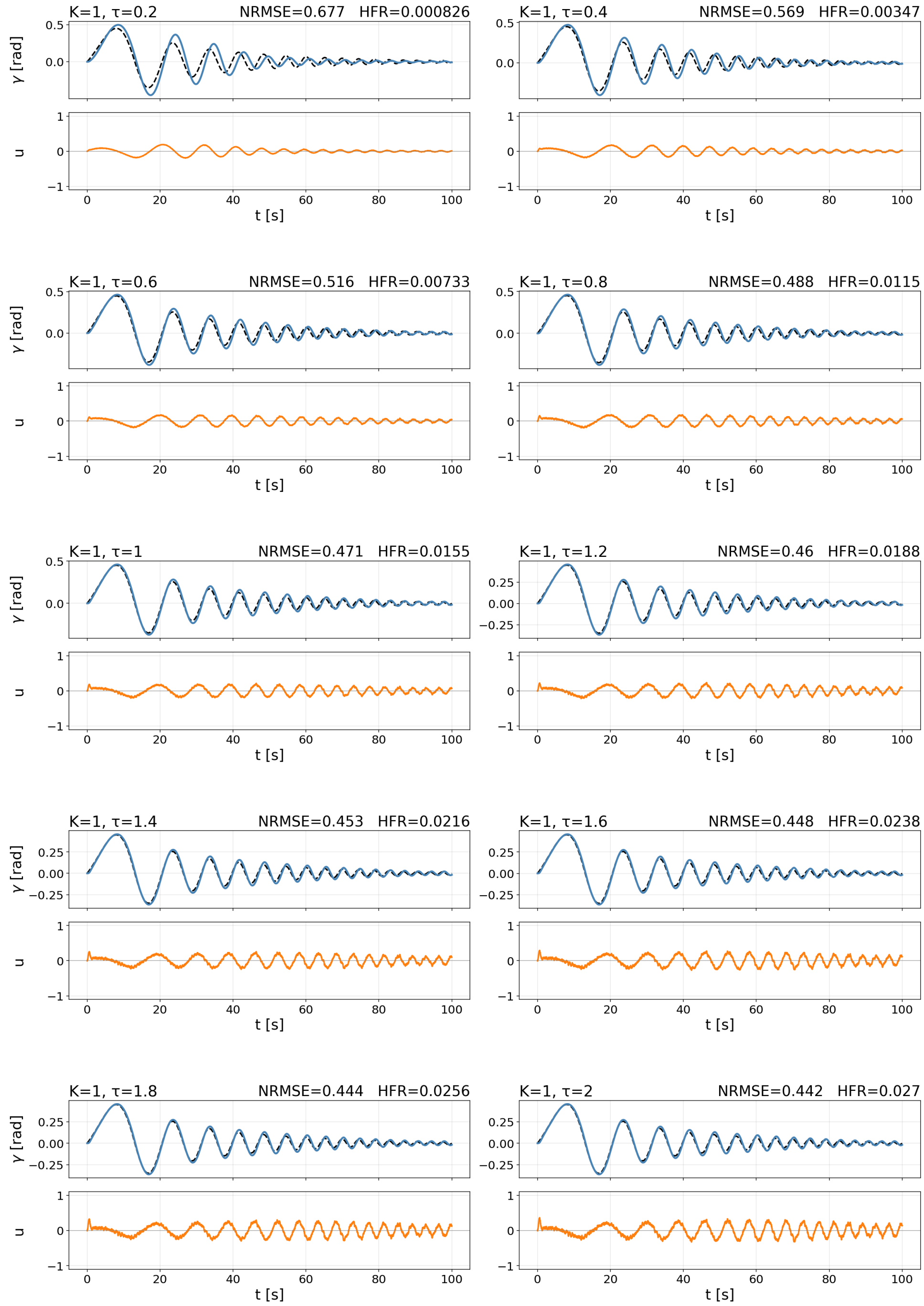

Figure C.2: Closed-loop response of the 2nd-order ADRC at $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$ and $k_{\mathrm{ESO}} = 3$ with measurement noise and half-gain tuning, for $\tau \in [0.2, 2.0]\,\mathrm{s}$. The chatter on the control signal grows with τ.

