CHALMERS
UNIVERSITY OF TECHNOLOGY

Identification-based Control of Engine Oil Pressure

Master's thesis in Systems, Control and Mechatronics

JULIUS GRESHAKE

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Master's thesis 2021

Identification-based Control of Engine Oil Pressure

JULIUS GRESHAKE

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2021

Identification-based Control of Engine Oil Pressure
JULIUS GRESHAKE

Supervisor: Milad Mousavi (Volvo Group)
Examiner: Paolo Falcone

Master's Thesis 2021
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Identification-based Control of Engine Oil Pressure
JULIUS GRESHAKE
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Current solutions for controlling engine oil pressure rely on model-free controllers that are tuned following tuning heuristics. Due to the complex structure of modern engine oil systems, the tuning process is often time-consuming and difficult. This thesis investigates if optimal control strategies based on black-box system identification of the engine oil system achieve desirable performance and how such controllers compare to currently used model-free control approaches. For this purpose, estimation data is first collected on a test rig that allows isolated experiments on the engine oil system. Using the collected data, linear state-space and polynomial models with varying complexity are identified and evaluated with regards to accuracy and general validity across the operation range of the system. Several linear-quadratic integral controllers are designed after selecting a suitable underlying model. Their ability to follow reference steps and reject disturbances is tested at varying operating conditions of the test rig. The results of these tests are compared to those achieved by a proportional-integral (PI) controller serving as a comparison baseline.

Linear models were able to reproduce the behaviour of the real system with a maximum accuracy of 65%. The two-state state-space (SS2) model was chosen for the subsequent step of model-based controller design due to its low complexity and marginal accuracy reduction compared to the other identified models. Two of the developed controller designs resulted in consistently improved reference tracking and disturbance rejection compared to the PI controller. For the system under consideration, tuning the model-based controllers was found to be more intuitive than the tuning process for the PI controller.

Keywords: system identification, model-based control, optimal control

Contents

1. Introduction

The current state of oil pressure control and the problem statement for this thesis are presented in the first section of this chapter. Next, the limitations of the thesis work are outlined and a quick note on the topic of confidentiality is given. The final section explains the overall structure of the thesis.

1.1. Problem Description

The engine oil system is one of the most important parts of a heavy-duty engine and plays a key role in lubricating moving parts, cooling down hot components, absorbing sound, limiting mechanical energy consumption, cleaning the inner side of the engine and protecting components against corrosion. The actuators of the engine oil system should be controlled in a way that guarantees the engine to be always supplied with appropriate amounts of oil. The oil pressure demand is determined through measurements of the engine's rotational speed, produced torques and possibly other factors which are fed into lookup tables usually provided by original equipment manufacturers (OEMs). Failure to meet the oil pressure demand can result in durability issues and harm the engine [1]. By only providing as much oil to the engine as requested at any given moment, energy loss can be significantly reduced and fuel consumption can be lowered [2]. Thus, the goals of this thesis are in line with "Sustainable Development Goal 12" of the United Nations: responsible consumption and production. Since heavy-duty engines are used on a large and global scale, even small reductions in fuel consumption can have a big impact.

Adequate controllers must attain stability, provide suitable set-point tracking and offer good disturbance rejection over a wide range of operating conditions; PI- and PID-control is currently the industry standard for this task. These model-free controllers deliver acceptable performance and allow heuristic tuning without requiring explicit knowledge of the system's internal dynamics. For improved reference tracking and disturbance rejection or an increased robustness against changing operating conditions, advanced methods from scientific fields like optimal control or robust control are available. These are model-based control approaches that require a mathematical description of the system dynamics to derive the control laws that decide on how to operate the system's actuators.

The required models can be derived from so-called "first principles" such as Newton's axioms, but the high degree of complexity of the engine oil system complicates this approach. This thesis investigates how well the system dynamics can be reproduced by models obtained through a black-box system identification process, which algorithmically derives the system dynamics from measurements of the system's inputs and outputs.

Additionally, the performance of model-based controllers based on such estimated models is compared to that of model-free controllers. The designed controllers are evaluated with regards to their reference tracking and disturbance rejection abilities.

1.2. Limitations

There are other system identification strategies such as white-box and grey-box modelling, which allow model identification based on partial knowledge of the true system dynamics, however this thesis will only investigate the improvements in control performance based on black-box-modelled system dynamics.

Additionally, it is possible to identify nonlinear models with the above strategies, which can be used for nonlinear control approaches such as exact feedback linearization or backstepping. Neither nonlinear system identification nor nonlinear control will be covered in this thesis. Instead, linear approximations of the system dynamics will be estimated from the system's input output measurements and will subsequently be used for linear control strategies.

As a final limitation, the system's basic configuration (i.e. the amount, location and type of sensors and actuators) is considered to be immutable. Introducing additional actuators to the system or changing the arrangement of those currently in use could lead to better overall performance, but such changes to the engine oil system are beyond the scope of this thesis.

1.3. Confidentiality

For reasons of confidentiality, this thesis will not describe the structure and components of the engine oil system in detail. In addition, output values will be rescaled and no model equations or controller gain matrices will be disclosed to avoid sharing internals of the Volvo Group.

1.4. Thesis Structure

After this introductory chapter, background information about the considered system as well as the applied scientific theory is provided in Chapter 2. The methods used for the system identification process and the obtained results are presented in the third chapter. Building on these results, Chapter 4 outlines the control design process and Chapter 5 shows the evaluation of the designed controllers. In Chapter 6, the overall results and potential further research directions are discussed.

2. Background Information

This chapter begins with two sections containing basic information about the oil engine system and the component rig used for obtaining experimental data. These sections are followed by an overview of the scientific concepts that are required for achieving the goals laid out in the introduction.

2.1. Engine Oil System

From a control perspective, the engine oil system considered in this thesis has a multiple input multiple output (MIMO) structure. There are two locations in the engine with independent oil pressure requirements. These pressure requirements are updated continuously based on measurements external to the system such as engine speed and motor torque. At both target locations, the current oil pressure is monitored through pressure sensors. Their readings constitute the two outputs of the control system and will be referred to as y_1 and y_2 throughout this thesis.

The oil pressure at the two target locations can be influenced by two actuators. They are controlled by the input signals u_1 and u_2, which are generated by an electronic control unit (ECU) and range from 0% to 100%. The ECU calculates these input values based on the sensor readings from y_1 and y_2 and the control strategy that is used. Because the control unit for the engine oil system is a digital device operating at a fixed frequency, discrete system descriptions and control strategies will be used in this thesis.

An abstraction of the system layout is shown in Figure 2.1. Due to the sequential arrangement of the two actuators, the MIMO system can not be simplified into two independent single input single output (SISO) systems, a technique that is used where possible to reduce model complexity and facilitate the design of controllers.

2.2. Component Rig

The experimental data used throughout this thesis is obtained from a component rig, which allows the isolated testing of the engine oil pump.

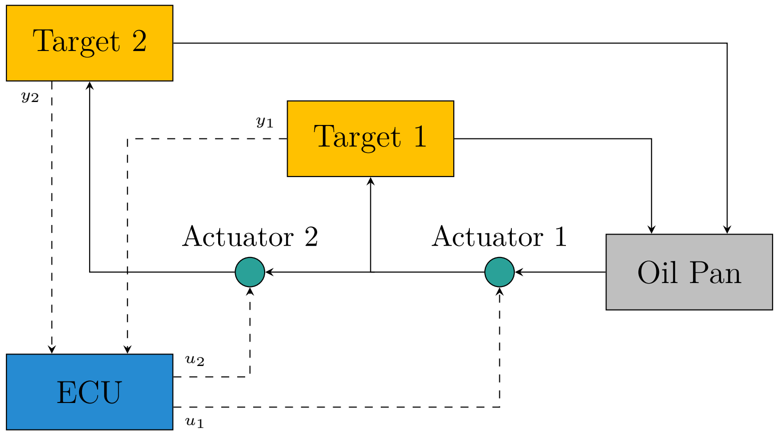


Figure 2.1.: Schematic of the engine oil system

Component rigs provide several advantages over testing in a fully assembled vehicle:

- Lower operational costs
- Less expertise needed for operation
- Less risk for (and damage in case of) critical operational errors
- Higher availability

The main drawback of this testing method is the difference in system behaviour compared to a fully assembled vehicle. This means that results obtained on a component rig are not directly transferrable to a real vehicle. Additionally, some of the system properties and behaviours may depend on the interactions of the engine oil system with other vehicle components, which are not included in the component rig.

Despite these limitations, the use of a component rig is an important first step when exploring new control strategies as it allows to quickly generate results and rule out unpromising approaches.

2.3. System Identification

System identification is the process of deriving dynamical models from observed input and output data [3], as opposed to manually creating these models by mathematically describing the physical phenomena that govern the system behaviour [4]. Due to the large scope of this scientific field, this section will be limited to explanations regarding the estimation of linear models from measured time-domain data. Additional information on nonlinear system identification and identification from frequency-domain data can be found in [5, 6].

MATLAB's System Identification Toolbox will be used for this thesis, which contains implementations for a wide variety of system identification algorithms.

2.3.1. Data Collection

There are multiple works concerned with identifying optimal input patterns for a system so that a minimal amount of input output data can be used for system identification [7, 8]. This is especially relevant for industrial applications, where the system identification process would require interrupting production. If there are no such concerns (which is the case here, since tests can be conducted on the component rig), a pseudo-random binary sequence (PRBS) input signal as proposed in [9] can be used. Using this type of signal as the system input ensures that the system dynamics are revealed in the produced output measurements and can then be identified [10].

2.3.2. Model Selection

Before running the system identification algorithms, the general structure of the model that will be estimated has to be chosen. Depending on the model structure, additional options such as the model order can be set.

State-space Models

The state-space representation is a compact mathematical description of a system's dynamic properties through first-order differential or difference equations [11]. In discrete time, these equations are as follows:

$$\begin{aligned} \mathbf{x}[n+1] &= \mathbf{A}\mathbf{x}[n] + \mathbf{B}\mathbf{u}[n] \\ \mathbf{y}[n+1] &= \mathbf{C}\mathbf{x}[n] + \mathbf{D}\mathbf{u}[n] \end{aligned} \tag{2.1}$$

The number of system inputs and outputs determines the size of the $\mathbf{u}$ and $\mathbf{y}$ vectors respectively. Each entry of the $\mathbf{x}$ vector represents a system state and the size of this vector is referred to as the *order* of the state-space model. For the system identification process, the state-space order can be chosen freely and is usually determined experimentally.

The form of the state-space matrices can be influenced through an additional function parameter in MATLAB. Choosing a canonical observer form causes $\mathbf{C}$ to be an identity matrix and $\mathbf{D}$ to contain only zeros [10]. In this configuration the system states are equivalent to the system outputs, which is particularly useful for state feedback design, where the measured system outputs can be used directly without the added step of state estimation through an observer structure or a Kalman Filter.

Polynomial Models

Polynomial models are transfer function representations of systems that have the following general structure [12]:

$$\sum_{j=1}^{n_y} A_{lj}(q)y_j(t) = \sum_{i=1}^{n_u} \frac{B_{li}(q)}{F_{li}(q)} u_i\left(t - nk_i\right) + \frac{C_l(q)}{D_l(q)} e_l(t) \tag{2.2}$$

In this equation, n_u and n_y are the numbers of system inputs and outputs, while $e(t)$ represents a white-noise term. The ith system input $u_i(t)$ is affected by the transport delay nk_i. To account for multiple system outputs, the subscript l denotes the lth output equation.

A_{lj}, B_{li}, C_l, D_l and F_{li} contain the polynomial orders for the combination of input and output denoted by their respective subscripts. Each polynomial is a function of q, a time-shift operator equivalent to the operator z found in the Z-transform. Assuming that A is of degree n_a for $l = n_y = 1$, we get:

$$A(q) \quad = 1 \quad + a_1 q^{(-1)} \quad + \cdots + a_{n_a} q^{(-n_a)} \tag{2.3}$$

$$A(q)y(t) = y(t) + a_1 y(t-1) + \cdots + a_{n_a} y(t - n_a) \tag{2.4}$$

The other polynomials follow the same structure (with the notable exception of B, which omits the leading 1).

By using specific combinations of polynomials, we obtain the model structures autoregressive with exogenous input (ARX), autoregressive moving average with exogenous input (ARMAX) and Box-Jenkins (BJ):

- ARX: Polynomials A and B
- ARMAX: Polynomials A, B and C
- BJ: Polynomials B, C, D and F

Black-box and grey-box estimation

With grey-box estimation, partial knowledge about the model structure can be used to pre-define some of the model parameters such as specific entries of the state-space matrices or known poles of the polynomial transfer functions [13]. Black-box estimation on the other hand implies no previous knowledge about the system, meaning that all model parameters are determined algorithmically. Only black-box estimation is used in this thesis.

2.3.3. Identification Process

While there are specific algorithms for the identification of each model structure, the general procedure of the system identification process remains the same: the difference between the measured system outputs and those generated by the estimated model is iteratively minimized [14]. In MATLAB, this difference is determined by the following general quadratic cost function, parameterized by the parameter vector θ:

$$V(\theta) = \frac{1}{N} \sum_{t=1}^{N} e^T(t, \theta) W(\theta) e(t, \theta) \tag{2.5}$$

In this equation, N represents the number of recorded datapoints used for the system identification process, $e(t, \theta)$ is the error vector (a row-vector with ny rows) and $W(\theta)$ is a weighting matrix that can be used to adjust the weighting of the error terms for each output.

2.3.4. Model Fit

For the identified model, a fit percentage is calculated according to the following equation [15]:

$$\text{Fit Percentage} = 100 \left(1 - \frac{\| y_{\text{measured}} - y_{\text{model}} \|}{\| y_{\text{measured}} - \text{mean}\left(y_{\text{measured}} \right) \|} \right) \tag{2.6}$$

With a fit percentage of 100%, the identified model recreates the measured output signal perfectly when provided with the recorded input data. A fit percentage of 0% is equivalent to a model that just takes the mean value of the measured data.

After the model estimation process is finished, MATLAB reports the final fit percentage achieved on the provided dataset. Additionally, the fit percentage of an estimated model on a separate validation dataset can be calculated, which is an important step in order to identify and avoid overfitting. Overfitting is a phenomenon where an identified model is able to achieve a good fit on the data used during the estimation process but fails to recreate the measured outputs of other datasets that were not used during the estimation process [16]. It signals a lack of general validity of the model and is an indicator that the chosen model order (and thus the number of parameters of the underlying model) is too high.

2.4. Linear-quadratic Control

Linear-quadratic control is a control strategy from the field of optimal control that aims to minimize the quadratic cost functional J, which (in discrete time) is defined as follows [17]:

$$J(u) = \sum_{n=1}^{\infty} \left(x[n]^T Q x[n] + u[n]^T R u[n] + 2x[n]^T N u[n] \right) \tag{2.7}$$

In J, the matrices Q, R and N can be chosen freely to weigh the impact of the system states, the system inputs and their cross-relation respectively. The discrete-time Algebraic Riccati Equation (ARE) associated with the above cost function is:

$$A^T PA - P - \left(A^T PB + N\right)\left(B^T PB + R\right)^{-1}\left(B^T PA + N^T\right) + Q = 0 \tag{2.8}$$

After solving this equation for P, the optimal state feedback law in the form of $u[n] = -Kx[n]$ that minimizes the cost function Equation (2.7) can be calculated:

$$K = \left(B^T PB + R\right)^{-1}\left(B^T PA + N^T\right) \tag{2.9}$$

A controller that implements this feedback law is called linear-quadratic regulator (LQR). Figure 2.2 shows an LQR in the block notation common for describing control systems.

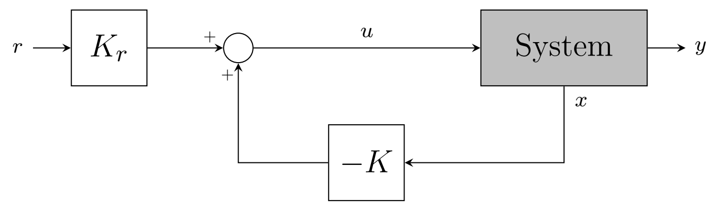


Figure 2.2.: LQR Structure

To ensure reference tracking for non-zero set points, the reference signal r is pre-multiplied with the matrix K_r, which is the inverse of the closed-loop transfer matrix H_c [17]:

$$K_r = H_c^{-1} = \left(D + C\left(I - (A - BK)\right)^{-1} B\right)^{-1} \tag{2.10}$$

2.4.1. LQI Control

When an LQR is used to control a system with slightly different dynamics than those used to determine the LQR's gain matrices K and K_r, undesired behaviour of the controlled system in the form of steady-state errors can be observed. To avoid this, the LQR structure can be extended by including the integral of the tracking error e [18].

The resulting linear-quadratic integral (LQI) controller eliminates steady-state errors in the case of disturbances or modelling inaccuracies. The LQI controller structure is shown in Figure 2.3.

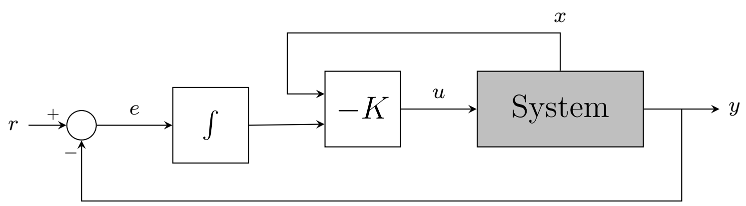


Figure 2.3.: LQI Structure

2.4.2. Tuning Procedure

Bryson's rule (named after its creator Arthur Bryson [19]) can be used to choose the initial values of the Q- and R-matrices of the quadratic cost function J (with N usually set to 0). If v_i is the largest expected value for state x_i, then the diagonal element Q_{ii} is set to v_i^{-2}. All off-diagonal matrix entries are set to 0. A similar procedure can be applied for the entries of the R matrix, where v_i represents the biggest allowed control signal for input u_i. This strategy limits the summands of the cost function J to a theoretical maximum of 1, ensuring an equal influence of every state and every input on the total cost.

Further adjustments to the matrix entries can be made by understanding them as indicators for the penalization weights on the controlled states (for Q) and controller inputs (for R). If reactions to reference changes or accumulated integral errors are perceived as too slow, the corresponding entries of the Q-matrix can be increased. Similarly, if the generated control values cause oscillations due to noise at steady state or because the output signal limits are reached too frequently, the entries of the R-matrix can be increased.

2.5. Control Performance Metrics

There are many methods for evaluating the performance of a controlled system. Its time behaviour can be formally analysed through metrics related to its step response. In addition, the output error integral can be calculated, which is a more general measure of a controller's ability to minimize the error between reference signals and system outputs.

2.5.1. Step Response

The step response describes the system behaviour after applying a Heaviside step function to its reference signal [20]. A typical step response is shown in Figure 2.4. The recorded output signals can be analysed to extract (among others) the following metrics:

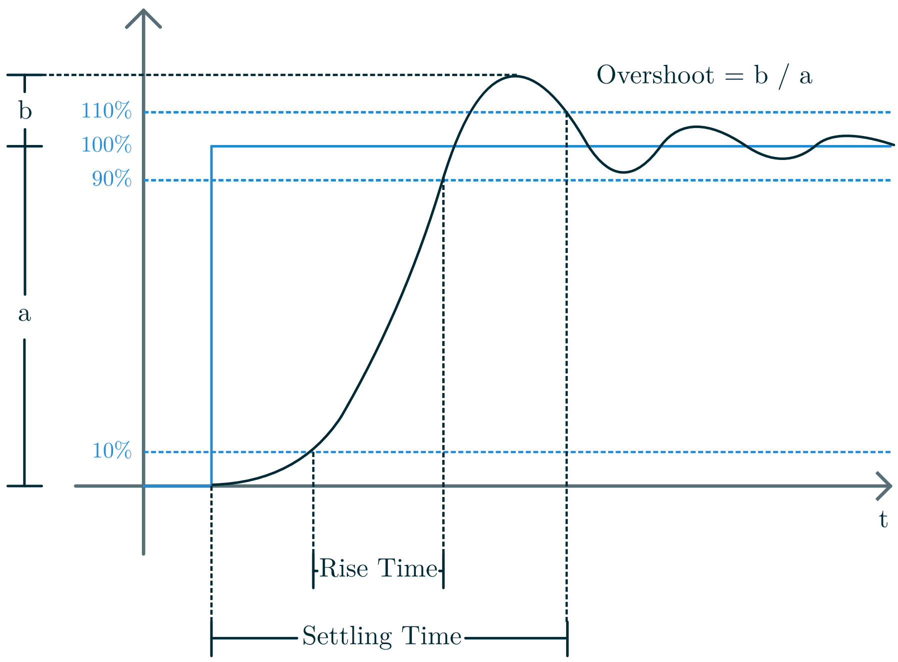


Figure 2.4.: A typical step response with rise time, settling time and overshoot highlighted

Rise Time

The rise time is defined as the time it takes for the signal to rise from a lower ratio of the step size (usually 10% to 30%) to a higher ratio of the step size (usually 90% or 95%) and is thus a general performance metric of a controlled system.

Settling Time

The settling time is a second performance metric defined as the time it takes the output signal to settle within an error band (usually 5% to 10%) of the desired output value. The error band can be chosen depending on the general level of signal noise and the process-specific error tolerances.

Overshoot

The overshoot is given as a percentage of the step amplitude and is defined as the ratio $\frac{b}{a}$ in Figure 2.4, describing the maximum relative error that can be observed after applying a step reference signal to the controlled system. If the output signal is never larger than the step amplitude (or stays within a pre-defined error band), the overshoot is considered to be 0.

2.5.2. Output Error Integral

The area between the reference signal and the measured output can be calculated and used as a general measure of how well the reference trajectory is followed over a given time period. The integral of $|y_{ref} - y_{meas}|$ is used to ensure that diversions from the reference signal in both directions increase this error term. For time-discrete signals, the integral can be approximated using the trapezoidal rule.

3. System Identification

The methods used for data collection and preparation are explained in the first section of this chapter. The results are then presented and discussed in sections 2 and 3. Finally, one of the identified models is chosen to be used for designing the model-based controllers in the following chapter.

3.1. Methodology

3.1.1. Data Collection and Pre-processing

Neither the base reference values of the system's outputs nor the system's desired operating range can be directly transferred from the fully assembled vehicle to the component rig (as mentioned in Section 2.2). To define suitable base reference values for the component rig, the range of values that can be observed on each of its outputs is determined first. The outputs' base reference values are then set to the midpoint of their respective value ranges. Next, the output extrema of the fully assembled vehicle are expressed as percentages of their respective base reference value (e.g. the largest observable y_1 value is 155% of its base reference value). These relative ranges can then be applied to the component rig's new base reference values to determine its desired operating range.

A PRBS signal is added to each actuator control signal to generate the input output data that will be used for system identification. The amplitudes of these PRBS signals are chosen such that the resulting output values cover the entire desired operating range. A section of one recorded dataset is shown in Figure 3.1.

There can be slight variations in the recorded output levels between the different datasets, caused by external factors such as the ambient temperature of the component rig. To ensure that these variations do not negatively affect the system identification process, all datasets are centered around 0. The gradual temperature increase of the engine oil during operation can cause linear trends in the recorded data that are not related to the input output dynamics. These trends can be identified and removed by comparing the output values at the beginning and the end of the recorded dataset. Figure 3.2 shows the same dataset section after applying these pre-processing steps.

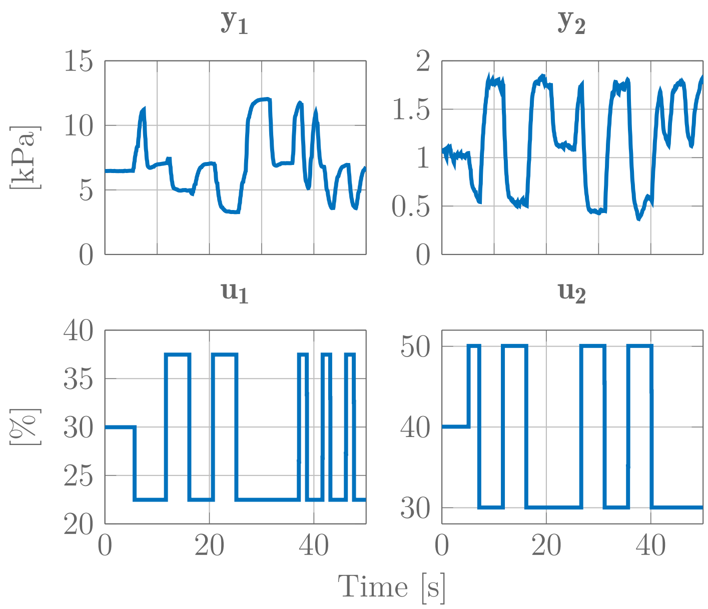


Figure 3.1.: Section of an input output dataset before pre-processing

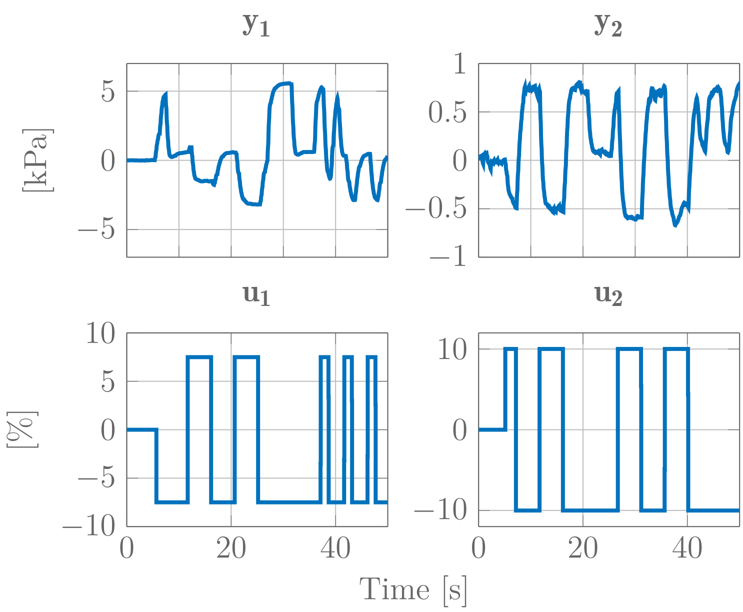


Figure 3.2.: Section of an input output dataset after pre-processing

To get a comprehensive overview of the system dynamics, this data collection procedure is repeated for a total of 12 combinations of oil temperatures and engine speeds, which are listed in Table 3.1. The datasets will be referred to by their indices in the following sections.

Table 3.1.: System Identification Testing Conditions

Index	Oil Temperature	Engine Speed
1	18°C	1000 rpm
2	18°C	1500 rpm
3	18°C	2000 rpm
4	38°C	1000 rpm
5	38°C	1500 rpm
6	38°C	2000 rpm
7	47°C	1000 rpm
8	47°C	1500 rpm
9	47°C	2000 rpm
10	56°C	1000 rpm
11	56°C	1500 rpm
12	56°C	2000 rpm

3.1.2. Estimation and Validation

The system identification process should generate a model that captures the system dynamics with a sufficient degree of accuracy across all testing conditions. If that is not possible, underlying structures in the obtained measurement data should be revealed that allow splitting the data into subsets and estimate a model for each of these subsets. To achieve these goals, two methods for separating the collected input output datasets into estimation and validation data were developed.

Single Dataset Estimation

For single dataset estimation (SDE) each of the datasets is used separately to estimate a model, yielding 12 models in total. Each of these models can then be validated on the remaining datasets that were not used during the estimation process. An illustration of this method is shown in Figure 3.3.

The obtained fit percentages can be arranged into a matrix, with each row representing one model (estimated on the dataset corresponding to the row number) and each column entry of that row showing the fit percentage for this combination of model and dataset. For the diagonal entries of the matrix, the fit percentage returned by the estimation process is used. All other matrix entries contain a validation fit

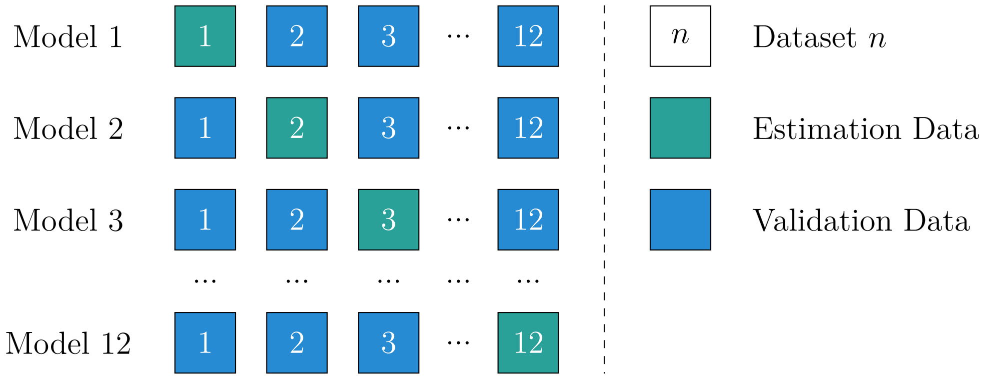


Figure 3.3.: Illustration of the SDE method

percentage that shows how well the model can recreate the output data of datasets that were not used during the estimation process.

The fit percentage matrix provides valuable insights into both the generated models as well as the underlying datasets. Overfitting is revealed when a model's diagonal matrix entry is significantly higher than the remaining entries of this row, whereas consistently high fit values in an entire row signify the general validity of this row's model. Low fit values in an entire column show that this dataset is particularly difficult to recreate, which can reveal errors in the data collection process or indicate nonlinear system behaviour. A block-diagonal structure, where some models only result in good fit percentages on neighbouring datasets, is an indicator that better control performance may be achieved by splitting up the dataset, identifying multiple models and applying techniques like gain scheduling or robust control.

Combined Dataset Estimation

For combined dataset estimation (CDE), each dataset is split in half to create two subsets of the data, one for estimation and one for validation. All estimation subsets are combined into one large estimation dataset that is provided to the system identification algorithm. This estimated model is then validated separately on each of the validation datasets. An illustration of this method is shown in Figure 3.4.

Conceptually, CDE has the advantage of estimating a model from a large dataset that encompasses data from all relevant operating conditions, whereas SDE can be regarded as a tool for identifying specific operating conditions (and the corresponding dataset and model) that are representative of a larger operating range.

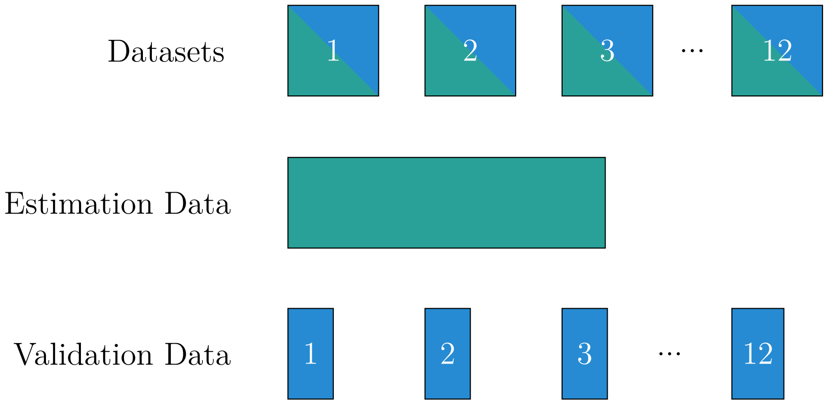


Figure 3.4.: Illustration of the CDE method

3.2. Results

The system identification results obtained from the estimation of a two-state state-space (SS2) model are presented in detail to demonstrate the process used for analysing the system identification results. The same process has been used to inspect the remaining models, but the discussion of their results will be limited to more general observations and comparisons in Sections 3.2.2 and 3.2.3. The relevant plots for these models can be found in Appendix A.

3.2.1. Two-state State-space Model

SDE

Figure 3.5 shows a heatmap visualisation of the fit percentage matrix of the SS2 model. It is noticeable that columns 1 to 3, which represent the datasets collected at cold oil temperature, have consistently low fit percentages of around 50%. This could indicate nonlinear system behaviour at cold oil temperatures, which can not be captured and recreated by a linear model.

In Figure 3.6, the row-average of the fit percentage matrix is visualised, showing the mean fit percentage across all datasets for every estimated model. There is little variation here, with the lowest and highest mean fits only separated by 5 percentage points. The model estimated on dataset 6 achieves the highest average fit of 58.1%. This model will be used for the following inspections and comparisons and will be referred to as SDE6 for clarity.

For the initial model assessment via the fit percentage matrix, the fit percentages of y_1 and y_2 are averaged. After identifying model 6 as the model with the highest average fit percentage, the model's individual output fits are inspected to ensure that both y_1 and y_2 are modelled well. This analysis is shown in Figure 3.7. While there is some variation between the fit percentages of y_1 and y_2, the discrepancy is

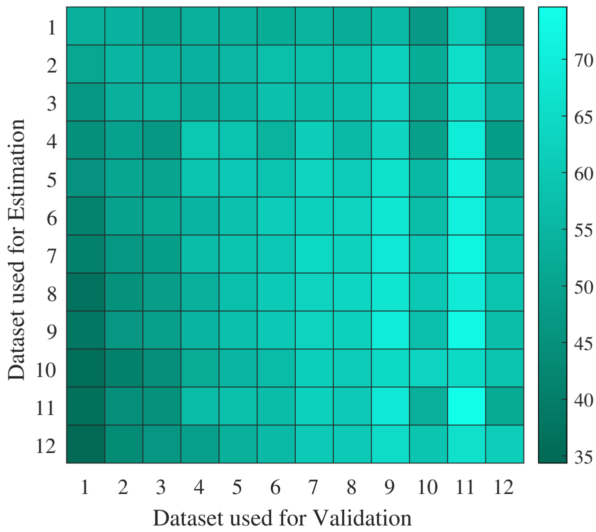


Figure 3.5.: SS2 model - fit percentage matrix obtained through SDE

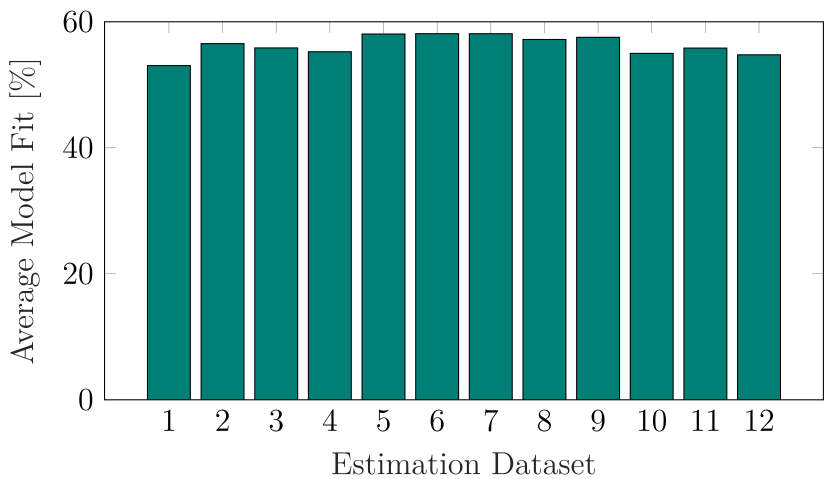


Figure 3.6.: SS2 model - row averages of the fit percentage matrix

only once larger than 10 percentage points (on dataset 5) and the average across all validations is almost identical at just over 58%.

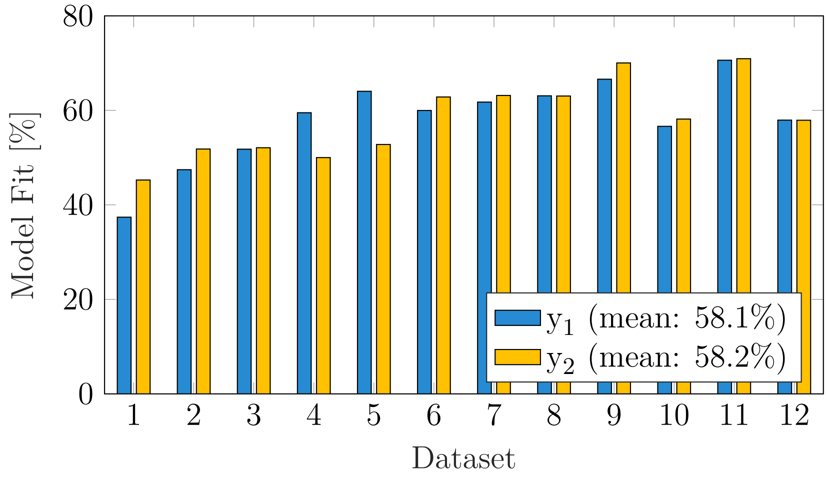


Figure 3.7.: SS2 model (SDE6) - detailed output fit percentages

CDE

As explained in Section 3.1, CDE only estimates a single model and thus does not produce a fit percentage matrix. The estimated model's individual fit percentages for y_1 and y_2 are shown in Figure 3.8. Compared to the best model of the SDE, the discrepancies between the output fits are slightly higher and tend to favour y_1 over y_2. This is also evident in the mean values of the output fit percentages, where a difference of 3.5 percentage points can now be observed between y_1 and y_2.

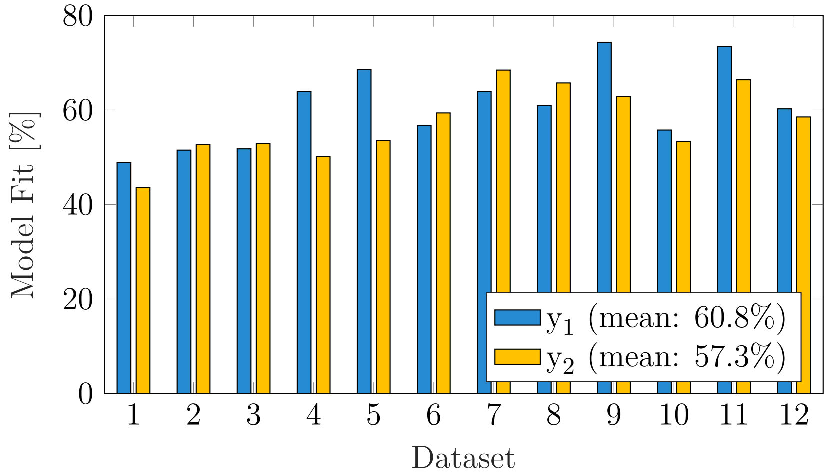


Figure 3.8.: SS2 model (CDE) - detailed output fit percentages

Comparison of SDE and CDE

After confirming that both SDE and CDE produce models with acceptable fit differences between y_1 and y_2, the average of the two output fit percentages will be used again for the remaining comparisons. In Figure 3.9 the per-dataset fit percentages for the SDE6 model and the CDE model are shown.

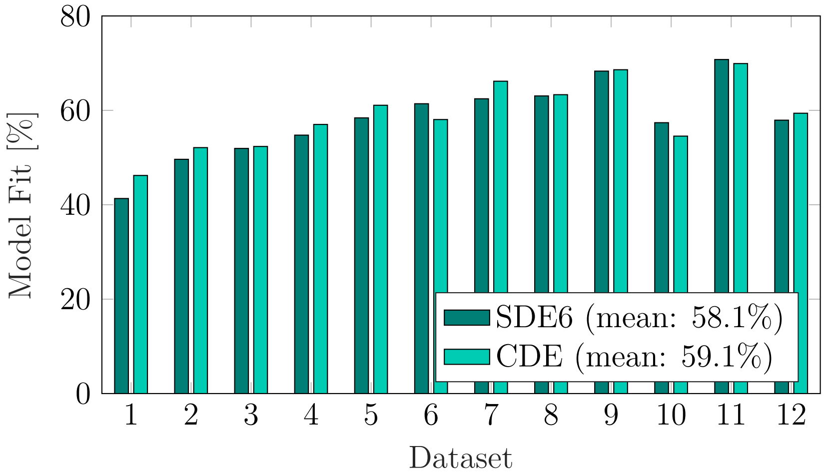


Figure 3.9.: SS2 model - fit percentage comparison between SDE6 and CDE

For this combination of datasets and model choice, the difference between the two estimation strategies is negligible. Nonetheless, some effects of the two different approaches can be observed. The SDE6 model is only able to achieve a better fit percentage than the CDE model on three datasets (one of them being dataset 6, on which the model was estimated and where a good fit is thus expected). This can be attributed to the fact that during the system identification process, the CDE model is iteratively optimized on a dataset comprising data from all operating conditions. The advantage of the SDE method lies in the number of models created. This allows picking a model that is also able to achieve a good fit on datasets from operating conditions it has not encountered before, sometimes even surpassing the performance of the CDE model.

3.2.2. Higher-order state-space models

Next, the effect of the state-space order on the obtained model fit is investigated. For this purpose, state-space models with an order of $n = 4, 6, 8, 10$ (referred to as SSn) are identified and compared to the SS2 model. Figure 3.10 shows the average fit across all datasets of those state-space models, both for the best model obtained through SDE and the single CDE model.

While there are marginal improvements in model accuracy when using 4 states (+3-4% over SS2) and 6 states (+5-6% over SS2) for the state-space representation, there are no discernible benefit when further increasing the number of states to 8 or

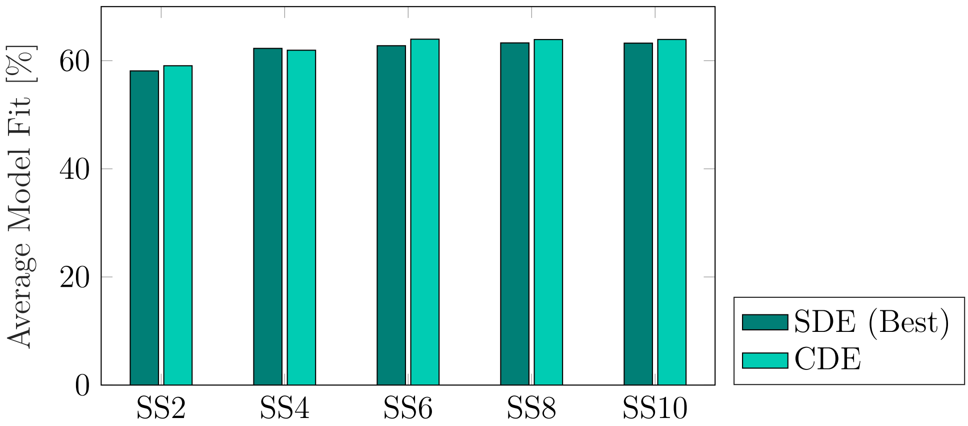


Figure 3.10.: Mean fit across all datasets for state-space models of different orders

10. In Figure 3.11 the poles and zeros of the SS10 CDE model are plotted together with a circle that indicates the confidence region for 2 standard deviations for each of them. The overlapping confidence regions of poles and zeros that can be observed are an indicator that this pole-zero pair can be removed without a noticeable impact on model accuracy, proving that there is no benefit in increasing the model order further.

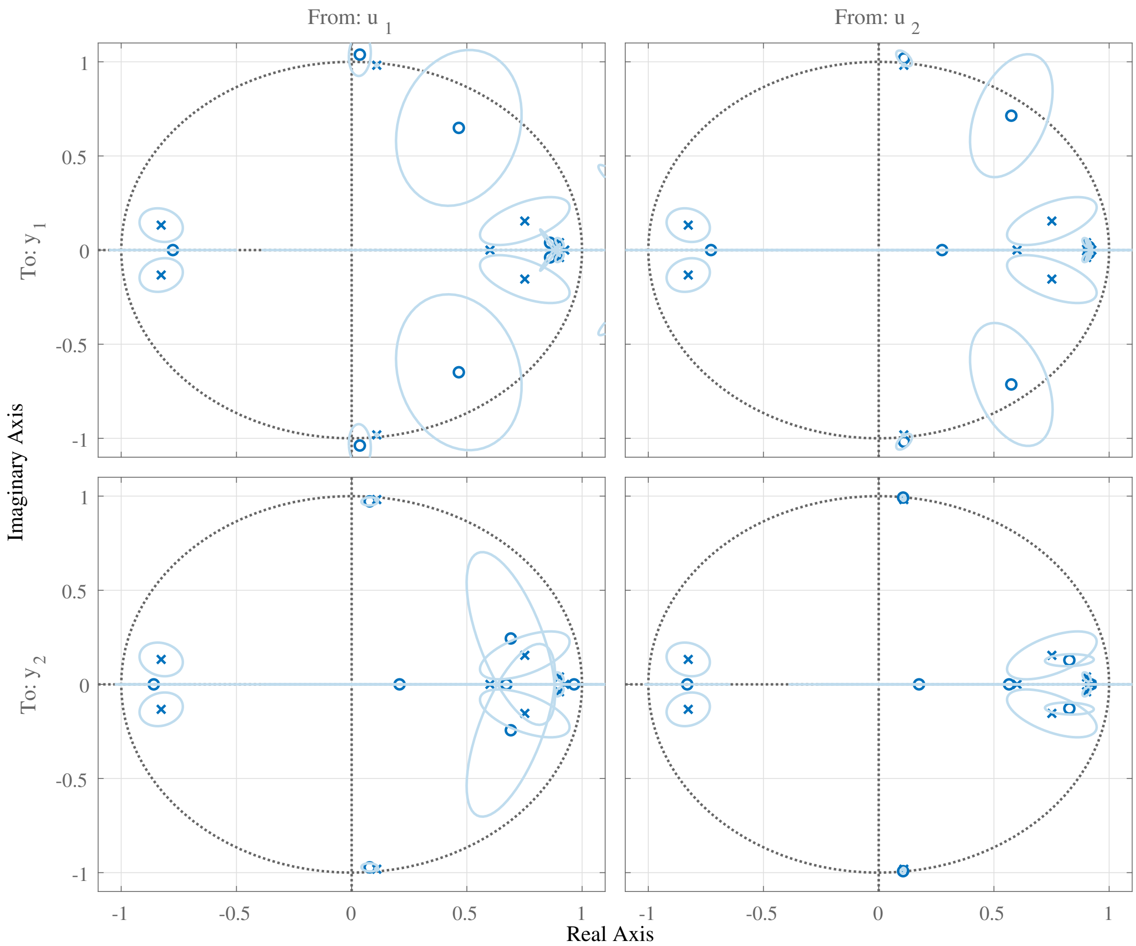


Figure 3.11.: Pole-zero map of the SS10 CDE model with confidence regions for 2 standard deviations

3.2.3. Polynomial models

Polynomial models were investigated alongside traditional state-space models, more specifically the polynomial model structures ARX, ARMAX and BJ. For each of them, orders 1 to 3 were applied to all polynomials used in the respective model structure, indicated by the number after the acronym.

Figure 3.12 shows the average fit across all datasets for all estimated polynomial models. The best average fit percentage of 65% is achieved by the BJ model of order 2, demonstrating that the tested polynomial models only result in single-digit improvements in model accuracy for this dataset.

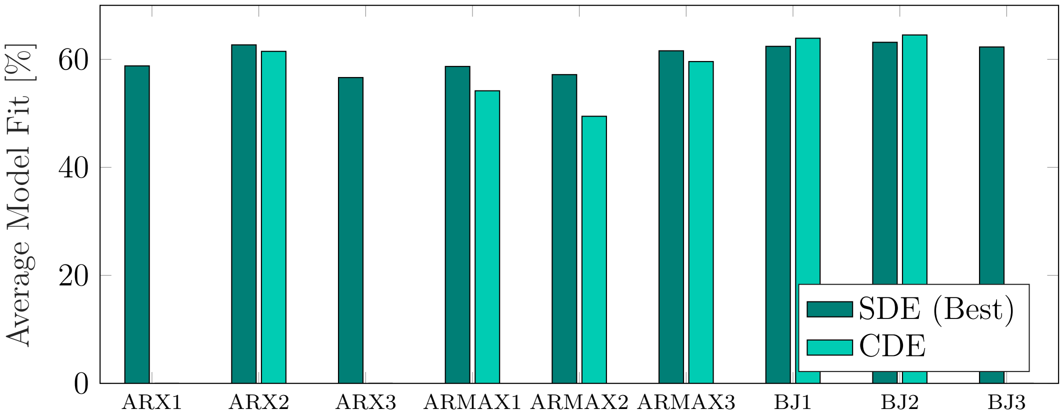


Figure 3.12.: Mean fit across all datasets for polynomial models of different orders

The fit percentages achieved by SDE and CDE are again in close proximity to each other, with the notable exception of ARX models with orders 1 and 3 and the BJ model of order 3, where the models obtained through CDE fail to recreate the measured data with sufficient accuracy.

3.3. Discussion

Despite the wide range of investigated models, it was not possible to increase the average fit percentage across all datasets beyond 65%. Figure 3.13 shows a comparison of a validation dataset's measured system outputs and the system outputs generated by an estimated SS2 model based on the same input data. The estimated model is able to capture the general input output dynamics, but sometimes fails to accurately recreate the signals (e.g. from 17s to 21s or between 41s and 45s).

The system behaviour that can not be recreated by these models is most likely not linear in nature. Initial experiments confirm this assumption by demonstrating that model accuracy can be increased further by estimating nonlinear models such as wavelet- or treepartition-based models. Since nonlinear models require a conceptually

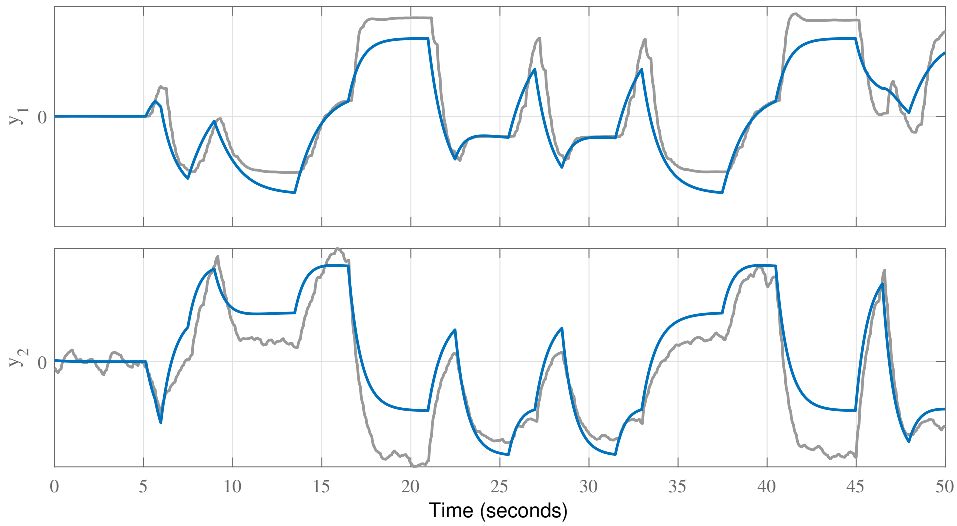


Figure 3.13.: SS2 - simulated outputs (blue) and measured outputs (grey)

different control approach, they will not be further investigated and discussed in this thesis but can be explored in future work.

Across all models, the observable differences between SDE and CDE were marginal. A clear benefit of SDE is the generated fit percentage matrix, which offers valuable insight in the quality of the underlying data and can reveal block-diagonal structures that can then be used to separate the collected data into smaller subsets. This was however not necessary for the component rig test system.

3.4. Model Choice for Controller Design

Choosing an underlying model is a fundamental first step of model-based control. The structure of the model as well as additional information such as the model's accuracy influence the subsequent step of designing a controller.

For this thesis, the SS2 model structure was chosen. It provides a fit percentage that is within single-digit range of the maximum achieved fit and has the benefit of low overall complexity. Since the model was estimated in observable canonical form, the system states are identical to the measured system outputs, simplifying the design of state feedback control.

Neither state-space models of higher orders nor the estimated polynomial models offer any substantial gains in model accuracy and the increased number of states would require a more complex controller structure including state estimation through an observer or a Kalman Filter.

The detailed comparison of the SDE and CDE methods in Section 3.2.1 showed a close similarity of the models obtained with the two strategies. The SDE model was ultimately chosen over the CDE model, since its average validation fit percentages for y_1 and y_2 were closer to each other (as shown in Figures 3.7 and 3.8).

4. Controller Design

This chapter outlines the decision process behind the chosen control strategies, along with some system-specific implementation details and the comparison baseline for the following evaluation chapter.

4.1. Linear-quadratic Control

The results from the previous chapter show that the chosen model does not capture the full dynamics of the engine oil system. While an LQR delivers good performance on accurately modelled systems, it results in steady-state errors in the presence of modelling inaccuracies, making it inadequate for satisfactory control of the engine oil system.

The deficiencies of the chosen model can be regarded as disturbances to the system, thus making LQI control a suitable choice for this control problem. Pure LQI control as proposed by [18] shows worse reference tracking performance than the tested LQR. For this reason, a version of LQI control that includes a feedforward term was implemented as well. This controller will be referred to as LQIf for the remainder of the thesis; its structure is shown in Figure 4.1.

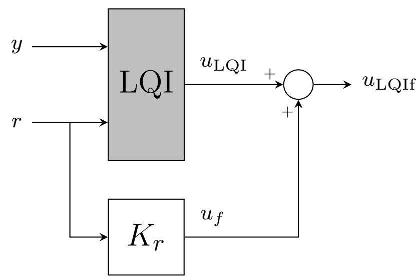


Figure 4.1.: Structure of the LQIf controller

For LQIf control, the control output generated by the LQI structure is augmented by the reference value r multiplied by K_r, which is determined the same way as in traditional LQR control (Section 2.4). It should be noted that only the first two columns of the 2-by-4 LQI-K-matrix are used for determining the transfer function of the closed loop system, since columns three and four are related to the integral of the control error.

4.2. Input Output Offsets

As described in Section 3.1, all models were estimated on data centered around the base operating output values, an approach that is conceptually similar to the manual derivation of a linearized model around a given operating point. Consequently, the obtained models describe the deviation dynamics from this operating point. Controllers based on such models expect reference and state signals that are offset in a similar way and will produce control signals that have to be added back to the input offsets subtracted before estimation. This results in the controller layout shown in Figure 4.2.

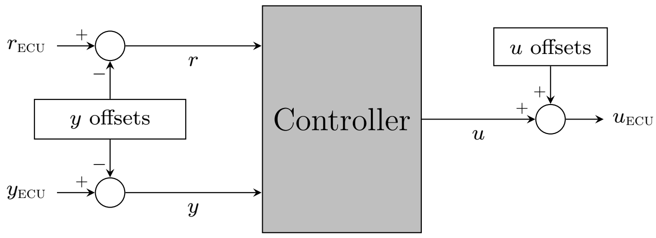


Figure 4.2.: Controller layout with applied offsets

4.3. Dynamic System State

The system state that is used inside the controllers is determined as follows:

$$x = y = y_{\text{ECU}} - [y \text{ offsets}] \tag{4.1}$$

During testing it was found that a second method of calculating the system state can lead to performance improvements compared to traditional LQI control: by using the reference values in place of the static y offsets, a more aggressive controller behaviour can be achieved:

$$\begin{aligned} x &= y_{\text{ECU}} - r_{\text{ECU}} \\ &= (y_{\text{ECU}} - [y \text{ offsets}]) - (r_{\text{ECU}} - [y \text{ offsets}]) \\ &= y - r \end{aligned} \tag{4.2}$$

Using this method, a change in reference values changes the system states, which are then brought back to a zero-equilibrium by the controller. It should be noted that this method assumes that the system's equilibrium point can be freely shifted within the operating range without changing the system dynamics.

In the following comparisons, the controllers using this dynamic state calculation method will be marked with a trailing letter d.

4.4. Comparison Baseline

The model-based controllers will be compared to a model-free proportional-integral (PI) controller, which approximates the currently used control strategies for the engine oil system and uses the same input signals as the linear-quadratic controllers: the measured system outputs and the integral of the control error. To ensure a fair comparison, an equal amount of time is spent on the tuning of the model-based controllers and the baseline controller.

5. Controller Evaluation

This chapter begins with an explanation of the test scenarios used for evaluating the controllers on the component rig. The test results are then presented and the performance of the model-based controllers is compared to that of the model-free PI controller. The results are then summarized and discussed further in the last section.

5.1. Methodology

As stated in Section 1.1, controllers of the engine oil system need to provide good reference tracking and disturbance rejection. For each of these requirements, separate test scenarios have been created with the goal of allowing a comparison of the controllers that is as fair and objective as possible.

Despite the differences between the component rig and the full engine oil systems that were discussed in Section 2.2, the results from these tests are indicative of the controller's behaviour in a fully assembled vehicle.

5.1.1. Reference Tracking

The reference tracking ability of the designed controllers is determined by 4 separate tests in which the reference values for y_1 and y_2 are modified. The reference trajectories of the first 2 step tests are shown in Figure 5.1. Each reference test is made up of 11 individual steps with target values of $\pm 20\%$, $\pm 40\%$ and $\pm 60\%$ of the baseline reference value. After reaching the maximum step size at step 6, the pattern is reversed. The time between steps is set to 10 seconds to give the output values enough time to settle around the reference value for most jumps while keeping the overall testing time within a reasonable limit. Cases where the reference value could not be reached in time are accounted for by setting the settling time for the affected step to 11 seconds. This method penalizes every non-settled step equally which could potentially lead to inaccuracies: two controllers reaching 120% and 150% of the reference value both get a settling time of 11s. During testing such occurrences were very rare, making this approximation acceptable for this specific use case.

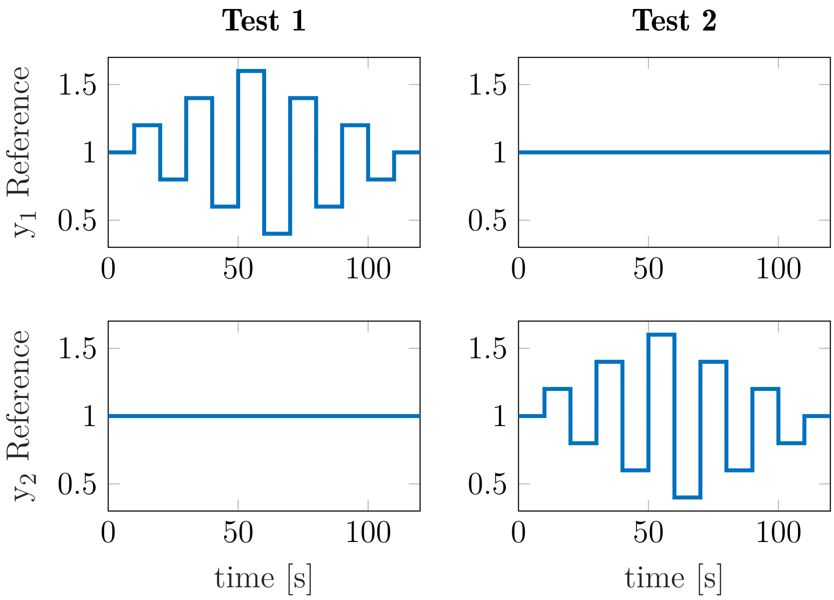


Figure 5.1.: Reference trajectories of tracking performance tests 1 and 2

By modifying the reference values in a step pattern (instead of e.g. ramping them up over a certain timespan), metrics such as settling time and overshoot can be determined for each reference change, which allow a more in-depth and standardized evaluation of the tested controllers.

Besides the independent stepping of the reference values, two additional step tests were carried out where the reference values for y_1 and y_2 were stepped at the same time. The simultaneous step tests were conducted twice, with both reference values stepping in the same direction during Test 3 and in opposite directions during Test 4 (Figure 5.2). To ensure that the reference values can be reached before the next step starts, the step amplitudes for these tests have been slightly decreased to ±15%, ±30% and ±45%.

For each step, the rise time, settling time and overshoot are determined. While the metrics change depending on the step size, these variations were found to be independent of the tested controller. For this reason, the results are then simply averaged across the 11 steps. Additionally, the total control error over the course of the entire test is evaluated by a trapezoidal integral approximation as described in Section 2.5.2.

5.1.2. Disturbance Rejection

To test the controllers' disturbance rejection abilities, the input signal offsets shown in Figure 4.2 are modified, which causes a sudden jump in the system's output values. Analysing how quickly the system is able to return to the (unchanged) output

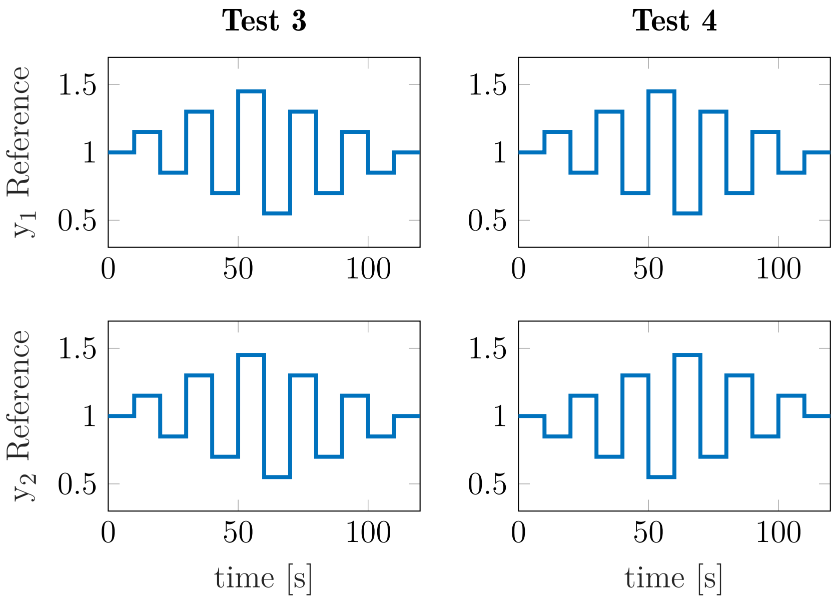


Figure 5.2.: Reference trajectories of tracking performance tests 3 and 4

reference values gives insights into each controller's ability to reject unexpected system disturbances. During each input's disturbance test, their respective offset value was changed 4 times, as shown in Figure 5.3.

Since the reference values for the outputs stay constant during these tests, each input disturbance can be regarded as a step with an amplitude of 0. Following this analogy, the settling time of such a step represents how quickly an output has returned back to its reference value. The settling time of the output corresponding to the disturbed input is first inspected in isolation, but the maximum of the settling times of both outputs is used as an additional metric to account for the actuators' cross-influences. Similar to the reference tracking tests, a failure to settle within 10 seconds after the disturbance occurred is accounted for by setting the settling time for that interval to 11 seconds. Finally, the overall disturbance error is evaluated by a trapezoidal integral approximation as described in Section 2.5.2.

It should be noted that other system disturbances can occur, such as a disturbance on the system output. Only input disturbance rejection is evaluated here to limit the scope of testing and due to technical reasons regarding the ease of implementation on the component rig.

5.2. Results

In this section, the controllers are evaluated with regards to ease of tuning and their performance in the reference tracking and disturbance rejection test scenarios. The engine speed was set to 1500 rpm during testing since initial tests showed no

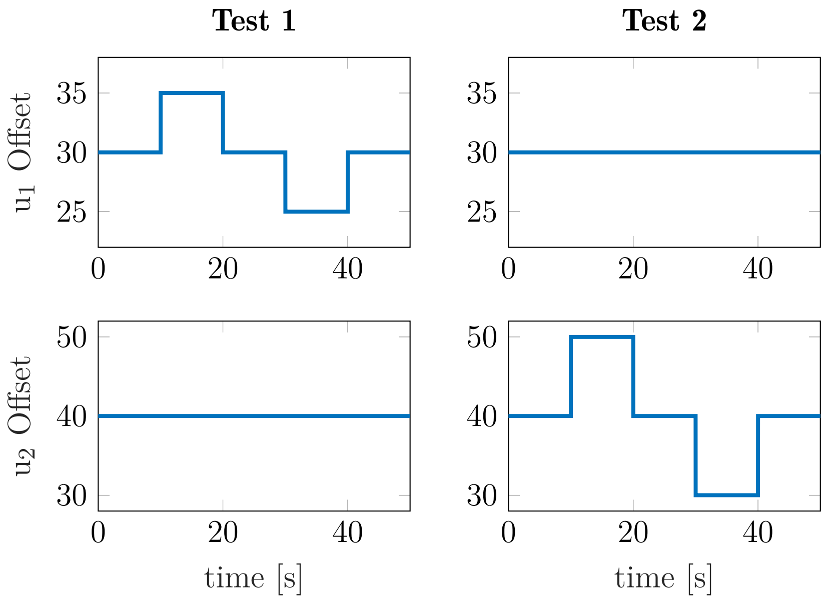


Figure 5.3.: Input offset values during disturbance tests

discernible difference between results obtained at 1000 rpm, 1500 rpm and 2000 rpm. All tests were then conducted at oil temperatures of 19°C, 38°C, 47°C and 56°C. To get an overview of the performance of all tested controllers, the average results across these four oil temperatures are shown here. Detailed results broken down by controller type, temperature and test scenario can be found in Appendix B.

5.2.1. Controller Tuning

While an equal amount of time has been spent on tuning all controllers to ensure a fair comparison, the tuning process for the model-based controllers is considerably more intuitive.

The layout of the engine oil system (Figure 2.1) results in a more difficult tuning process for the model-free controllers since the controller for one input influences the behaviour of the entire system and thus the tuning requirements for the other input. Model-based tuning of PI controllers is also possible but was not applied here to give a more realistic comparison between the currently employed method for oil pressure control and the new model-based strategies.

When using Bryson's rule for the initial values and with an understanding of the impact of the Q- and R-matrices' entries, the tuning process of linear-quadratic controllers is more straightforward in this case. The K_r-matrix of the LQIf controllers can be scaled up and down independently from the main feedback matrix if required, which adds a bit of complexity compared to pure LQI control.

All controllers have been tuned with the goal of optimizing overall performance for the described testing scenarios. For real-world applications, tuning is usually carried out in accordance with the maximum step amplitude of reference values and process-related limits for overshoot percentages and rise times. The controllers can then be tuned to stay within these limits during operation.

5.2.2. Reference Tracking

A comparison of the rise times achieved by the considered controllers is shown in Figure 5.4. LQId and LQIf show consistent rise time reductions of up to 58% over the PI controller, while LQI has a noticeably higher rise time in Test 1 and LQIfd shows a slightly increased rise time in Test 2.

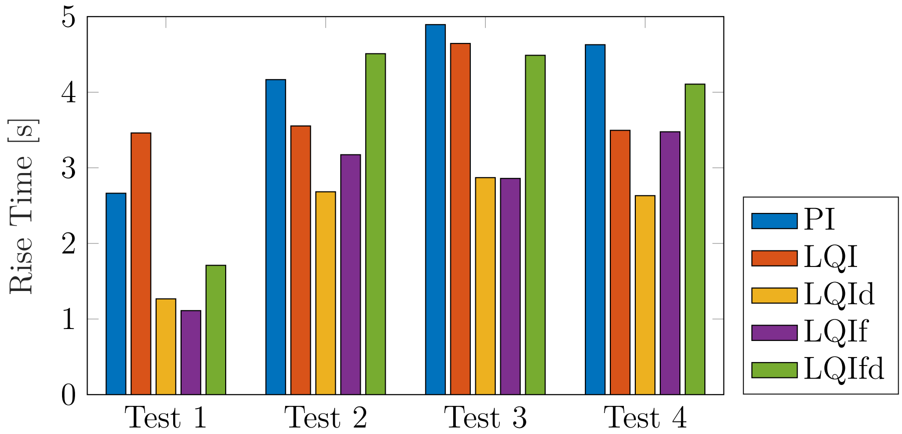


Figure 5.4.: Average rise times for each controller

Similar trends can be observed for the settling times, which are depicted in Figure 5.5. The lowest times are again achieved by LQId and LQIf. The LQI controller takes 0.7 seconds longer to settle than the PI controller in Test 1 but demonstrates improved settling times on the remaining tests. The LQIfd's settling times are very close to those of the PI controller in Tests 2 and 3, but improvements can be seen on Tests 1 and 4.

Figure 5.6 shows the average overshoot percentages for the reference tracking tests. The overshoot percentages of Test 2 are more than twice as high than those of Test 1. Since signal y_2 is almost an order of magnitude smaller than signal y_1, measurement noise has a higher relative impact on the signal, affecting the perceived overshoot. On Tests 1 and 2, all model-based controllers achieve lower overshoot than the PI controller. The LQI controller has the lowest overshoot values across all tests, the remaining model-based controllers achieve lower overshoot percentages than the PI controller on Test 4 but higher overshoot percentages on Test 3.

As a final metric for the controllers' reference tracking ability, the approximated integral of the absolute control error is shown in Figure 5.7. The best performance is

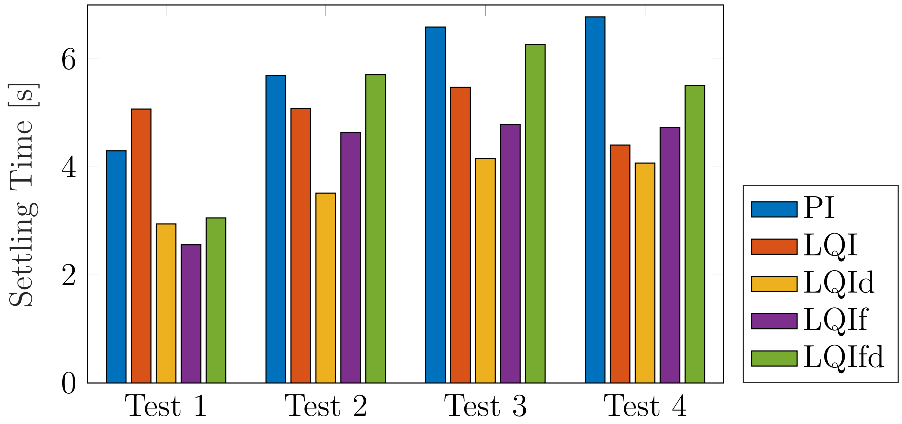


Figure 5.5.: Average settling times for each controller

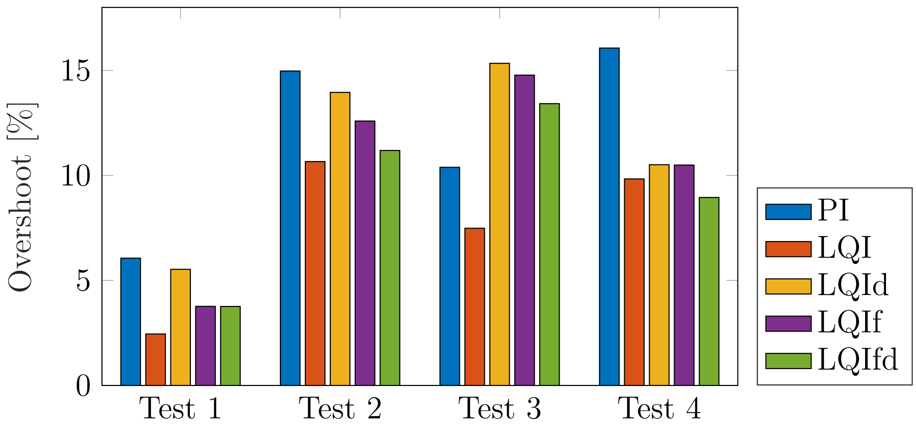


Figure 5.6.: Average overshoot percentages for each controller

achieved by the controllers LQId and LQIf, which both achieve a reduction of the integral value of ca. 40% compared to the PI controllers. The LQIfd controller's results are similar, although slightly worse in Tests 2 to 4. The LQI controller is on par with the PI controller during Test 1, features better error integrals on Tests 2 and 4 but a worse result on Test 3.

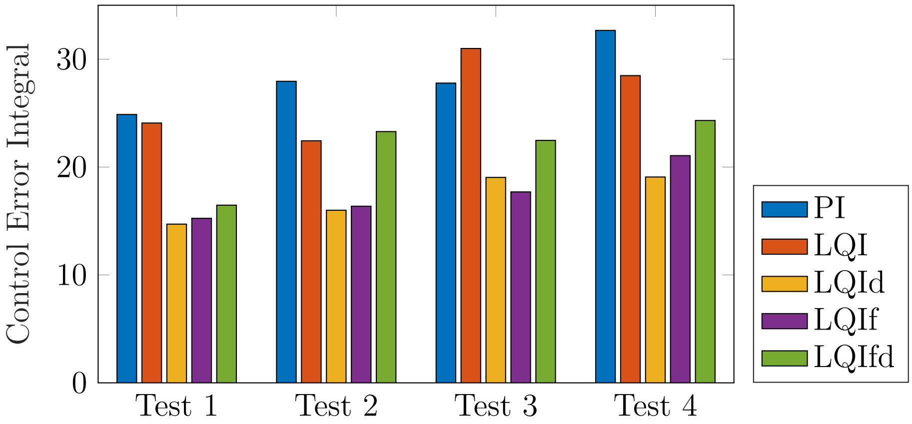


Figure 5.7.: Average control error integral for each controller

Overall, the controllers LQId and LQIf achieved the best results, which only performed worse on a single test and metric combination (overshoot on Test 3). The results for the other two model-based controllers were usually slightly better than those of the PI controller but fell behind the model-free controller's performance for several test and metric combinations.

In Figure 5.8 the reference tracking results from Tests 1 and 2 at 56°C are shown for the PI, LQId and LQIf controllers. The effects of the system layout are clearly visible in Test 2, where the reference steps on y_2 cause noticeable disturbances on y_1. While the disturbance peaks are of similar height for all three controllers, the model-based controllers are able to return to the baseline much quicker than the PI controller. The improvements in rise time are especially noticeable on the larger steps of both tests.

In the plot of Tests 3 and 4 (shown in Figure 5.9), the overshoot differences observed in the bar charts are very pronounced. The PI controller has noticeable overshoot in Test 4 that is drawn out across the larger steps. In Test 3, the model-based controllers show clear overshoot during the larger steps but are able to settle around the reference value a lot quicker.

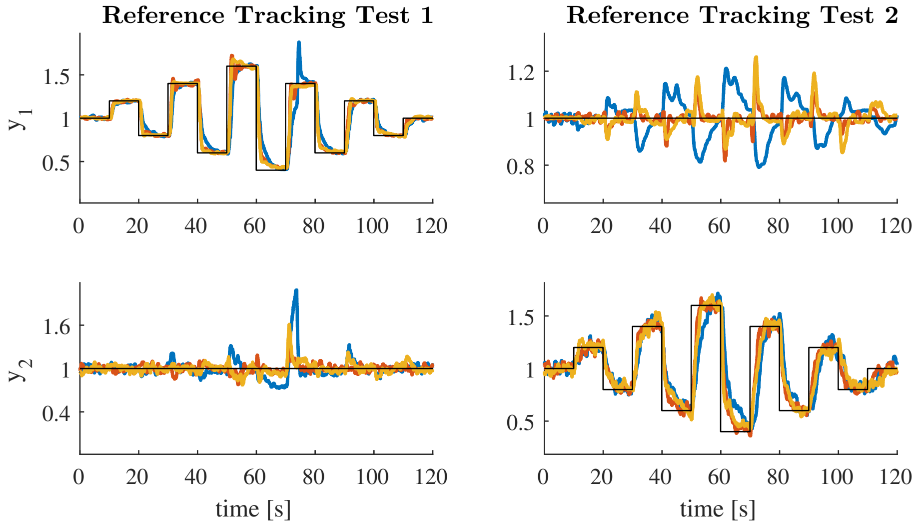


Figure 5.8.: Reference tracking tests 1 and 2 at 56°C, 1500 rpm
PI (blue) vs. LQId (red) vs. LQIf (yellow)

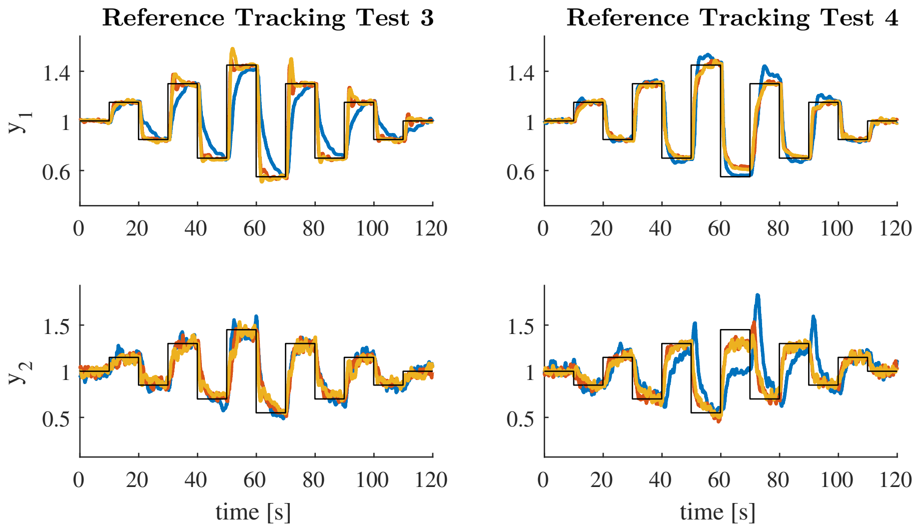


Figure 5.9.: Reference tracking - Tests 3 and 4 at 56°C, 1500 rpm
PI (blue) vs. LQId (red) vs. LQIf (yellow)

5.2.3. Disturbance Rejection

Figure 5.10 shows the average settling time achieved by each controller for Test 1 (disturbance at u_1) and Test 2 (disturbance at u_2). For this metric, only the settling time of the output corresponding to the disturbed input, i.e. y_1 for u_1 and y_2 for u_2, has been considered. The plot clearly shows that all model-based controllers improve upon the baseline performance of the PI controller. For disturbances on u_1, the LQI controllers without a feedforward component show a greater reduction than the LQIf controllers. The best result on Test 2 was achieved by the LQIfd controller, with LQId and LQIf following close behind.

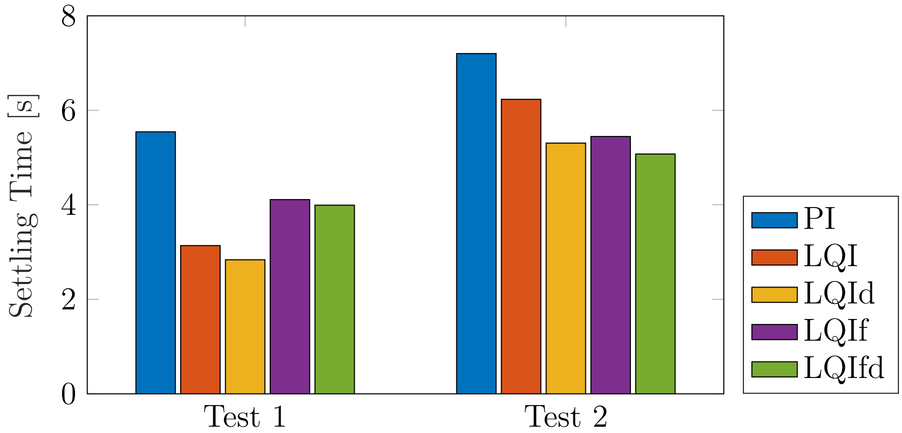


Figure 5.10.: Average settling time for each controller (disturbed output only)

In Figure 5.11 the maximum settling time of both outputs is considered instead. The most notable changes can be seen in the results of the PI controller, whose results increase by 0.6s for Test 1 and 0.9s for Test 2. For the model-based controllers results are mostly the same, with the biggest change being a 0.2s increase in the LQIf's result for Test 1.

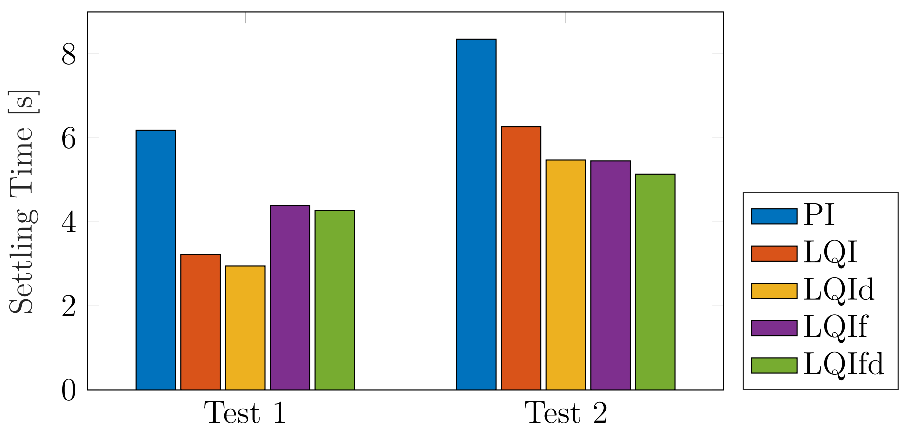


Figure 5.11.: Average settling time for each controller (maximum of both outputs)

The control error integrals are shown in Figure 5.12. The results are in line with the previous comparisons and show clear improvements for all model-based controllers (around 30% on Test 1 and 20% on Test 2). The LQId controller achieves the best performance in both tests.

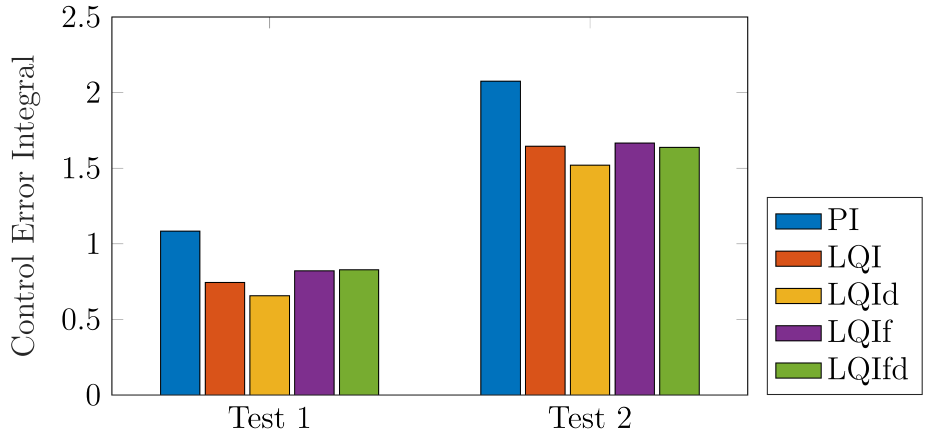


Figure 5.12.: Average control error integral for each controller

For the PI, LQId and LQIf controllers, the disturbance tests are shown in detail in Figure 5.13. In Test 1, the effects of the disturbances on y_1 are less severe during model-based controller tests, while y_2 is mostly affected by signal noise. Disturbances on u_2 clearly affect both output signals. The PI controller tends to overcorrect the deviation on y_1 resulting in the longer overall settling time that could be observed in the previous graphs. On y_1 the disturbance peaks are usually the same height, but the model-based controllers are again able to return the signal to its original reference point quicker than the PI controller.

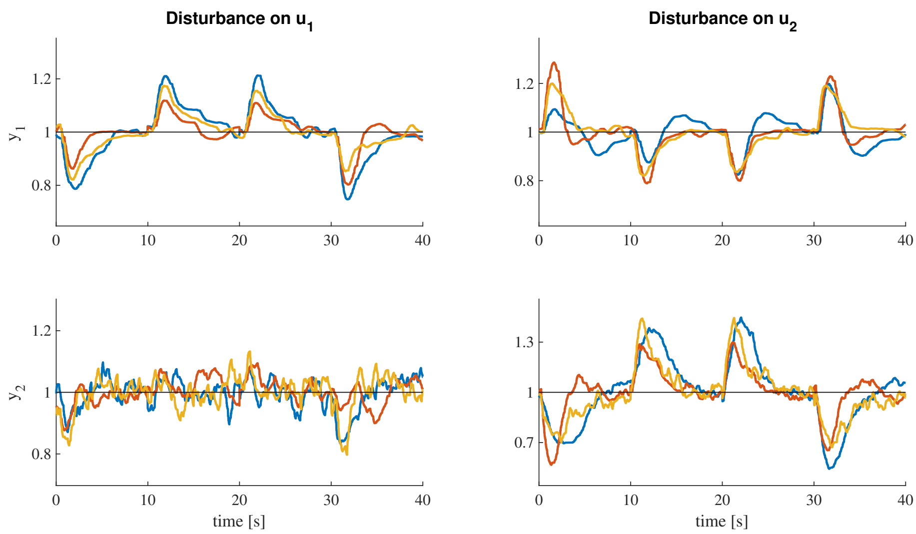


Figure 5.13.: Disturbance rejection tests at 19°C, 1500 rpm
PI (blue) vs. LQId (red) vs. LQIf (yellow)

5.3. Discussion

Table 5.1 showcases the results of the previous section. Evidently, the controllers LQId and LQIf achieve the most consistent improvements for all considered criteria. The increased effort required for their implementation is outweighed by the measurable gains in performance and the more intuitive tuning process. A performance decrease compared to the PI controller could only be observed on the overshoot metric of one of the four test scenarios. This is additionally relativized by the applied tuning process, which favoured overall control performance at the cost of overshoot in some scenarios. If the controllers are implemented in an industrial context, the controller tuning can be adjusted to meet specific requirements.

The information about the system dynamics that was gained from the system identification step can be fully utilized in the feedforward term of the LQIf controller, giving it an inherent advantage over the model-free PI controller. Any modelling inaccuracies are compensated by the LQI controller's inherent steady-state error correction.

From Figures 5.8, 5.9 and 5.13 it is clear that the dynamic state calculation applied in the LQId controller yields similar results. Whenever there is a change in reference values, the system state changes as well. This change in system state is then acted upon by the controller's K matrix.

The LQI controller acts only on the accumulated integral error and thus suffers from slower overall performance which is somewhat remedied by its good performance with regards to overshoot. The combination of the feedforward component and the dynamic system state required less aggressive overall tuning for the LQIfd controller, negating any beneficial effects these two modifications have when used in isolation.

It should be noted that model-free controllers like the PI controller considered here have a long history of being used across industries and benefit from the low overall complexity of the approach. When proposing the use of more involved control strategies, the results thus have to be clearly in favour of the new method to overcome decision inertia within the industry.

Table 5.1.: Comparison of all tested control strategies

	Effort		**Reference Tracking**			**Disturbance Rejection**	
Controller	Implementation	Tuning	Performance	Overshoot	Error	Performance	Error
PI	●	◔	◔	◑	◔	◔	◔
LQI	◑	◕	◑	●	◑	◑	◕
LQId	◑	◕	●	◑	●	●	●
LQIf	◑	◑	◕	◕	●	◕	◕
LQIfd	◑	◑	◕	◕	◕	◕	◕

Worst ◔ → ◑ → ◕ → ● Best

6. Conclusion

In this chapter, the achieved results are summarized and possibilities for additional research and improvements are discussed.

6.1. Contributions

In Chapter 3, the capabilities of linear system identification for the investigated engine oil system were explored. Linear models were able to recreate the system behaviour of the collected datasets with a maximum accuracy of 65%. The system behaviour unexplained by the identified models is presumably nonlinear in nature. The estimation of state-space models with high orders and polynomial ARX, ARMAX and BJ models resulted only in single-digit accuracy gains compared to a two-state state-space model, which was able to recreate the measured output data with an accuracy of 58% across all datasets.

The two-state state-space model was used to design linear-quadratic controllers, whose performance was compared to that of a model-free PI controller in Chapter 5. Controlling the engine oil system with model-based controllers resulted in quicker rise and settling times in reference tracking tests, faster return to the reference value during input disturbance tests and a reduction of the control error integral in both of these test cases.

6.2. Further Research

An important next step is applying the methods devised in this thesis to a fully assembled test vehicle to investigate the degree to which the results of this thesis are directly transferrable. If the fit percentage matrix of the SDE reveals a block-diagonal structure, gain scheduling can be applied to the linear quadratic controllers or a robust control approach can be tested. Observer structures or state-estimation through a Kalman Filter can be combined with the linear-quadratic controllers for added noise reduction or if higher-order state-space models result in a noticeable increase in model accuracy.

Other works have successfully demonstrated an automated controller tuning approach for linear-quadratic controllers based on Bayesian optimization [21]. The described method focuses on optimizing information gain from each tuning step, making it highly suitable for industrial applications.

In initial tests, the identified models were able to predict the system behaviour for a limited time horizon with a high degree of accuracy. This indicates that a combination of the existing system identification process with model predictive control (MPC) strategies could lead to good results and should thus be investigated in future projects. It should be noted that MPC requires more implementation effort and potentially significant additional computational resources, as it is a fundamentally different control strategy than the state feedback controllers tested in this thesis.

Finally, the methods of this thesis can be extended through nonlinear system identification and nonlinear control strategies, both of which have been successfully applied to similar tasks in other works [22, 23].

Bibliography

[1] M. Devlin. “Common Properties of Lubricants that Affect Vehicle Fuel Efficiency: A North American Historical Perspective”. In: *Lubricants* 6.3 (2018), p. 68.

[2] F. Will. “Fuel conservation and emission reduction through novel waste heat recovery for internal combustion engines”. In: *Fuel* 102 (2012), pp. 247–255.

[3] L. Ljung. *System Identification*. Linköping: Linköping University Electronic Press, 2007.

[4] L. Ljung and T. Glad. *Modeling of dynamic systems*. Englewood Cliffs, N.J.: Prentice Hall, 1994.

[5] T. Söderström and P. Stoica. *System identification*. New York and London: Prentice Hall, 1989.

[6] J. Schoukens and L. Ljung. “Nonlinear System Identification: A User-Oriented Road Map”. In: *IEEE Control Systems Magazine* 39.6 (2019), pp. 28–99.

[7] H. Jansson. “Experiment design with applications in identification for control”. PhD thesis. Stockholm: Royal Institute of Technology, 2004.

[8] M. Barenthin. “On input design in system identification for control”. Licentiate Thesis. Stockholm: Royal Institute of Technology, 2006.

[9] L. Ljung. *Characterization of the Concept of ’Persistently Exciting’ in the Frequency Domain*. Department of Automatic Control, Lund Institute of Technology (LTH), 1971.

[10] L. Ljung. *System identification: Theory for the user*. 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999.

[11] N. S. Nise. *Control systems engineering*. 6th ed. Hoboken, N.J.: Wiley, 2011.

[12] L. Ljung. “Some Classical and Some New Ideas for Identification of Linear Systems”. In: *Journal of Control, Automation and Electrical Systems* 24.1-2 (2013), pp. 3–10.

[13] T. Bohlin. *Practical grey-box process identification: Theory and applications*. London: Springer, 2006.

[14] A. K. Tangirala. *Principles of system identification: Theory and practice*. Boca Raton: CRC Press Taylor & Francis Group, 2015.

[15] The MathWorks. *Loss Function and Model Quality Metrics*. URL: `https://www.mathworks.com/help/ident/ug/model-quality-metrics.html`.

[16] K. P. Murphy. *Machine learning: A probabilistic perspective*. Cambridge MA: MIT Press, 2012.

[17] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. New York: Wiley Interscience, 1972.

[18] P. C. Young and J. C. Willems. "An approach to the linear multivariable servomechanism problem". In: *International Journal of Control* 15.5 (1972), pp. 961–979.

[19] A. E. Bryson. *Control of spacecraft and aircraft*. Princeton, N.J. and Chichester: Princeton University Press, 1994.

[20] S. Skogestad and I. Postlethwaite. *Multivariable feedback control: Analysis and design*. 2nd ed. Chichester: Wiley, 2005.

[21] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe. "Automatic LQR tuning based on Gaussian process global optimization". In: *2016 IEEE International Conference on Robotics and Automation (ICRA 2016)*. Piscataway, NJ: IEEE, 2016, pp. 270–277.

[22] J. Schlagenhauf. "Nonlinear model predictive control of a constrained model airplane". Master's Thesis. Albert-Ludwigs-Universität Freiburg, 2017.

[23] S. Tayamon. "Nonlinear System Identification and Control Applied to Selective Catalytic Reduction Systems". PhD thesis. Uppsala University, 2014.

List of Figures

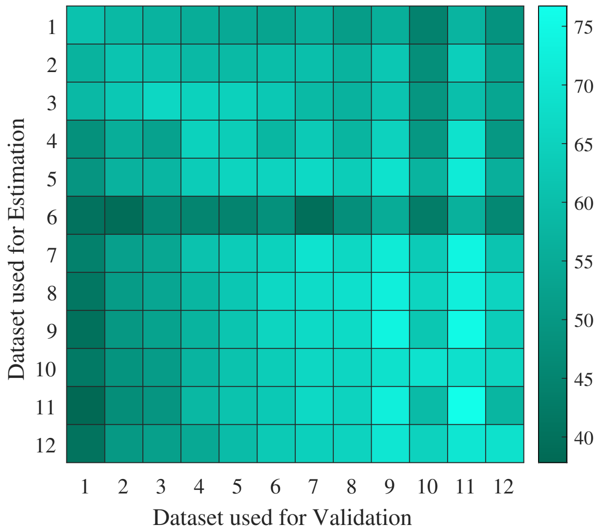
List of Tables

List of Acronyms

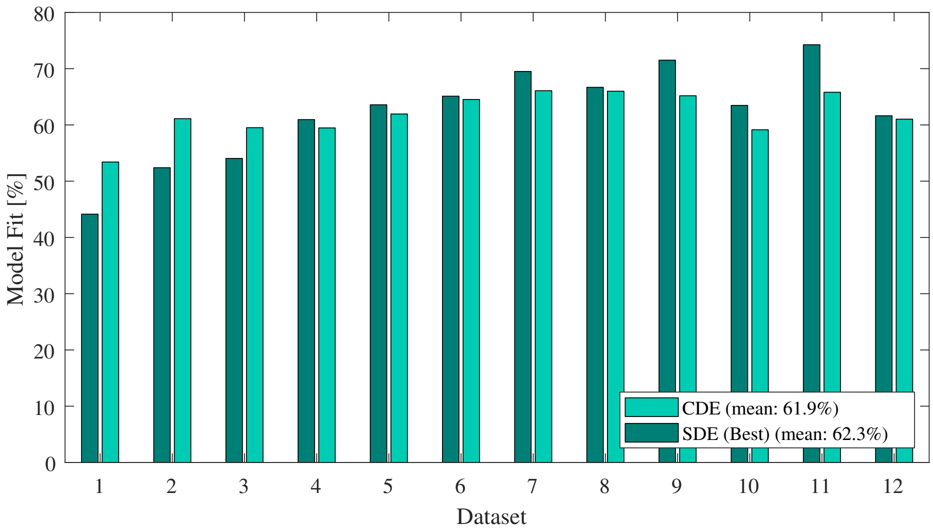
LQR	linear-quadratic regulator
LQI	linear-quadratic integral
ECU	electronic control unit
PRBS	pseudo-random binary sequence
ARE	Algebraic Riccati Equation
SISO	single input single output
MIMO	multiple input multiple output
OEM	original equipment manufacturer
MPC	model predictive control
SDE	single dataset estimation
CDE	combined dataset estimation
SS2	two-state state-space
ARX	autoregressive with exogenous input
ARMAX	autoregressive moving average with exogenous input
BJ	Box-Jenkins
PI	proportional-integral

A. Additional System Identification Plots

SS4

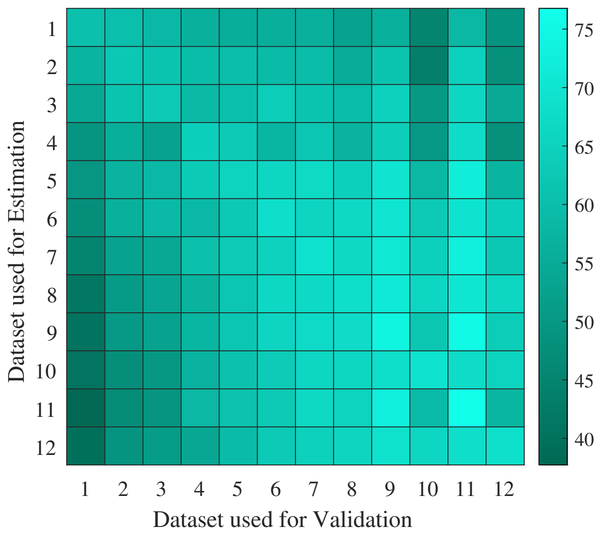


SS4 model - fit percentage matrix obtained through SDE

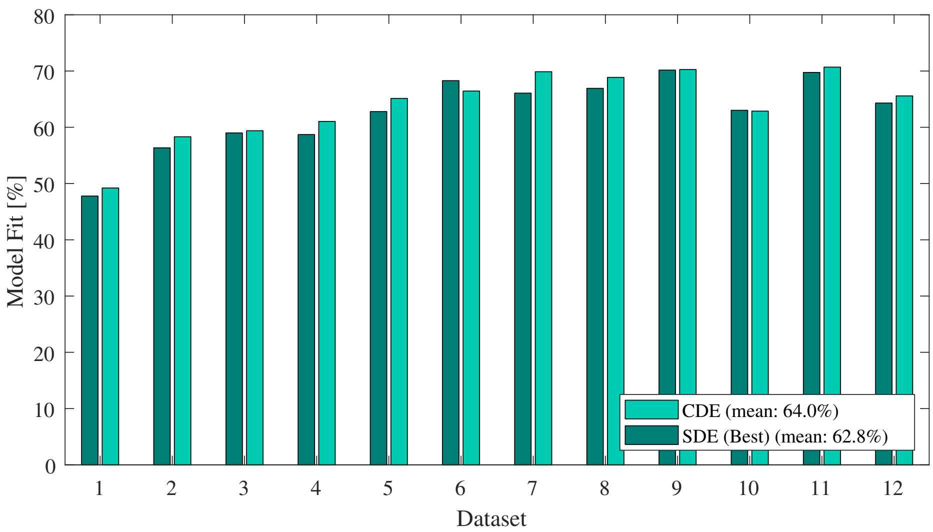


SS4 model - fit percentage comparison between SDE and CDE

SS6

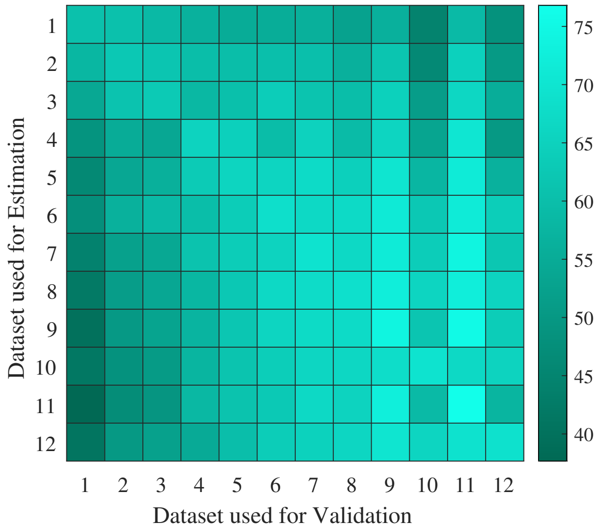


SS6 model - fit percentage matrix obtained through SDE

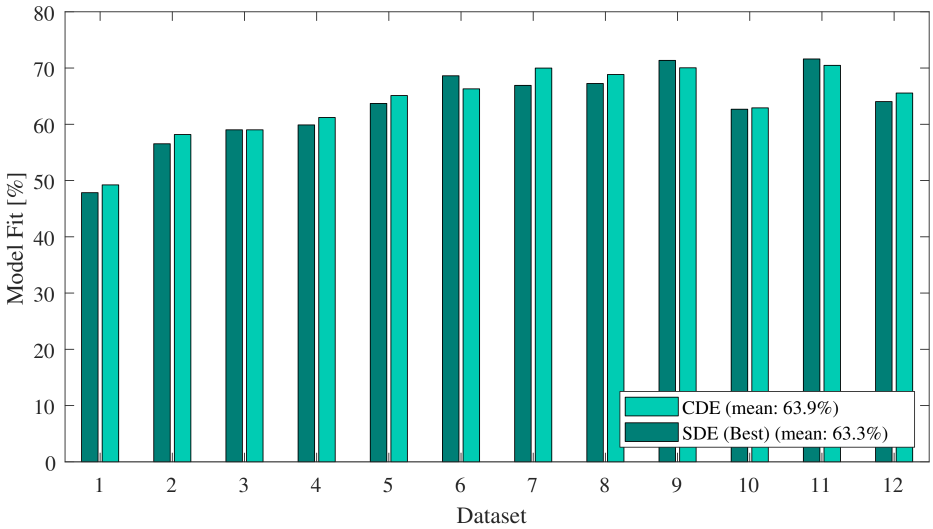


SS6 model - fit percentage comparison between SDE and CDE

SS8

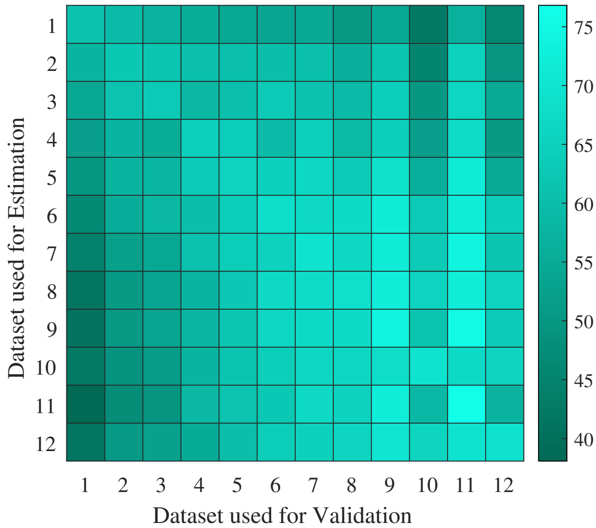


SS8 model - fit percentage matrix obtained through SDE

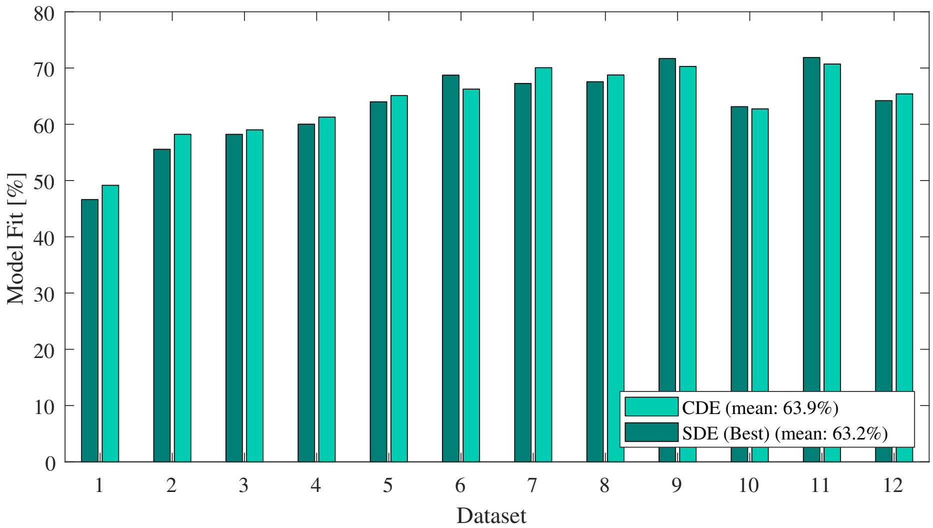


SS8 model - fit percentage comparison between SDE and CDE

SS10

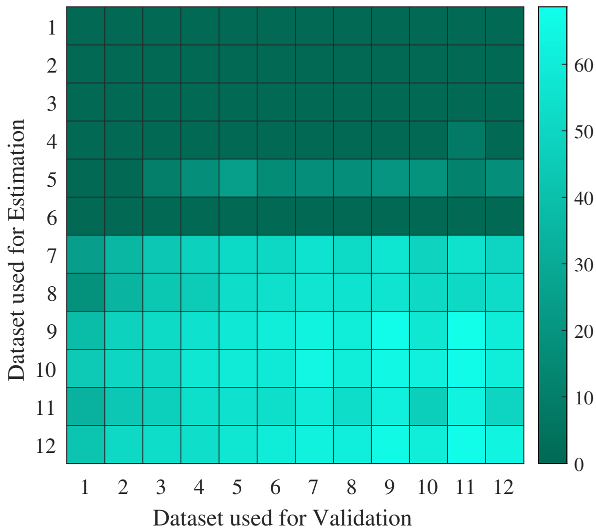


SS10 model - fit percentage matrix obtained through SDE

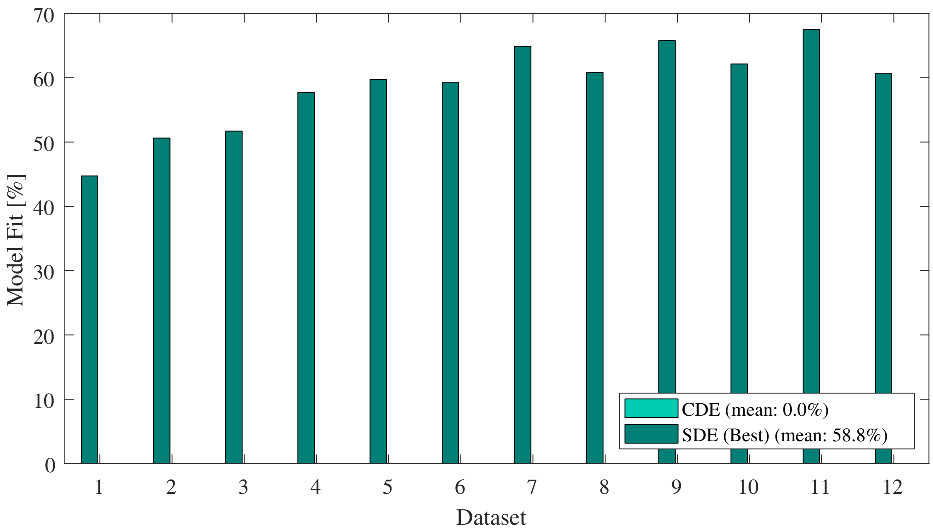


SS10 model - fit percentage comparison between SDE and CDE

ARX1

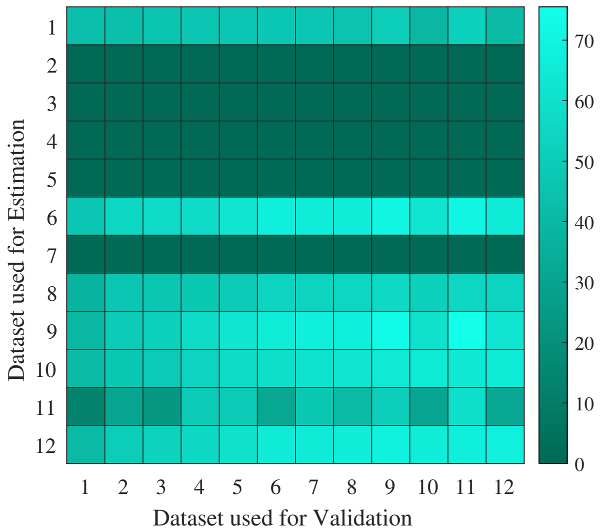


ARX1 model - fit percentage matrix obtained through SDE

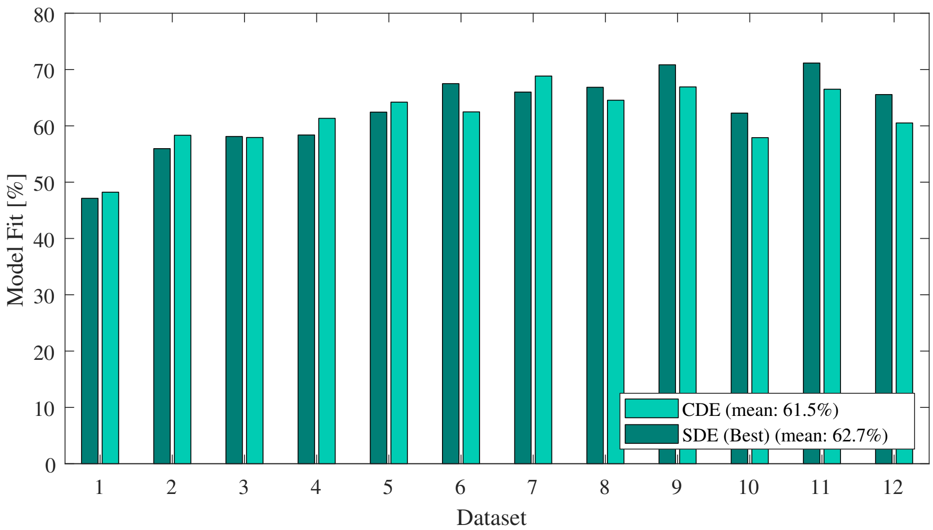


ARX1 model - fit percentage comparison between SDE and CDE

ARX2

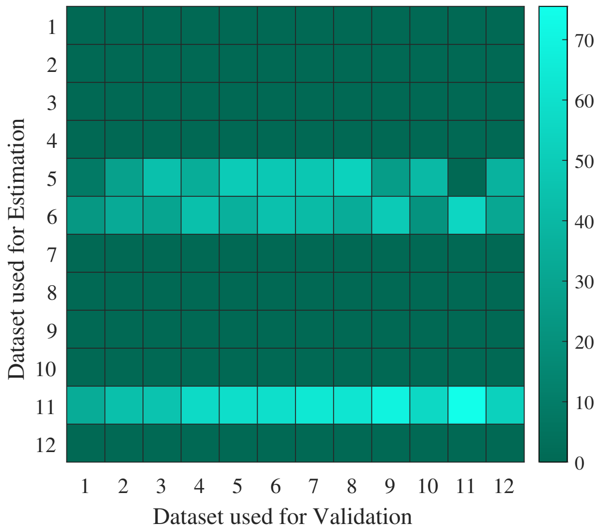


ARX2 model - fit percentage matrix obtained through SDE

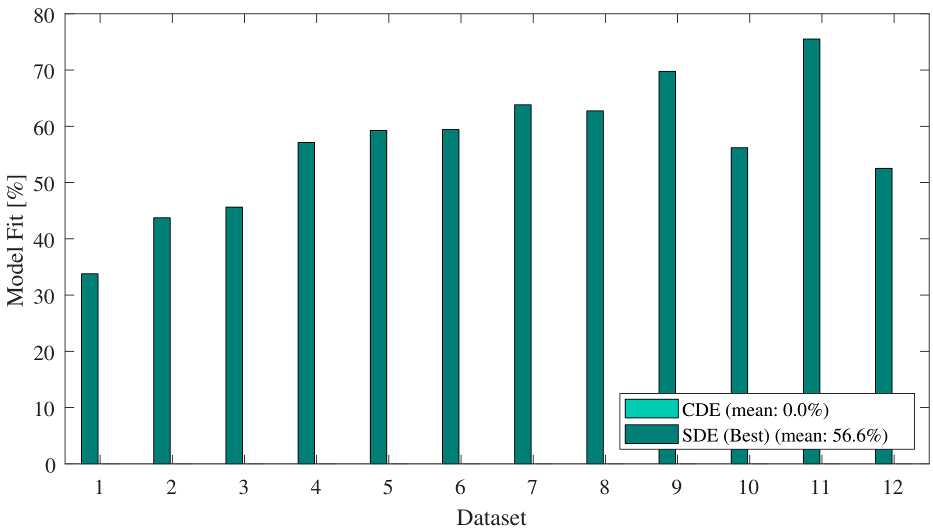


ARX2 model - fit percentage comparison between SDE and CDE

ARX3

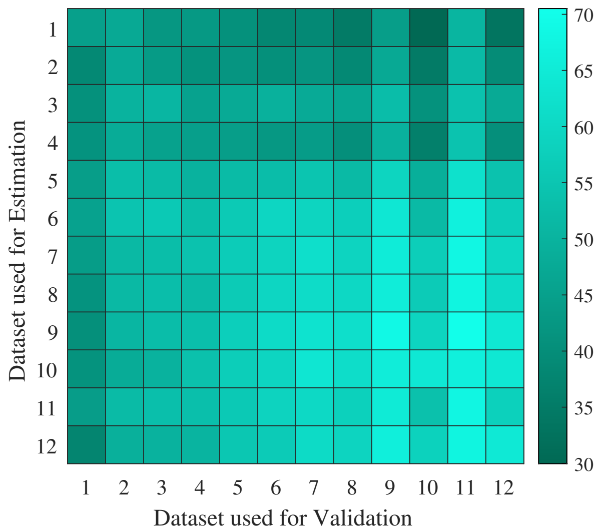


ARX3 model - fit percentage matrix obtained through SDE

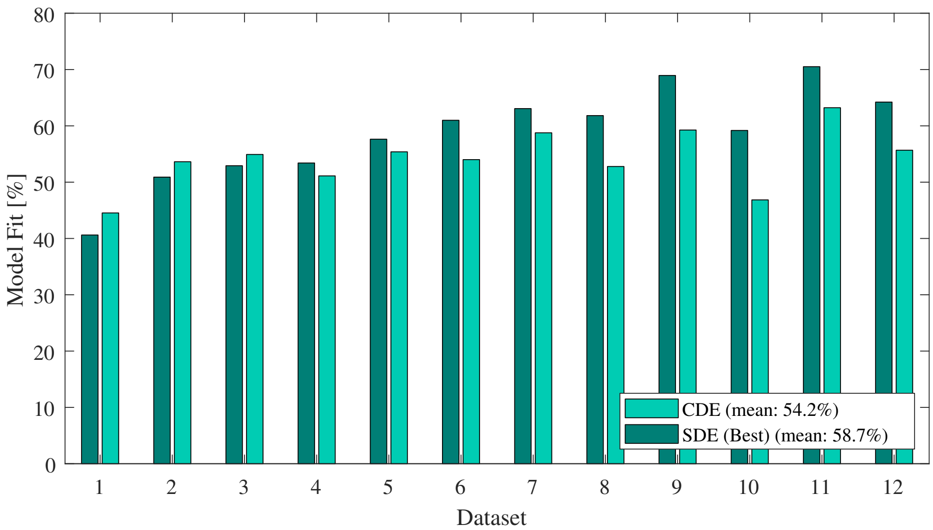


ARX3 model - fit percentage comparison between SDE and CDE

ARMAX1

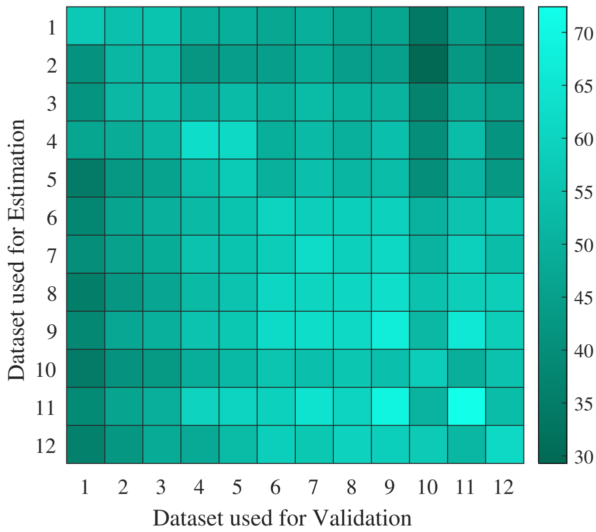


ARMAX1 model - fit percentage matrix obtained through SDE

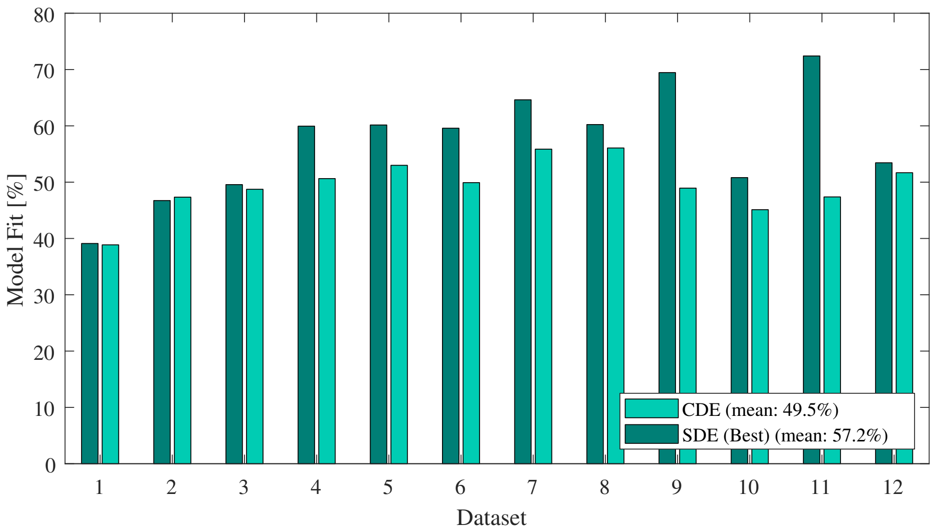


ARMAX1 model - fit percentage comparison between SDE and CDE

ARMAX2

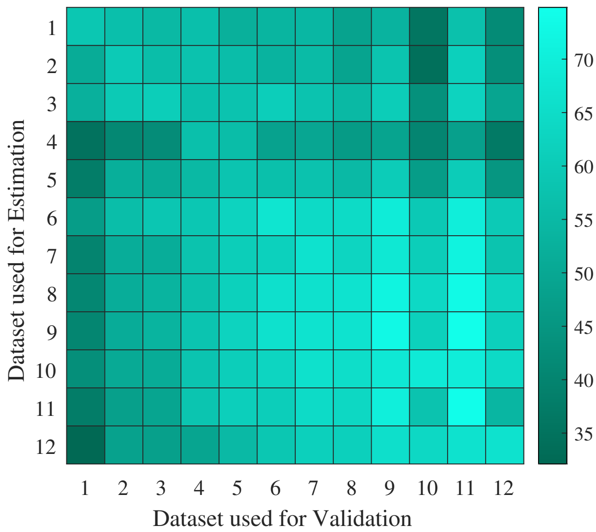


ARMAX2 model - fit percentage matrix obtained through SDE

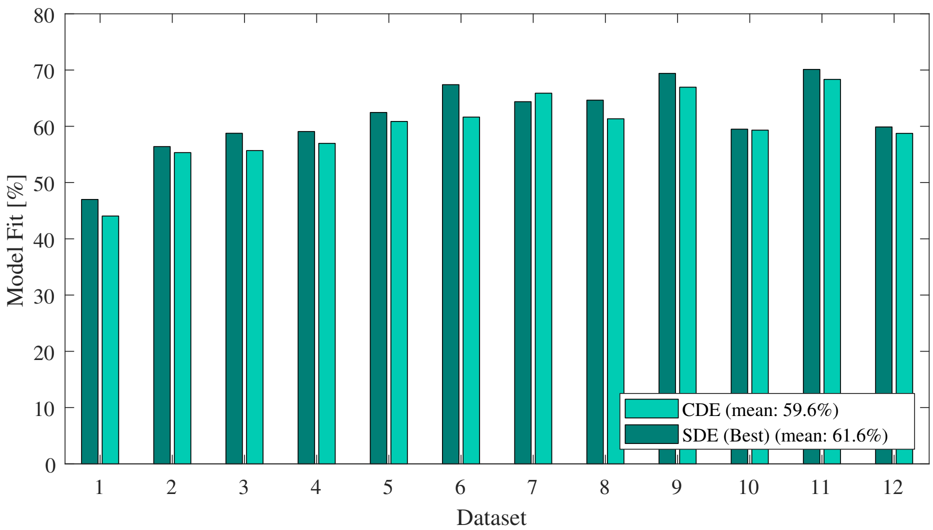


ARMAX2 model - fit percentage comparison between SDE and CDE

ARMAX3

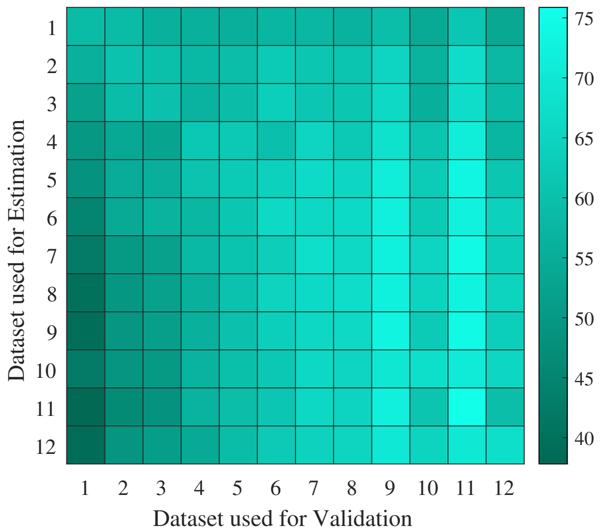


ARMAX3 model - fit percentage matrix obtained through SDE

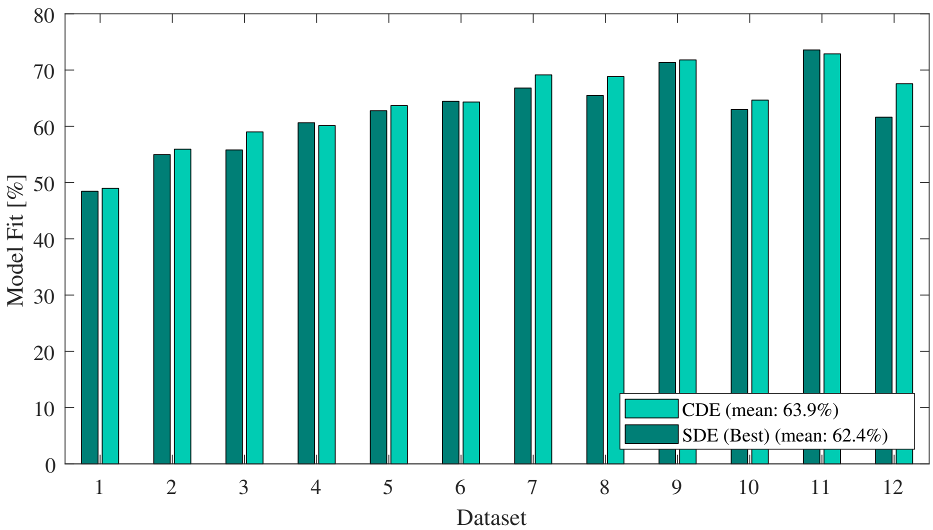


ARMAX3 model - fit percentage comparison between SDE and CDE

BJ1

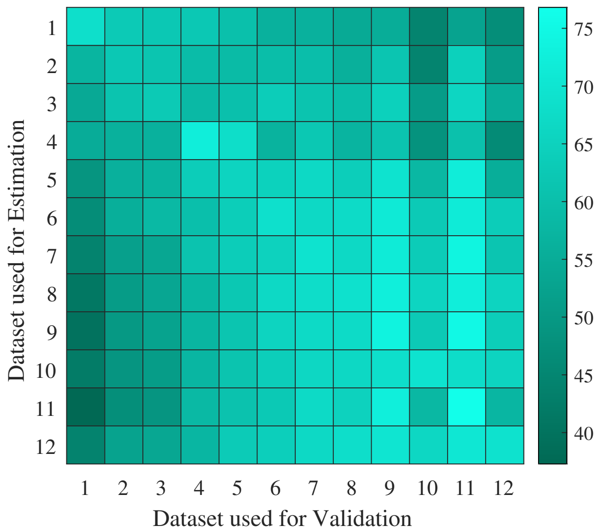


BJ1 model - fit percentage matrix obtained through SDE

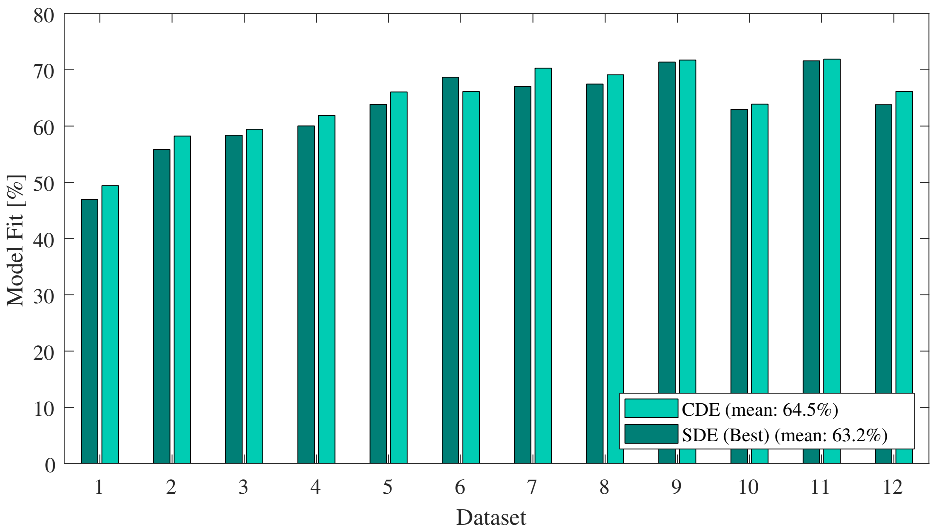


BJ1 model - fit percentage comparison between SDE and CDE

BJ2

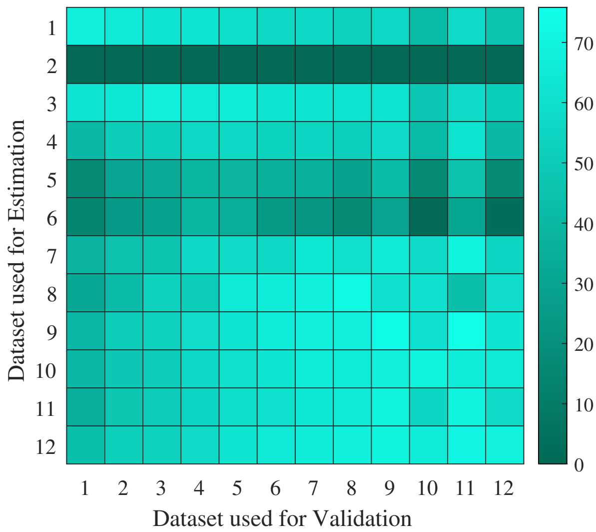


BJ2 model - fit percentage matrix obtained through SDE

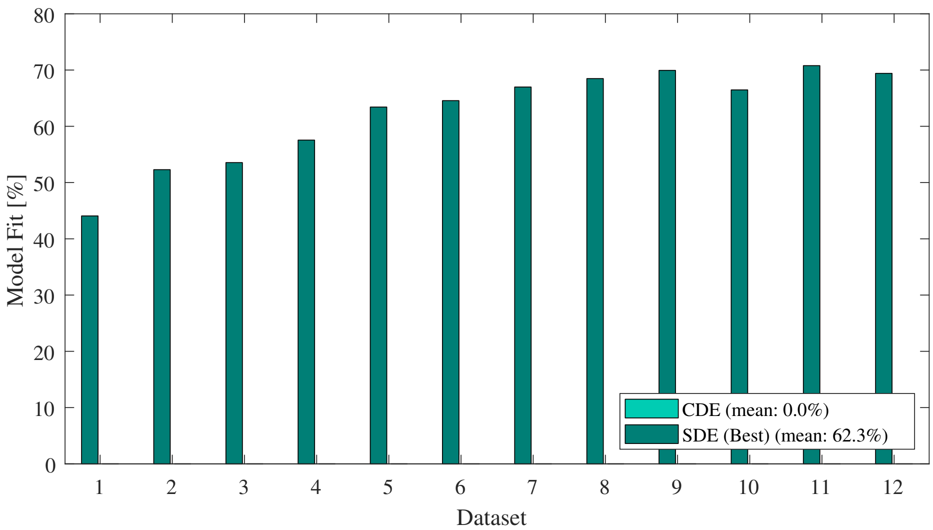


BJ2 model - fit percentage comparison between SDE and CDE

BJ3



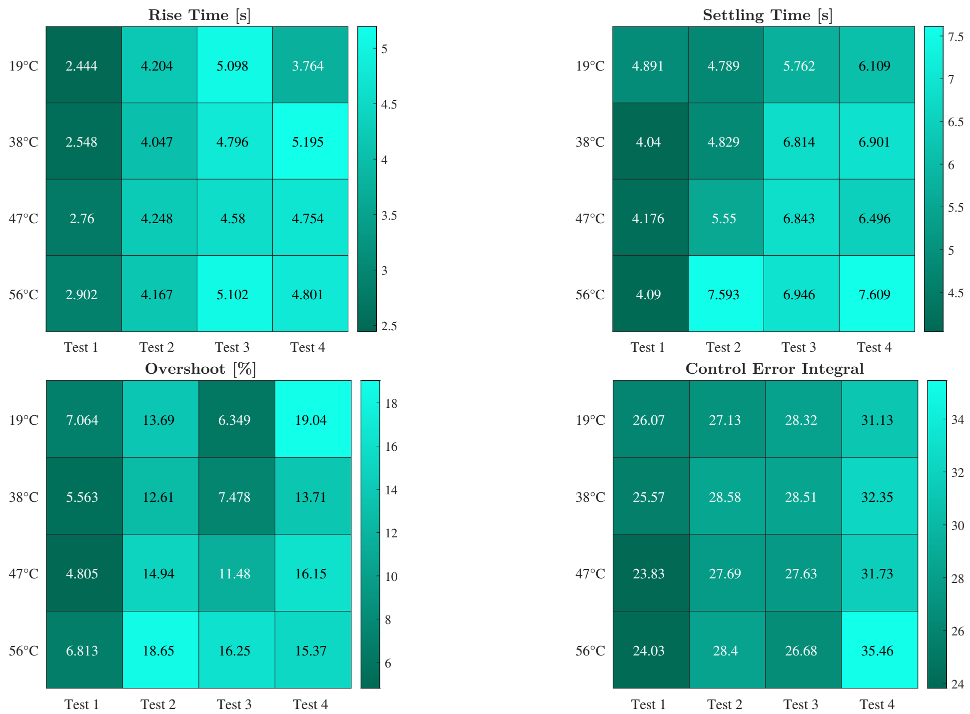
BJ3 model - fit percentage matrix obtained through SDE



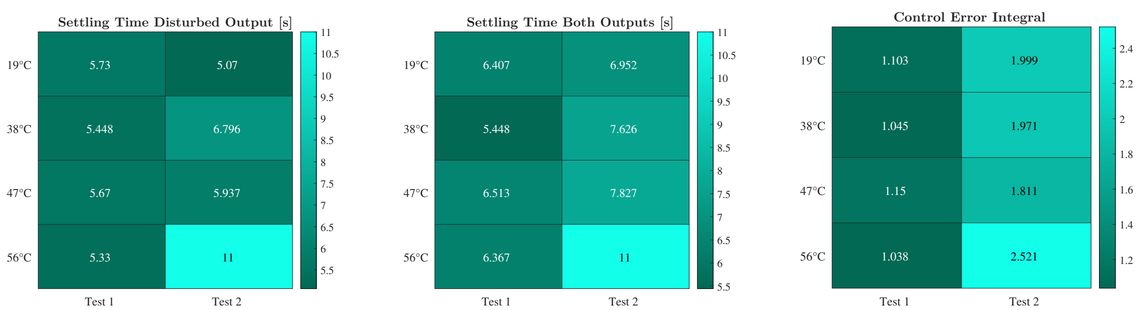
BJ3 model - fit percentage comparison between SDE and CDE

B. Additional Controller Evaluation Plots

PI Controller

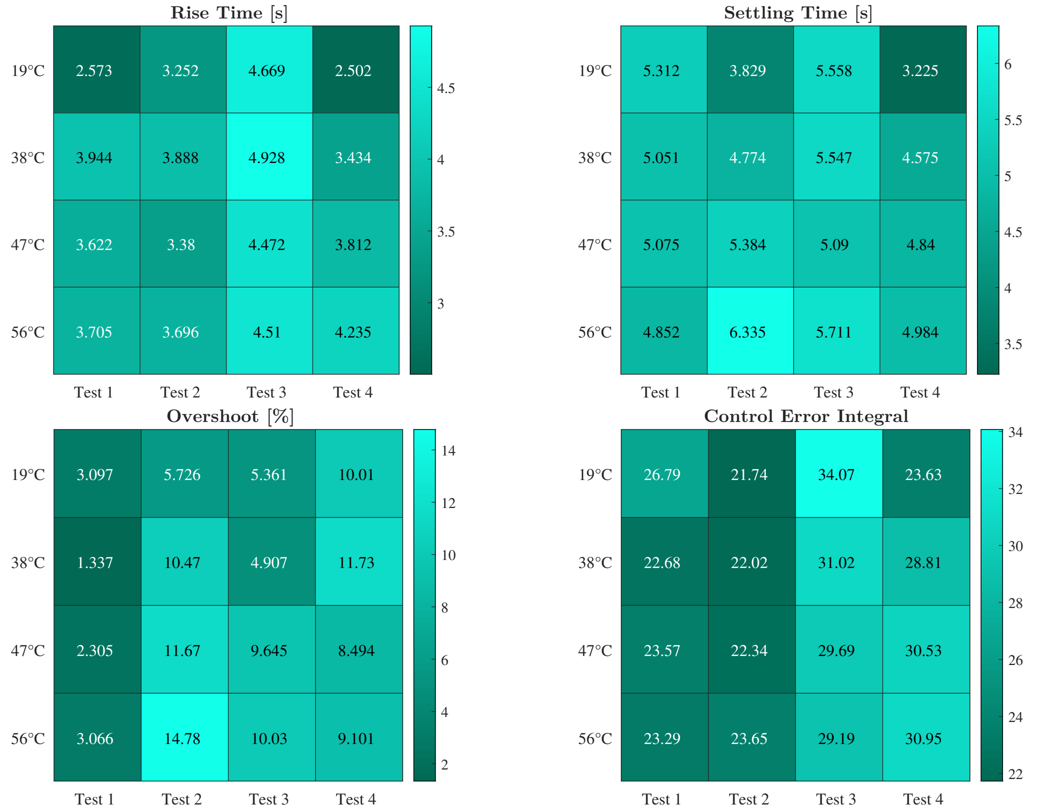


Reference Tracking Tests

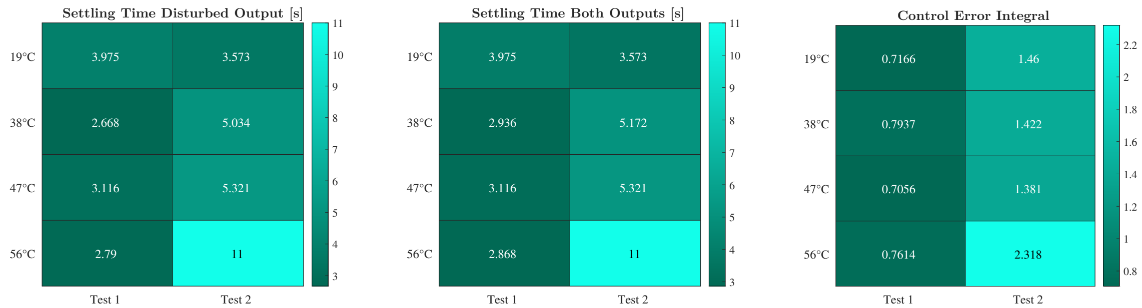


Disturbance Rejection Tests

LQI Controller

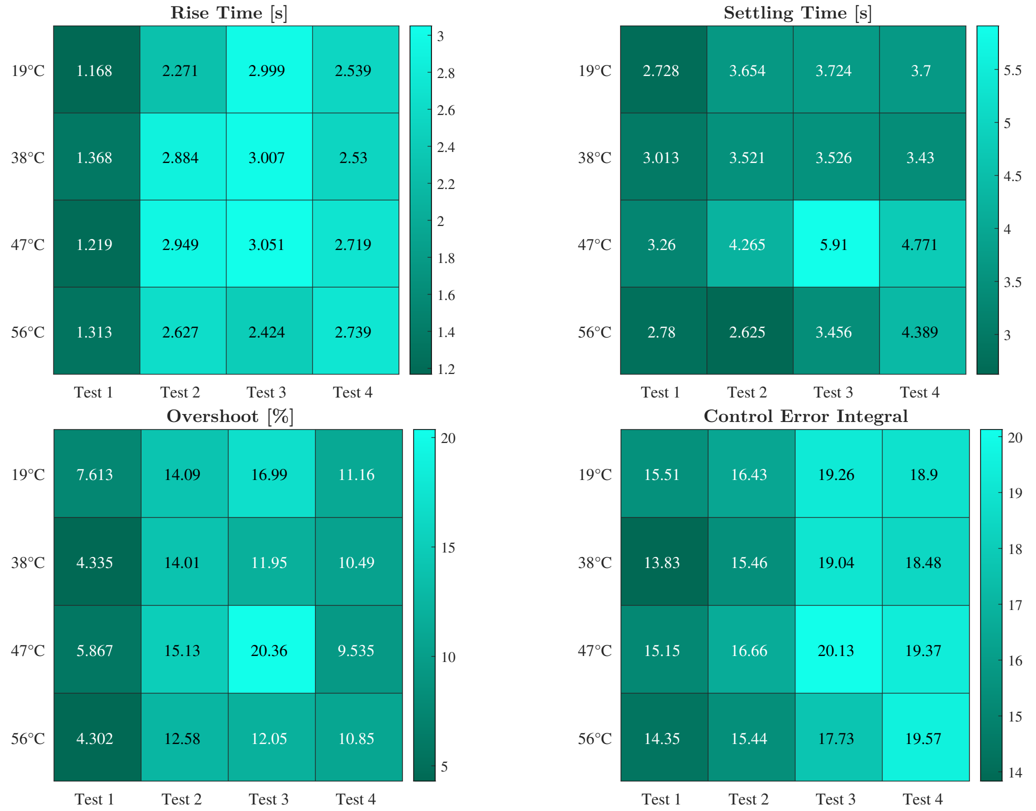


Reference Tracking Tests

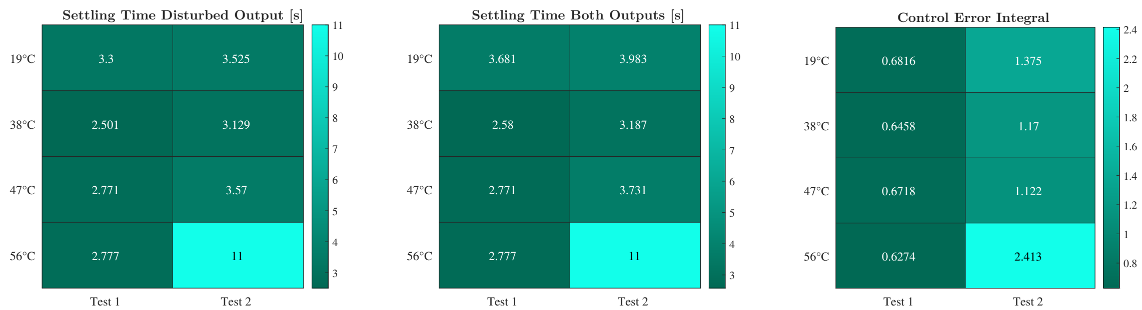


Disturbance Rejection Tests

LQId Controller

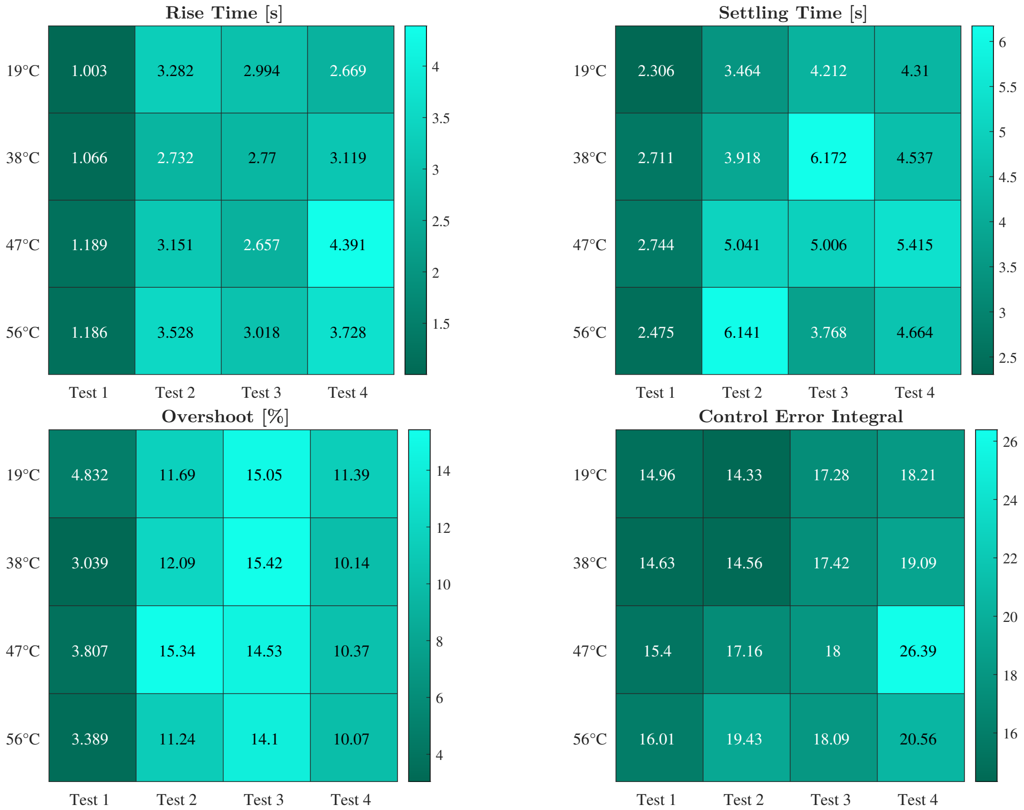


Reference Tracking Tests

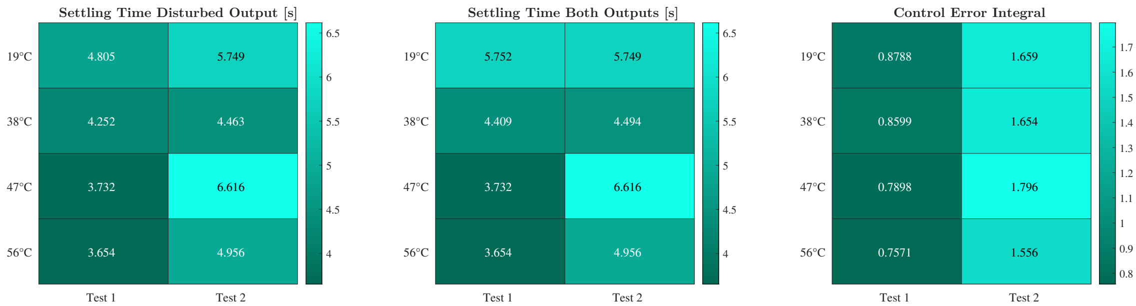


Disturbance Rejection Tests

LQIf Controller

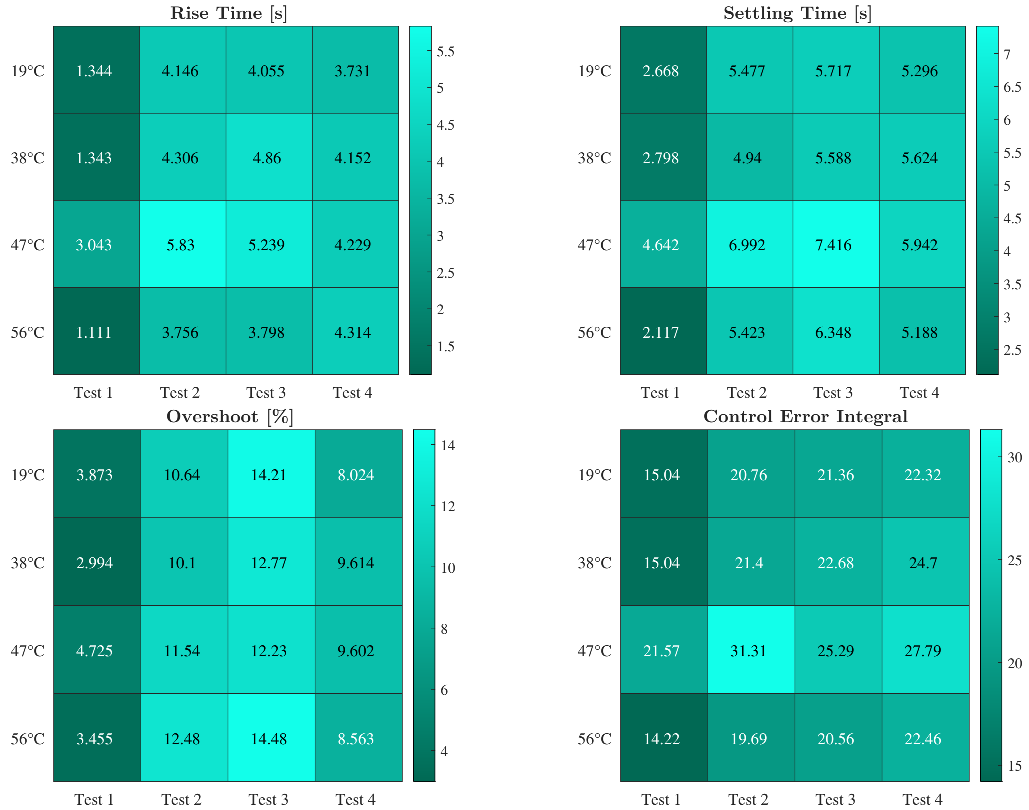


Reference Tracking Tests

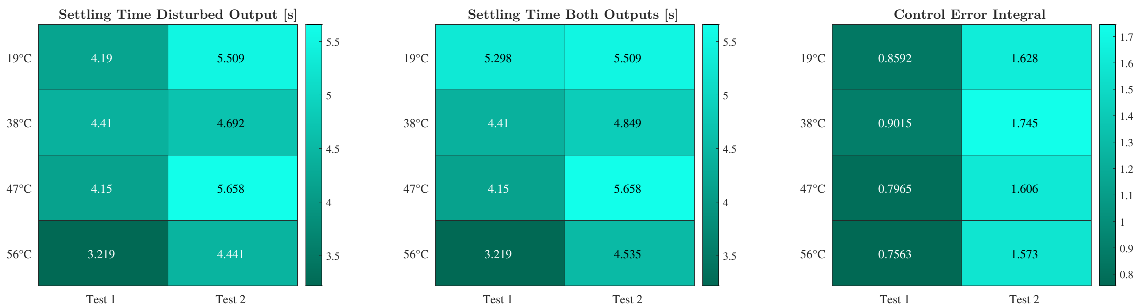


Disturbance Rejection Tests

LQIfd Controller



Reference Tracking Tests



Disturbance Rejection Tests

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se