Zooming the 2nd-order sweep around this region (Figure C.3) shows that for $\tau > 0.8\,\mathrm{s}$ the NRMSE crosses sit above the HFR $= 0.01$ line, so these slower plants cannot reach NRMSE $= 0.5$ while holding HFR ≤ 0.01. Noise sensitivity is prioritised, so the NRMSE is sacrificed on them and the operating point is read off the HFR $= 0.01$ circles, which is the source of the polynomial rule (3.15).

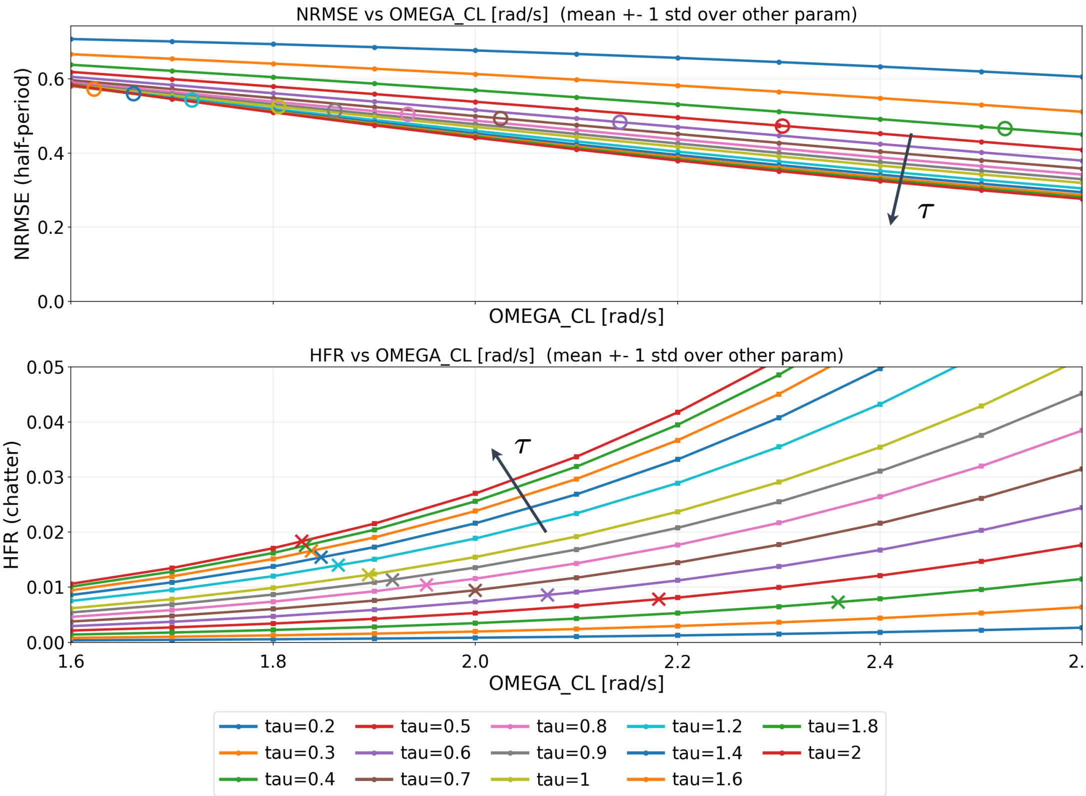

Figure C.3: Zoomed linear-scale version of Figure 3.8 around the region where the NRMSE crosses and the HFR circles sit, with k_{ESO} fixed at 3.

C.3 First-order tracking

The 1st-order tracking at $\omega_{\mathrm{CL}} = 2\,\mathrm{rad/s}$ and $k_{\mathrm{ESO}} = 5$ (Figure C.4) confirms the fast-plant rule. The reference is followed accurately over the full fast range and the control signal stays calm. From $\tau = 0.45\,\mathrm{s}$ upward the control signal degrades, and at $\tau = 0.6\,\mathrm{s}$ it shows harsh chatter without saturation, which is consistent with placing the order split at $\tau = 0.5\,\mathrm{s}$.

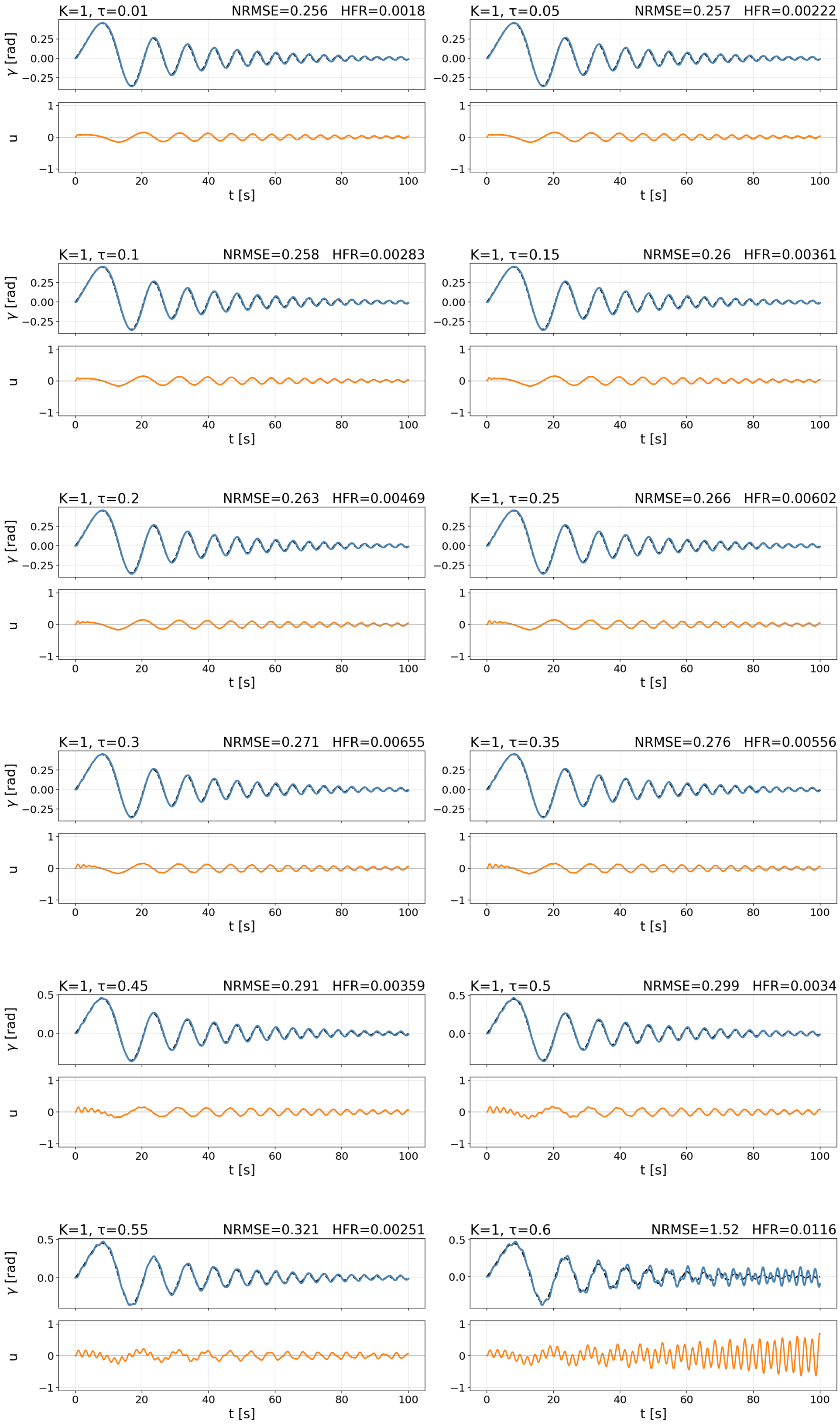

Figure C.4: Closed-loop response of the 1st-order ADRC at $\omega_{\text{CL}} = 2\,\text{rad/s}$ and $k_{\text{ESO}} = 5$ with measurement noise and half-gain tuning, on the fast plants.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY