Experimental Open-Loop Evaluation of some Time-Delay Estimation Methods in the Laguerre Domain

Svante Björklund

Control & Communication
Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
WWW: http://www.control.isy.liu.se
E-mail: svabj@isy.liu.se

26th October 2003

Report no.: LiTH-ISY-R-2537

Technical reports from the Control & Communication group in Linköping are available at http://www.control.isy.liu.se/publications.
In this report we study estimation of time-delays in linear dynamical systems with additive noise. Estimating time-delays is a common engineering problem, e.g. in automatic control, system identification and signal processing.

The purpose with this work is to test and evaluate a certain class of methods for time-delay estimation, especially with automatic control applications in mind. The class of methods consists of estimating the time-delay from the Laguerre transform of the input and output signals. The methods are evaluated experimentally with the aid of simulations and plots of approximation error, plots of original and Laguerre approximated input and output signals, plots of estimates, plots of RMS error, tables of ANOVA and plots of confidence intervals for different cases.

The results are: Only certain input signals, e.g. steps, are useful. Systems with a not too fast dynamics give better estimation quality than pure time-delay systems despite the fact that the estimation methods were derived for pure time-delay systems. The Laguerre pole should be chosen in a certain way. The number of Laguerre functions should be as high as possible.

time-delay, dead-time, estimation, system identification, linear dynamic systems, Laguerre, ANOVA, confidence intervals, simulations, open loop
# Contents

1 Introduction 1

2 The time-delay estimation methods 2
   2.1 The continuous-time Laguerre domain 2
   2.2 System representation in the continuous-time Laguerre domain 3
   2.3 A continuous-time Laguerre domain time-delay estimation algorithm 5
   2.4 The discrete-time Laguerre domain 5
   2.5 System representation in the discrete-time Laguerre domain 6
   2.6 Two discrete-time Laguerre domain time-delay estimation algorithms 7
      2.6.1 Algorithms 7
      2.6.2 Implementation 8

3 Simulation setup and analysis 10
   3.1 Simulation setup 10
   3.2 Analysis methods 13

4 Simple experiments 15
   4.1 Approximating the signals with the Laguerre transform 15
      4.1.1 Choice of the pole 15
      4.1.2 Approximation quality 18
   4.2 Estimating the time-delay 24
   4.3 Execution time 24

5 Statistical analysis 25
   5.1 Narrowband signal and pure time-delay system 25
   5.2 A standard benchmark 26
   5.3 All methods with only steps 28
   5.4 The method fischer8 with step input signals 29
      5.4.1 Both high and low SNR 30
      5.4.2 High SNR 31
      5.4.3 Low SNR 33
      5.4.4 Discussion 35
      5.4.5 Conclusions 35
## Discussion and conclusions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Discussion</td>
<td>36</td>
</tr>
<tr>
<td>6.2 Conclusions</td>
<td>37</td>
</tr>
<tr>
<td>6.3 Future work</td>
<td>38</td>
</tr>
</tbody>
</table>

## References

<table>
<thead>
<tr>
<th>References</th>
<th>1</th>
</tr>
</thead>
</table>

## Appendix A: Validation of ANOVA and confidence intervals

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1 All methods with only steps</td>
<td>1</td>
</tr>
<tr>
<td>A.2 The method <code>fischer8</code> with step input signals at high SNR</td>
<td>3</td>
</tr>
<tr>
<td>A.3 The method <code>fischer8</code> with step input signals at low SNR</td>
<td>5</td>
</tr>
</tbody>
</table>
1 Introduction

The problem we address in this report is estimating *time-delays* in linear dynamical systems with additive noise. A synonym for time delay is *dead-time*. Estimating time-delays is a common engineering problem, e.g. in control performance monitoring of industrial processes [Hor00, Swa99], in design and tuning of controllers [LMS94, AH95], in range estimation in radar [KQ92] and in direction estimation by time-delay of arrival in signal intelligence [HR97, FHJ02]. Dead-time estimation is also a necessary part in all system identification [Lju99].

The purpose with this work is to test and evaluate a certain class of methods for time-delay estimation, especially with automatic control applications in mind. The class of methods consists of estimating the time-delay from the Laguerre transform of the input and output signals [Fis99, FM99a, FM99b, FM99c]. The methods are tested and evaluated experimentally with the aid of simulations.

In the next chapter, the time-delay estimation methods are briefly described. Then, Chapter 3 is about the simulation setup and analysis tools. Chapter 4 mainly treats how well signals can be approximated in the Laguerre domain. After that, in Chapter 5 the analysis of the simulations is conducted. Following, Chapter 6 contains discussion, conclusions and suggestions for further work. Appendix A contains validation of required prerequisites for the analysis.
2 The time-delay estimation methods

In this chapter we will describe the methods studied in this report.

2.1 The continuous-time Laguerre domain

The functions

\[ L_k(s) = \frac{\sqrt{2} \beta}{s + \beta} \left( \frac{s - \beta}{s + \beta} \right)^k \]  

(2.1)

with \( \beta > 0 \) and \( k \geq 0 \) are an orthonormal basis in \( \mathcal{H}_2(\mathbb{C}_+) \), where \( \mathcal{H}_2(\mathbb{C}_+) \) is the Hardy space of all analytic functions in the right hand complex plane \( (\mathbb{C}_+) \) which are square integrable on the imaginary axis with scalar product

\[ \langle V, W \rangle = \frac{1}{2\pi i} \oint_{\Gamma} V(s)W(-s)ds \]  

(2.2)

where \( \Gamma \) is the boundary of the left hand plane [FM99a, p. 456].

The time-domain continuous-time Laguerre functions are

\[ l_k = \mathcal{L}^{-1}\{L_k(s)\} \]  

(2.3)

where \( \mathcal{L}^{-1}\{\cdot\} \) is the inverse Laplace transform. They are orthogonal in \( L_2(0, 1) \) and form a complete set in \( L_2(0, 1) \) and \( L_1(0, 1) \) [Wah91, p. 552]. The space \( L_2[a, b] \) is the space which consists of all square integrable and Lebesgue measurable functions defined on an interval \([a, b]\) with the inner product defined as

\[ \langle f, g \rangle = \int_a^b f(t)^* g(t)dt \]  

(2.4)

for \( f, g \in L_2[a, b] \) [ZD98, p. 46-47].

Example 1.

The first six continuous-time Laguerre functions are depicted in figure 2.1.
A continuous-time signal $w(t)$ can now be represented by

$$w(t) = \sum_{k=0}^{\infty} w_k l_k(t), \quad (2.5)$$

where the Laguerre coefficients $w_k$ are given by

$$w_k = \langle w(t), l_k(t) \rangle. \quad (2.6)$$

This means that the coefficients $w_k$ describe (linearly) independent properties of the signal $w(t)$ and by summing an infinite number of weighted Laguerre functions all signals in $L_2(0, \infty)$ can be described exactly.

### 2.2 System representation in the continuous-time Laguerre domain

For the signal model

$$y(t) = u(t - T_d) + n(t)$$

in the continuous-time Laguerre domain the relation between the input, output and noise signals is [Fis99, FM99c]

$$y_k = \sum_{l=0}^{k} \phi_{k,l} + n_k, \quad (2.7)$$
where \( y_k, u_k \) and \( n_k \) are the Laguerre coefficients for the output, input and noise signals, respectively. Thus, the relation between the input and the output in the Laguerre domain can be seen as a time-variant linear filter. The time-variant impulse response \( \phi_{k,l} \) is given by [Fis99, FM99c]

\[
\phi_{k,l} = \langle GL_l, L_k \rangle,
\]

where \( G = G(s) = 1 \cdot e^{-sT_d} \).

The relation in equation (2.7) can be rewritten as a linear regression [Fis99, FM99c, FM99a]

\[
y_k = \varphi_k^T \Theta + n_k,
\]

with the regression vector \( \varphi_k = [\varphi_{k,1}, \ldots, \varphi_{k,N+1}]^T \), whose elements are

\[
\begin{align*}
\varphi_{k,1} &= u_0 \\
\varphi_{k,l+1} &= \frac{(-2)^l}{l!(l-1)!} \sum_{m=0}^{k-l} \frac{(k-m-1)!}{(k-m-1)!} u_m, & k \geq l > 0 \\
\varphi_{k,l+1} &= 0 & N \geq l > k.
\end{align*}
\]

\( N + 1 \) is the total number of Laguerre coefficients. The parameter vector is

\[
\Theta = \begin{bmatrix} 1 & \beta T_d & \ldots & (\beta T_d)^N \end{bmatrix} e^{-\beta T_d}.
\]

Here it is seen where the time-delay \( T_d \) comes in.

To estimate the time-delay is more complicated here in the Laguerre domain than in the time and frequency domains. In [Fis99, FM99c, FM99a] it is done in the following way: Let

\[
Y = [y_0, \ldots, y_N]^T
\]

and

\[
\Phi = [\varphi_0, \ldots, \varphi_N]^T.
\]

Since \( \Phi \) is a square nonsingular matrix, \( \Theta \) could be estimated by

\[
\hat{\Theta} = \Phi^{-1} Y.
\]

Now the time delay can be estimated by

\[
\hat{T}_d = \beta^{-1}(\hat{\Theta}^T \hat{\Theta})^{-1} \hat{\Theta}^T \overline{\Theta},
\]

where \( \hat{\Theta} = \hat{\Theta}(1 : N) \) and \( \overline{\Theta} = \hat{\Theta}(2 : N + 1) \). Here Matlab style indexing of the elements of a vector has been used [Mat98].

In [Fis99, FM99c, FM99a] only the case \( G(s) = G_1(s) \cdot e^{-sT_d} \) with \( G_1(s) = 1 \) is dealt with. For a more general case, e.g. \( G_1(s) \) a rational function, an other relationship than equations (2.7)-2.11 between the input and output Laguerre coefficients should be derived. However, see the conclusions in Section 6.2.
2.3 A continuous-time Laguerre domain time-delay estimation algorithm

In [Fis99, FM99c, FM99a] an algorithm for the implementation of time-delay estimation based on the equations (2.14)-(2.15) in the continuous-time Laguerre domain is given. It uses discrete-time data but assumes that the sampling interval is “small”. It can estimate time-delays that contains fractions of the sampling interval. It has some features, compared to equations (2.14)-(2.15) to enhance the numerical properties.

2.4 The discrete-time Laguerre domain

We now will present the corresponding theory as in Section 2.1 for discrete-time signals.

The functions

\[ L_k(z) = \frac{z^{1-\alpha^2}}{z-\alpha} \left( \frac{1-\alpha z}{z-\alpha} \right)^k \]  

with \(-1 < \alpha < 1\) and \(k \geq 0\) are an orthonormal basis in \(H_2(D^c)\), where \(H_2(D^c)\) is the Hardy space of all analytic functions in the complement of the unit disc \((D^c)\) which are square integrable on the unit circle with scalar product

\[ \langle V, W \rangle = \frac{1}{2\pi i} \oint _{\Gamma} V(z)W^*(z) \frac{dz}{z}, \]  

where \(\Gamma\) is the unit circle [FM99b].

The time-domain continuous-time Laguerre functions are

\[ l_k = Z^{-1}\{L_k(z)\} \]  

where \(Z^{-1}\{\cdot\}\) is the inverse Z transform. They are orthonormal in \(l_2(0, \infty)\). The space \(l_2[a, b]\) is the space which consists of all square sumable discrete-time functions defined on an interval \([a, b]\) with the inner product defined as

\[ \langle f, g \rangle = \sum_{t=0}^{\infty} f(t)g(t) \]  

for \(f, g \in l_2[a, b]\).

---

Example 2.

The first four discrete-time Laguerre functions are depicted in figure 2.2.
A discrete-time signal $w(t)$ can now be represented by the same formulas (2.5)-(2.6) as for continuous-time signals but with the continuous-time Laguerre coefficient replaced by the discrete-time ones. Since all real signals have finite extent in the time, they belong to $l_2$ (for discrete-time) and can therefore be represented exactly by an infinite sum as in equations (2.5)-(2.6).

### 2.5 System representation in the discrete-time Laguerre domain

For the discrete-time signal model

$$y(t) = u(t - d) + n(t),$$

where $t$ and $d$ only can be integer valued [Fis99], the relation between the input, output and noise signals in the discrete-time Laguerre domain is according to the same formulas (2.7)-(2.9) [Fis99, FM99c] as in Section 2.2 for the continuous-time case but with $G = G(z) = z^{-d}$ and different expressions for the regression and parameter vectors [Fis99, FM99c,
The regression vector is
\[
\begin{align*}
\varphi_{k,1} &= u_0 \\
\varphi_{k,l+1} &= \frac{(1-\alpha^2)^l}{l!(l-1)!} \sum_{m=0}^{k-l} (-1)^{k+l-m} \alpha^{k-l-m} \frac{(k-m-l)!}{(k-m-l)!} u_m, & k \geq l > 0 \\
\varphi_{k,1+l} &= 0 & N \geq l > k
\end{align*}
\] (2.20)

The parameter vector is [FM99b]
\[
\Theta = \left[ 1, \quad d, \quad \ldots, \quad d(d-1) \ldots (d-(N-1)) \right]^T a^d. 
\] (2.21)

Here again it is seen where the time-delay \( T_d \) comes in.

In [Fis99, FM99c] a method to estimate the time-delay is suggested:
\[
\hat{d} = (1^T 1)^{-1} 1^T Y_d = \frac{1}{N} 1^T \hat{D} + \frac{N-1}{2}, 
\] (2.22)

where \( 1 = [1, \ldots, 1]^T, \ Y_d = \hat{D} + [0, 1, \ldots, N-1]^T \) and \( \hat{D} = \text{diag} \left( \hat{\Theta} \right)^\dagger \hat{\Theta} \). The symbol \( \dagger \) means pseudoinverse. The estimate \( \hat{\Theta} \) of the parameter vector is computed in the same way as for the continuous-time case (equation (2.14)).

In [FM99b] an other method to estimate the time-delay is suggested. See also the next section.

### 2.6 Two discrete-time Laguerre domain time-delay estimation algorithms

#### 2.6.1 Algorithms

In [Fis99, FM99c, FM99b] two algorithms for the implementation of time-delay estimation based on the system representation (2.9), (2.20)-(2.21) in the discrete-time Laguerre domain are given. One of them is using equation (2.22). They also have some features to enhance the numerical properties.

The matrix \( \Phi \) will not be well-conditioned for large \( N \) [Fis99]. The numerical properties of \( \Phi \) is improved by using SVD (singular value decomposition [HJ91]) to extract the most significant part of \( \Phi \). See [Fis99, FM99a, FM99b] for details. The number of singular values (the largest ones) to retain is a user-defined parameter. It is in this report set to 5, which is chosen after some simple tests.

The time-delay \( d \) can be estimated in two ways, here called \texttt{tausvd} and \texttt{taulp1}. In \texttt{tausvd}, \( d \) is estimated by least squares as in equation (2.22). In \texttt{taulp1} the \( \Theta \) estimate (equation (2.14)) is further improved by an iterative algorithm called MIRLS (Modified Iteratively Reweighted Least Squares) [FM99b, CCS94] before estimating the time-delay by equation (2.22). See Algorithms 1-2.
Algorithm 1 \texttt{tausvd}

1. Optimize the pole $\alpha$ given the input and output signals: “minimizing the squared equation error between the input signal and its approximation by a truncated ($N = 16$) Laguerre series with respect to $\alpha$” [Fis99, FM99b, FM99c].

2. Compute the regression matrix $\Phi$ and the Laguerre spectrum $Y$ of the output signal from the input and output signals.

3. Retain the most significant part $\Phi_s$ of the regression matrix $\Phi$ by SVD

$$\Phi = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

$$\Phi_s = U_1 \Sigma_1 V_1^T.$$ 

The diagonal matrix $\Sigma_1$ has the largest singular values on the diagonal. The number of used singular values in $\Sigma_1$ is a user-selected parameter.

4. Estimate the parameter vector

$$\hat{\Theta} = \Theta_s^T Y_s,$$

where $Y_s = [Y^T(1 : s), 0, \ldots, 0]^T$.

5. Estimate the time-delay

$$\hat{d} = (1^T 1)^\dagger 1^T Y_d$$

Algorithm 2 \texttt{taulp1}

1. Estimate the parameter vector $\Theta$ as in step 1-4 in algorithm 1.

2. Use the estimated parameter vector $\hat{\Theta}$ as start value to the MIRLS [FM99b, CCS94] to improve the estimate of the parameter vector.

3. Estimate the time delay as in step 5 in algorithm 1.

2.6.2 Implementation

The algorithms \texttt{tausvd} and \texttt{taulp1} (Algorithm 1 and 2) were implemented in eight versions in MATLAB. Table 2.1 contains the used parameters of the implementations. For the implementations \texttt{fischer1}-\texttt{fischer4} the pole was chosen according to Section 4.1.1 to $\alpha = 0.6$ or $\alpha = 0.8$. For the implementations \texttt{fischer5}-\texttt{fischer8} the pole position 0.995 was selected according to [Fis99, FM99b, FM99c]: “by minimizing the squared equation error between the input signal and its approximation by a truncated ($N = 16$) Laguerre series with respect to $\alpha$”.
<table>
<thead>
<tr>
<th>Name</th>
<th>$N + 1$</th>
<th>Algorithm</th>
<th>Pole $\alpha$</th>
<th># sing. vals</th>
</tr>
</thead>
<tbody>
<tr>
<td>fischer1</td>
<td>51</td>
<td>tausvd</td>
<td>0.8</td>
<td>5</td>
</tr>
<tr>
<td>fischer2</td>
<td>51</td>
<td>taulp1</td>
<td>0.8</td>
<td>5</td>
</tr>
<tr>
<td>fischer3</td>
<td>150</td>
<td>tausvd</td>
<td>0.6</td>
<td>5</td>
</tr>
<tr>
<td>fischer4</td>
<td>150</td>
<td>taulp1</td>
<td>0.6</td>
<td>5</td>
</tr>
<tr>
<td>fischer5</td>
<td>51</td>
<td>tausvd</td>
<td>0.955</td>
<td>5</td>
</tr>
<tr>
<td>fischer6</td>
<td>51</td>
<td>taulp1</td>
<td>0.955</td>
<td>5</td>
</tr>
<tr>
<td>fischer7</td>
<td>150</td>
<td>tausvd</td>
<td>0.955</td>
<td>5</td>
</tr>
<tr>
<td>fischer8</td>
<td>150</td>
<td>taulp1</td>
<td>0.955</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 2.1. Parameters of the implemented algorithms. The number of used Laguerre functions is $N + 1$. “# sing. vals” means the number of retained singular values in Algorithm 1.
3 Simulation setup and analysis

3.1 Simulation setup

In order to assess the performance of the time-delay estimation methods in Section 2.6.2 some simulations were conducted in MATLAB. The setup for the simulations was the same as in [Bjo03b] with the following exceptions:

- The signal-to-noise ratio (SNR) was defined in a different way. In [Bjo03b] it was defined as the power of the input signal divided by the noise at the output of the system. In this report, the SNR is defined as the power of the output signal without the noise divided by the power of the noise. The relation between the SNR “at the output” and the SNR “at the input” is not a constant but depends on the system and the input signal. In [Bjo03b] it was said and showed with a graph that this relation varied only between one and two, i.e. with a factor of two between different combinations of input signal types and systems. That graph and statement was incorrect. Later it has been shown that the two definitions of SNR differed with up to a factor of up to 54 (t158a.m). This implies that the result should depend on the definition of the SNR.

- Two extra input signals were used in some of the simulations. They both were sinusoids, windowed with a triangle window:
  \[ u_i(k) = \sin(\omega_i k T_s) \cdot \text{triang}(k T_s) \]
  but with different frequencies \( \omega_i \). The signal \( \sin1 \) has the frequency \( \omega_1 = 0.5 \) and the signal \( \sin2 \) the frequency \( \omega_1 = 0.1 \). The quantity \( T_s \) is the sampling interval. See Figures 3.2-3.3. These signals were used because they are similar to the signals in the suggested application of the studied time-delay estimation methods [Fis99, FM99a, FM99b, FM99c].

- One extra system was used: the continuous-time system consisting of a pure time-delay of 9 sampling intervals (before the sampling): \( G_9(s) = e^{-9T_s s} \). Here \( T_s \) is the sampling interval. The time-delay after zoh (zero order hold) sampling will be 10 sampling intervals, like for the other systems [Bjo03b].

- One extra system was used: the continuous-time system consisting of a pure time-delay of 9 sampling intervals (before the sampling): \( G_9(s) = e^{-9T_s s} \). Here \( T_s \) is the sampling interval. The time-delay after zoh (zero order hold) sampling will be 10 sampling intervals, like for the other systems [Bjo03b].

- The numbers of trials in the statistical analyses were different. See Chapter 5.

- The methods were of course different. See Chapter 2.

Three environment factors were varied during the simulations: The system, the input signal type and the SNR [Bjo03b]. The signal-to-noise ratio (SNR) was either 1 or 100. The impulse responses of four of the used systems are depicted in Figure 3.1. Note that for all these systems the time delay will be 10 after the sampling. More information about these systems can be found in [Bjo03b]. The fifth used system, \( G_9 \), is described above.
Figures 3.2-3.6 show the used input signals in the time and frequency domains. Figure 3.4 depicts the signal RBS 10-30% (a bandpass random signal with frequency contents between 10% and 30% of the Nyquist frequency). Figure 3.5 depicts the signal RBS 0-100% (a wideband random signal with frequency contents between 0% and 100% of the Nyquist frequency). It is approximately white noise. Figure 3.6 depicts the signal Steps (a signal with three steps). It has a frequency contents between 0% and about 5% of the Nyquist frequency. More information about these input signals can be found in [Bjo03b].

Figure 3.1. Impulse response of system $G_1$-$G_2$ and $G_5$-$G_6$. True time-delay after sampling $T_d = 10$.

Figure 3.2. Time signal (left) and frequency spectrum (with estimated uncertainty of 2 standard deviations) (right) for a realization of the input signal type $\sin 1$. 
Figure 3.3. Time signal (left) and frequency spectrum (with estimated uncertainty of 2 standard deviations) (right) for a realization of the input signal type $\sin^2$.

Figure 3.4. Time signal (left) and frequency spectrum (with estimated uncertainty of 2 standard deviations) (right) for a realization of the input signal type RBS 10-30%.

Figure 3.5. Time signal (left) and frequency spectrum (with estimated uncertainty of 2 standard deviations) (right) for a realization of the input signal type RBS 0-100%.
3.2 Analysis methods

Several analysis methods for the evaluation of the quality of the time-delay estimates have been used:

1. Plots of the approximation error for different Laguerre poles $\alpha$ when approximating the input and output signals with Laguerre functions. See Section 4.1.1. Using these plots we can choose an optimal $\alpha$.

2. Plot of original and Laguerre approximated input and output signal for different number of used Laguerre functions, $N + 1$. See Section 4.1.2. With these plots we can get an idea of how good the approximation will be and how it depends on $N + 1$.

3. Plot of the estimates gives the possibility to discover outliers and behavior for different factor level combinations.

4. Bar plots of the RMS (root mean square) error of the estimates for different factor level combinations give a common estimation quality measure which can be used to determine if the estimates are good enough. They also indicate which factor level combinations, e.g. methods, are better or worse than others.

5. ANOVA (ANalysis Of VAriance) [Mon97, Mat01] on the (transformed [Mon97]) RMS errors for different factor level combinations. ANOVA can tell us if there is a statistically significant difference between factor level combinations, e.g. different methods or systems.

6. Plot of confidence intervals for pair-wise comparisons [Mon97, Mat01] for the (transformed [Mon97]) RMS errors for different factor level combinations. This can give us the possibility to say that certain factor level combinations, e.g. some methods, are better than others.

Before using the the analysis methods 4-6 above a special treatment of outliers and failed estimates have been used. As in [BL03] estimates outside the range (0, 20) (including the
boundaries) have been set to the value 20 and then only the 95% best estimates have been retained. The motivation is that a good implementation of the estimation method can detect time-delay estimates outside an expected range (here (0,20)) and should be able to handle this, e.g. by restarting an optimization with a different initial value.
4 Simple experiments

4.1 Approximating the signals with the Laguerre transform

A necessary condition for the time-delay estimation methods described in Chapter 2 to be successful is that the input and output signals can be represented accurately in the Laguerre domain. In this section we will look at how well this can be done.

4.1.1 Choice of the pole

If we could use an infinite number of Laguerre basis functions as in equation (2.5), all real signals which have finite extent in the time domain could be represented exactly, independent of the choice of the pole $\alpha$. In reality we cannot use an infinite number of Laguerre basis functions. Therefore the quality of the representation in the Laguerre domain depends on the choice of $\alpha$. In Figures 4.1-4.2 the power $J_{\text{tot}}(\alpha)$ of the error between the real and the approximated input and output signals is depicted for $N + 1 = 51$ used Laguerre functions. The power $J_{\text{tot}}(\alpha)$ is defined by

$$J_{\text{tot}}(\alpha) = \sum_{t=1}^{M} (u(t) - \hat{u}(t))^2 + \sum_{t=1}^{M} (y(t) - \hat{y}(t))^2,$$

where $M$ is the number of time samples and $\hat{u}(t)$ and $\hat{y}(t)$ are the Laguerre approximations of the input and the output signals. The output signal included noise. The SNR was 100 (t184b3func.m). We see from Figures 4.1-4.2 that

- The system has no large influence on the quality of the approximation. Compare with Section 5.4.
- The input signal has a large influence.
- At their respective optima, the input signal sin2 can be best approximated, Steps second best, sin1 third best. The two RBS signals are the most difficult to approximate. For sin2 and Steps at SNR=100 this is in accordance with Figures 5.2 and 5.5.
- $J_{\text{tot}}(\alpha)$ is fluctuating for the two RBS signals but rather stable with a clear minimum for the other signals.
- The lower the frequency of the input signal, the higher is the optimum $\alpha$. For the white signal there is no clear minimum but $J_{\text{tot}}(\alpha)$ is rather flat in the interior of $[-1, 1]$.
- $J_{\text{tot}}(\alpha)$ varies most for the signal steps between its minimum and maximum values ($\approx 0.6$). It varies least for sin1.
- A pole position around 0.8 is acceptable for all input signal types when $N + 1 = 51$. 

15
Figure 4.1. The power $J_{\text{tot}}(\alpha)$ of the error between the real and the approximated input and output signals. $N + 1 = 51$. Signal $\sin 1$ (left) and $\sin 2$ (right). Three different line styles for three different systems.

Figure 4.2. The power $J_{\text{tot}}(\alpha)$ of the error between the real and the approximated input and output signals. $N + 1 = 51$. Signal RBS 10-30% (upper left), RBS 0-100% (upper right) and Steps (lower left). Three different line styles for three different systems.
If we use more Laguerre function than in Figures 4.1-4.2, the approximation for all signals will be better. This can be seen in Figures 4.3-4.4, where $N + 1 = 150$. It is most important for the two RBS signals.

By comparing Figures 4.1-4.2 with Figures 4.3-4.4 we notice that

- The approximation quality will be much better for the signal $\sin 1$ with $N + 1 = 150$. It will be somewhat better for the RBS signals and $\sin 2$ and Steps.
- The minimum is broader for $\sin 2$ and Steps with $N + 1 = 150$ than with $N + 1 = 51$.
- A pole position around 0.6 appear to be a good choice for all input signal types when $N + 1 = 150$.

**Figure 4.3.** The power $J_{tot}(\alpha)$ of the error between the real and the approximated input and output signals. $N + 1 = 150$. Signal $\sin 1$ (left) and $\sin 2$ (right). Three different line styles for three different systems.
Figure 4.4. The power $J_{\text{tot}}(\alpha)$ of the error between the real and the approximated input and output signals. $N + 1 = 150$. Signal RBS 10-30% (upper left), RBS 0-100% (upper right) and Steps (lower left). Three different line styles for three different systems.

When performing some simple tests with $\alpha = 0.955$ (suggested in [Fis99, FM99b, FM99c]), they gave better results (better time-delay estimates) than using $\alpha = 0.6$ or $\alpha = 0.8$. The reason is unclear. See also Sections 2.6.2 and 6.1.

4.1.2 Approximation quality

Figures 4.5-4.9 display the original and the Laguerre approximated input and output signals for different input signal types. These figures are in agreement with Figures 4.1-4.4. The Laguerre approximation behaves similar to a low-pass filtering but it works best at the beginning of the signals. We see that:

- The signals $\text{sin2}$ and Steps are easy to approximate, $\text{sin1}$ is hard and the RBS signals very hard.
- The approximations become better with more Laguerre coefficients.
Figure 4.5. Original and Laguerre approximated input and output signal for $\sin 1$ with $N + 1 = 51$ (above) and $N + 1 = 150$ (below). $\alpha = 0.8$, $G_2$, SNR=100.
Figure 4.6. Original and Laguerre approximated input and output signal for \( \sin^2 \) with \( N + 1 = 51 \) (above) and \( N + 1 = 150 \) (below). \( \alpha = 0.8, G_2, \text{SNR}=100. \)
Figure 4.7. Original and Laguerre approximated input and output signal for RBS 10-30% with $N + 1 = 51$ (above) and $N + 1 = 150$ (below). $\alpha = 0.8$, $G_2$, SNR=100.
Figure 4.8. Original and Laguerre approximated input and output signal for RBS 0-100% with $N + 1 = 51$ (above) and $N + 1 = 150$ (below). $\alpha = 0.8$, $G_2$, SNR=100.
Figure 4.9. Original and Laguerre approximated input and output signal for Stepswith $N+1 = 51$ (above) and $N + 1 = 150$ (below). $\alpha = 0.8$, $G_2$, SNR=100.
4.2 Estimating the time-delay

When executing some simulations and estimations “by hand” we noticed:

- The methods are non-robust. They can for example give negative estimates or very large positive estimates.
- Using $\alpha = 0.8$ appears to give time-delay estimates with an incorrect sign but $\alpha = 0.955$ with a correct sign. The reason for this behavior is unknown.
- By some simulations that used random systems it could be seen that fischer8 can not handle all types of systems. For example it can not handle systems with negative gain or integrating systems.
- It seems to be more difficult to estimate the time-delay of the systems $G_2$ and $G_9$ than the systems $G_1$, $G_5$ and $G_6$ by the method fischer8. This is surprising since the methods are derived for a pure time-delay like $G_9$.
- For the system $G_1$, steps input and fischer8 a high SNR gives better estimates than a low SNR.
- For the system $G_9$, steps input and fischer8 a high SNR appears to give a lower variance in the estimates but also a larger bias than with a low SNR.
- The result depends much on how many singular values are retained.

4.3 Execution time

The execution time of the method fischer8 is high. Table 4.1 shows the execution time for the methods fischer8 and arxstruc3 [Bjo03e] in a simple test in MATLAB. The execution speed of fischer8 can probably be optimized further. The most execution time is required for the calculation of the regression matrix $\Phi$ (Equations 2.13 and 2.20). In our test the calculation of $\Phi$ took 82% of the total execution time (=the first column in Table 4.1). The execution time for $\Phi$ increases rapidly with increasing number of used Laguerre function $N + 1$. Fortunately, $\Phi$ depends only on the input signal and not on the output signal. This means that if we repeatedly use the same input signal, $\Phi$ needs to be computed only once. In the second column in Table 4.1 the already computed $\Phi$ is used. Some of the differences in execution time between the first and the second column in Table 4.1 is probably because MATLAB does not load a function into memory before it is needed. Asumably, the whole difference for arxstruc3 between the first and later calls is due to this loading of functions into memory. The loading has a very small effect on the execution time for fischer8. Comparison of the execution time of more methods are reported in [Bjo03a].

<table>
<thead>
<tr>
<th>Method</th>
<th>First call</th>
<th>Later calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>fischer8</td>
<td>63.994</td>
<td>2.882</td>
</tr>
<tr>
<td>arxstruc3</td>
<td>0.202</td>
<td>0.118</td>
</tr>
</tbody>
</table>

Table 4.1. Execution time for fischer8 in seconds on a SUN SunBlade 100 computer. The column “Later calls” shows the mean execution time in calls 2-5. (t227b1.m)
5 Statistical analysis

In this chapter we compare the tested methods and see in which cases they could be used. We will perform a statistical analysis from which it is possible to draw conclusions. Bear in mind that all results are in average over all the factors that are not explicitly studied. Certain special cases can give a different result.

5.1 Narrowband signal and pure time-delay system

We start by using the input signal type $\sin 2$, which is a typical signal in the intended application of the time-delay estimation methods [Fis99, FM99a, FM99b, FM99c]. This signal was also easy to approximate by Laguerre functions according to Sections 4.1.1 and 4.1.2. We also use a system for which the methods were derived, i.e. a pure time-delay. This should be ideal circumstances for the methods with very good performance. The number of trials was 1024 for each factor level combination.

Figure 5.1 shows the time-delay estimates. As can be seen there are some outliers. Such outliers can easily be detected and handled. Therefore they have be removed from the following analysis (See Section 3.2). Figure 5.2 shows the RMS error for different methods and SNR. No method succeed when the SNR is low despite the fact that the signal should be appropriate and the system is of the correct type. The methods $\text{fischer6}$ and $\text{fischer8}$ are the best methods for high SNR with RMS error 0.60 and 0.47 sampling intervals, respectively.

![Figure 5.1](t223b1.m)  
Figure 5.1. Top graph: Time-delay estimates for different methods and SNR when the continuous-time system is a pure time-delay of 9 samples and the input signal is a low-pass triangle windowed sinusoid ($\sin 2$, Section 3.1 and Figure 3.3). The vertical axis of the top graph should be scaled by $10^4$. (t223b1.m)
Figure 5.2. RMS error (in number of sampling intervals) for different methods and SNR when the system is a pure time-delay of 9 samples and the input signal is a low-pass triangle windowed sinusoid ($\sin^2$, Section 3.1 and Figure 3.3). 1024 trials. (t223b.m)

5.2 A standard benchmark

In this section we will use a simulation setup that has been used in several earlier reports [Bjö02, Bjö03b, Bjö03c, Bjö03d, Bjö03e] (with the exceptions listed in Section 3.1). Figure 5.3 shows the time-delay estimates. As can be seen there are some outliers. These outliers are removed by the procedure in Section 3.2. From Figures 5.4-5.5 we draw the conclusions:

- Step input signals are possible to use but RBS signals are not (Figures 5.4-5.5).
- For step input signals, the method fischer8 seems to be the best (Figures 5.4-5.5).
- For step signal input, fischer8 has an RMS error of 1.69 and 0.81 sampling intervals for low (SNR=1) and high SNR (SNR=100), respectively (Figure 5.5).
Figure 5.3. Top graph: Time-delay estimates for different methods, input signal types, SNR and systems for the standard benchmark. The vertical axis of the top graph should be scaled by $10^5$. The horizontal axes of both graphs should be scaled by $10^5$. (t224b1.m)

t224b1: rootmean, data(:,:,m,m,m,m,m,m)

Figure 5.4. Average RMS error (in number of sampling intervals) for different methods and input signal types for the standard benchmark. (t224b1)
Figure 5.5. Average RMS error (in number of sampling intervals) for different methods, input signal type and SNR for the standard benchmark. (t224b1)

5.3 All methods with only steps

We saw in Section 5.2 that the tested Laguerre domain methods worked best with step input signals. Therefore, in this section we restrict ourselves to such signals and compare the different methods with the aid of confidence intervals for pair-wise comparisons.

We use the estimates for the low SNR (SNR=1). The number of trials was 1152. The trials were split into four groups of 288 trials each and each group was used to compute an estimate of the RMS error (our response variable [Mon97]) of the time-delay estimate. This gave 4 estimates of the RMS error that was used in the ANOVA and the calculation of the confidence intervals in Figure 5.6. The ANOVA table (Table 5.1) tells us that there are significant differences between combinations of methods and systems (the column “Prob>F” contains very small values). The confidence interval plot shows that fischer8 indeed is significantly better than the other methods for step inputs. Appendix A.1 contains validation graphs for the use of ANOVA and the confidence interval plot when SNR=1.

When using high SNR (SNR=100) the prerequisites for the ANOVA and the confidence interval plot were not fulfilled (determined by the same type of graphs as in Appendix A.1). Since the case with low SNR is a more difficult case than with high SNR, we consider it to be more informative. It must be more difficult for a method to be best for low SNR. It should then not be bad for high SNR either.
Table 5.1. Analysis of Variance table for all methods with input signal Steps. SNR = 1. Constrained (Type III) sums of squares [Mat01]. Transformation: (RMS error)$^{1.1153}$. ($t224b1.m$)

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum Sq.</th>
<th>d.f.</th>
<th>Mean Sq.</th>
<th>F</th>
<th>Prob&gt;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>2039.4036</td>
<td>7</td>
<td>291.3434</td>
<td>13441.2379</td>
<td>0</td>
</tr>
<tr>
<td>Sys</td>
<td>64.5481</td>
<td>3</td>
<td>21.516</td>
<td>992.6497</td>
<td>0</td>
</tr>
<tr>
<td>Method*Sys</td>
<td>84.2187</td>
<td>21</td>
<td>4.0104</td>
<td>185.0219</td>
<td>0</td>
</tr>
<tr>
<td>Error</td>
<td>2.0808</td>
<td>96</td>
<td>0.021675</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>2190.2511</td>
<td>127</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.6. Confidence intervals (95% simultaneous confidence level) comparing all methods with step input at SNR = 1. Transformation: (RMS error)$^{1.1153}$. Tukey’s honestly significant difference criterion [Mat01] is used. ($t224b1.m$)

5.4 The method fischer8 with step input signals

We have seen that the estimation methods in the Laguerre domain work best for steps as input signals. We have also seen that fischer8 is the best of the tested methods. Therefore we concentrate on fischer8 and steps in this section. Since the methods which we study in this report were derived for systems being a pure time-delay, an interesting question is: Is the method (fischer8) only suitable for pure time-delay systems or can it be used also for systems with dynamics? We will investigate this in this section. We will mostly treat high and low SNR separately.
5.4.1 Both high and low SNR

Figure 5.7 shows the time-delay estimates. The outlier with the value of over 800 would destroy the ANOVA and confidence interval analysis (Section 3.2). It has therefore been removed according to the procedure in Section 3.2.

Figure 5.8 displays the average (over SNR) of the RMS error for different systems for fischer8 and Steps. The systems with slow dynamics ($G_1$, $G_5$ and $G_6$) seem to be the easiest and the pure time delay ($G_9$) the hardest. Figure 5.9 shows the RMS error for high and low SNR, separately. The difference between the systems is more clear for the low SNR. Strangely enough, for some systems ($G_2$ and $G_9$) the RMS error seems to be larger for high SNR than for low SNR.

![Figure 5.7. Top graph: Response variable for different systems for fischer8 with input signal Steps. Both high and low SNR. Four systems. (t225b1)](image-url)
To easier fulfill the requirements of the subsequent ANOVA and confidence interval plots the following two sections treats the cases with high (100) and low (1) SNR separately.

5.4.2 High SNR

Figure 5.10 depicts the RMS estimation error for 16 groups of time-delay estimates for each factor level combination (= each system) for high SNR. The RMS values seem to be clearly separated for two of the systems. The spread for the each system is small. The maximum RMS value is 10 also here because of the treatment in Section 3.2.
The number of trials was 4096. The trials were split into four groups of 1024 trials each and each group was used to compute an estimate of the RMS error (our response variable [Mon97]) of the time-delay estimate. This gave 4 estimates of the RMS error that was used in the calculation of the confidence intervals.

The ANOVA table in Table 5.2 simply says that there are differences between the systems. Figure 5.11 shows confidence intervals for pairwise comparisons of systems. We cannot say that there is any difference between the systems $G_1$, $G_5$ and $G_6$ but we can say that these three systems give significantly better estimation result than the others. The system $G_2$, which is faster, is not that good. The system $G_9$, i.e. the pure time delay has the worst estimation performance. It is significantly worse than the other systems. This is surprising since the method fischer8 was derived for pure time-delay systems. Appendix A.2 contains validation graphs for the ANOVA and the confidence interval plot.

Figure 5.10. Top graph: RMS error for different systems for fischer8 with input signal Stepsat SNR = 100. The RMS error is plotted in the order of the system: $G_1$, $G_2$, $G_5$, $G_6$ and $G_9$ according to the stairstep graph (bottom graph).

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum Sq.</th>
<th>d.f.</th>
<th>Mean Sq.</th>
<th>F</th>
<th>Prob&gt;F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sys</td>
<td>70.33</td>
<td>4</td>
<td>17.5825</td>
<td>543040.8413</td>
<td>0</td>
</tr>
<tr>
<td>Error</td>
<td>0.00048567</td>
<td>15</td>
<td>3.2378e-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70.3305</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2. Analysis of Variance table for fischer8 with input signal Steps. SNR = 100. Constrained (Type III) sums of squares [Mat01]. Transformation: (RMS error)$^{2.0313}$. (t225b1.m)
Figure 5.11. Confidence intervals (95% simultaneous confidence level) for different systems for fischer8 with input signal Steps at SNR = 100. Transformation: \((\text{RMS error})^{2.0313}\). Tukey’s honestly significant difference criterion [Mat01] is used. (t225b1.m)

5.4.3 Low SNR

Figure 5.12 depicts the RMS estimation error for 16 groups of time-delay estimates for each factor level combination for low SNR. The RMS values seem to be clearly separated for two of the systems. The spread for the each system larger than for the high SNR in Section 5.4.2. The maximum RMS value is again 10 because of the treatment in Section 3.2.

The number of trials was 4096. The trials were split into four groups of 1024 trials each and each group was used to compute an estimate of the RMS error (our response variable) of the time-delay estimate. This gave 4 estimates of the RMS error that was used in the calculation of the confidence intervals.

The ANOVA table in Table 5.3 says that there are differences between the systems. Figure 5.13 shows confidence intervals for pairwise comparisons of systems. As for the high SNR in Section 5.4.2, we cannot say say that there is any difference between the systems \(G_1, G_5\) and \(G_6\) but these three systems give significantly better estimation result than the others. The system \(G_2\), which is faster, is not that good. The system \(G_9\), i.e. the pure time delay has the worst estimation performance. It is significantly worse than the other systems. Appendix A.3 contains validation graphs for the ANOVA and the confidence interval plot.
Figure 5.12. Top graph: RMS error for different systems for fischer8 with input signal Steps at SNR = 1. The RMS error is plotted in the order of the system: G1, G2, G5, G6 and G9 according to the stairstep graph (bottom graph).

Table 5.3. Analysis of Variance table for fischer8 with input signal Steps. SNR = 1. Constrained (Type III) sums of squares [Mat01]. Transformation: (RMS error)$^{4.4792}$. (t225b1.m)
Figure 5.13. Confidence intervals (95% simultaneous confidence level) for different systems for fischer8 with input signal Stepsat SNR = 1. Transformation: (RMS error)$^4$. Tukey’s honestly significant difference criterion [Mat01] is used. (t225b1.m)

5.4.4 Discussion

Even if we only have studied the method fischer8 in Sections 5.4.1-5.4.3, it is reasonable to believe that the same results should apply also for the other Laguerre domain methods (Section 2.6.2), i.e. a pure time-delay system is not the easiest.

Also the simple experiments reported in Section 4.2 indicate that a pure time-delay system is not the easiest.

It is unclear why low SNR seems to give a lower bias than high SNR for the system $G_9$ with fischer8 and step input (Figures 5.9, 5.10 and 5.12).

5.4.5 Conclusions

From the results in Section 4.1.1 and 5.4.1-5.4.3 we draw the astonishing conclusion:

- The best estimation result with the tested methods is not achieved with a pure time-delay but with more complicated systems with a not too fast dynamics.

This is despite the fact that the methods were derived for pure time-delay systems.
6 Discussion and conclusions

6.1 Discussion

The pole $\alpha = 0.955$ (suggested in [Fis99, FM99b, FM99c]) worked better than $\alpha = 0.6$ or $\alpha = 0.8$ both in simple experiments and statistical analyses. It is unclear why $\alpha = 0.955$ give better performance than $\alpha = 0.6$ or $\alpha = 0.8$. In 4.1.1 the sum of the squared equation error of the input signal and of the output signal using Laguerre series of length $N+1 = 51$ and $N+1 = 150$ is used as the criterion for the choice of the Laguerre pole $\alpha$. The output signal contains noise with SNR = 100. This results in the poles $\alpha = 0.8$ or $\alpha = 0.6$ for $N+1 = 51$ and $N+1 = 150$, respectively. In [Fis99, FM99b, FM99c] the optimization criterion only models the input signal by few (16) Laguerre coefficients. The output signal or noise is no part of that criterion. Perhaps is the difference in the optimum pole position caused by that trying also to model the output signal with its noise is too demanding.

The methods in this report are different from the method Laguerre DAP [Bjö02, IHD01, Hor00]. In Laguerre DAP the impulse response of the system is modeled by Laguerre functions. In this report the input and output signals are modeled by Laguerre functions. Whether the modeling of a signal with Laguerre functions is successful depends on how similar the signal and the Laguerre functions are. Theoretically ($N \to \infty$) all finite length signals can be modeled exactly by Laguerre functions (independent of choice of the pole). But this is less interesting since we cannot use infinite sums in the reality. In order to represent a discrete-time signal with a finite length $M$ in the frequency domain it is enough with $M$ DFT (Discrete Fourier Transform) coefficients [GLM01]. For the discrete-time Laguerre domain this is probably not enough. A discrete-time signal with a finite length $M$ is seen as signal of infinite length where only $M$ signal values are different from zero. (See extended spaces in [Fis99, FM99c].) This means that we probably need many more Laguerre coefficients than $M$ to exactly represent the signal.

If, however, the truncated Laguerre functions are linear independent, it should be enough with $M$ Laguerre coefficients. Then equation 2.5 will be a nonsingular quadratic linear equation system. The discrete-time Laguerre functions in Figure 2.2 looks “independent” but can we be sure?

Modeling of impulse responses with Laguerre functions [Bjö02, IHD01, Hor00] probably is more suitable than modeling input and output signals because typical impulse responses are more “Laguerre-like” than typical input and output signals. Compare Figures 2.1-2.2 with Figures 3.1 and 3.4-3.6.

Laguerre modeling of input and output signal can work well when the signals have a short extent in time and do not oscillate too much (e.g. are low-pass). See Figures 4.5-4.9. The pole $\alpha$ must be suitably chosen (Figures 4.1-4.4) and the number $N$ of Laguerre functions must be large enough (Figures 4.1-4.9). Of our signals the windowed sinusoid with low frequency and the signal with steps work fine but the other signals not so well.

Laguerre functions are chosen in [Fis99, FM99a, FM99b, FM99c] as the basis functions probably because the signals in the intended applications are easy to represent in the Laguerre domain. Other basis functions could also be possible, e.g. Kautz functions.

Laguerre coefficients with low subscripts are suitable for low-pass signals. For other signals perhaps other ranges of subscripts should be used.
It is unclear why low SNR seems to give a lower bias than high SNR for the system $G_9$ with fischer8 and step input (Figures 5.9, 5.10 and 5.12).

**Drawbacks with the Laguerre domain methods:**

- The pole $\alpha$ must be chosen. The choice affects the performance of the time-delay estimation method. See Section 2.6.2 for how this can be done.

- For accurate approximations of signals many Laguerre functions are needed. This requires much computation time and can maybe give numerical problems.

- The continuous-time Laguerre domain time-delay estimation algorithm in Sections 2.1-2.3 requires fast sampling.

- Sometimes the estimation fails, which the simulations shows, e.g. the estimates can be negative or large positive (Figures 5.2, 5.3 and 5.7).

- Not suitable for all types of input signals.

**Advantages with the Laguerre domain methods:**

- Good for step inputs.

- Good for systems with not too fast dynamics.

- The continuous-time Laguerre domain time-delay estimation algorithm in Sections 2.1-2.3 can estimate subsample time-delay.

**6.2 Conclusions**

We draw the following conclusions from the work in this report.

- **The input signal:**
  - The input signal has a large influence on the time-delay estimation quality.
  - The Laguerre domain methods are suitable for input signals that can be well approximated by truncated series of Laguerre functions. The signals should be low-pass and not too long in time.
  - One or several steps or a triangle windowed low-frequency sinusoid are examples of suitable signals. Random binary signals are not suitable.

- **The system:**
  - The system has a smaller influence on the time-delay estimation quality than the input signal.
  - Systems with a not too fast dynamics give better time-delay estimation quality than pure time-delay systems despite the fact that the estimation methods were derived for pure time-delay systems.
• The estimation methods:
  
  – The method \texttt{fischer8} (Section 2.6.2) is the best of the tested methods. In the tests, \texttt{fischer8} gave an RMS error of 0.81 sampling intervals for SNR=100 and 1.69 for SNR=1 with an input signal consisting of three steps.
  
  – The Laguerre pole $\alpha$ should be chosen as in [Fis99, FM99a, FM99b, FM99c] (Section 2.6.2 in this report).
  
  – The number of used Laguerre functions should be as high as possible. However, a high number gives a longer execution time. The choice will therefore be a trade-off between estimation quality and execution speed.
  
  – The methods do not seem to be robust. They sometimes fail and deliver unreasonable estimates.
  
  – The execution time of the methods is high, especially if many Laguerre functions are used. The execution time can, however, be reduced for subsequent estimations if the same input signal is employed.

6.3 Future work

Some suggestions of future work:

• Compare the result for different number of used Laguerre functions.

• Investigate why $\alpha = 0.955$ is better than $\alpha = 0.8$ or $\alpha = 0.6$.

• Investigate why pure time-delay systems do not give the best time-delay estimation result.

• Investigate why low SNR appears to give a lower bias of the time-delay estimate than high SNR for pure time-delay systems with step inputs.

• Develop more robust estimation methods that can handle failures and outliers.

• Implement and test the time-delay estimation method in the continuous-time Laguerre domain referenced in Section 2.3.
References


Appendix A: Validation of ANOVA and confidence intervals

This appendix comments on the use and applicability of the ANOVA and confidence intervals in Chapter 5.

The four RMS estimates (Sections 5.3, 5.4.2 and 5.4.3) have first gone through a data dependent transformation (Figures A.1, A.4 and A.7) to make the variance more constant [Mon97, Bjo03b]. Then ANOVA [Mon97] was executed and confidence intervals for pair-wise comparisons [Mat01] plotted. For the *p-values* (the column Prob>F) in the ANOVA tables (Table 5.1-5.3) which are very small, the corresponding factors or interactions have effect with a very high confidence level and we say there are *significant differences*. If there are significant differences, confidence intervals for pair-wise comparisons can be plotted (Figures 5.6, 5.11 and 5.13). For the ANOVA and confidence intervals to be valid some prerequisites must be fulfilled [Mon97]. These are usually tested by studying some validation graphs [Mon97]. The following sections contain these graphs.

A.1 All methods with only steps

In this section we show validation graphs for the ANOVA and confidence intervals in Section 5.3. The used transformation [Mon97] is depicted in Figure A.1. We see in Figures A.2-A.3 that the prerequisites are approximately fulfilled [Mon97].

![Figure A.1. Plot (without transform) for choosing a variance-stabilizing transform [Mon97] for ANOVA for several Laguerre domain methods (Section 2.6.2) with input signal Steps. SNR = 1. Transformation: (RMS error)\(^{1.1153}\). The transformation is chosen by fitting a straight line to the data points by the least squares method.](image-url)
Figure A.2. Residual analysis for ANOVA and confidence intervals for several methods (Section 2.6.2) with input signal Steps. SNR = 1. Ideally the residuals should be Gaussian and the residuals vs. time and fitted value should be within a horizontal band and be "structureless" [Mon97].

Figure A.3. Residual standard deviation versus factor levels for ANOVA and confidence intervals for several Laguerre domain methods (Section 2.6.2) with input signal Steps. SNR = 1. Ideally the standard deviation for all factor levels should be equal [Mon97].
A.2 The method **fischer8** with step input signals at high SNR

In this section we show validation graphs for the ANOVA and confidence intervals in Section 5.4.2. The used transformation [Mon97] is depicted in Figure A.4. We see in Figures A.5-A.6 that the prerequisites are not completely fulfilled [Mon97]. The standard deviation of the residuals is not constant (Figure A.6). The results in Table 5.2 and Figure 5.11 are, however, very clear. Therefore we trust the conclusions in Section 5.4.2.

![Figure A.4](image)

**Figure A.4.** Plot (without transform) for choosing a variance-stabilizing transform [Mon97] for ANOVA for fischer8 with input signal Steps. SNR = 100. Transformation: (RMS error)$^{2.0313}$. The transformation is chosen by fitting a straight line to the data points by the least squares method.
Figure A.5. Residual analysis for ANOVA and confidence intervals for fischer8 with input signal Steps. SNR = 100. Ideally the residuals should be Gaussian and the residuals vs. time and fitted value should be within a horizontal band and be "structureless" [Mon97].

Figure A.6. Residual standard deviation versus factor levels for ANOVA and confidence intervals for fischer8 with input signal Steps. SNR = 100. Ideally the standard deviation for all factor levels should be equal [Mon97].
A.3 The method fischer8 with step input signals at low SNR

In this section we show validation graphs for the ANOVA and confidence intervals in Section 5.4.3. The used transformation [Mon97] is depicted in Figure A.7. We see in Figures A.8-A.9 that the prerequisites are approximately fulfilled [Mon97].

![Figure A.7. Plot (without transform) for choosing a variance-stabilizing transform [Mon97] for ANOVA for fischer8 with input signal Steps. SNR = 1. Transformation: (RMS error)^4.4792. The transformation is chosen by fitting a straight line to the data points by the least squares method.](image)
Figure A.8. Residual analysis for ANOVA and confidence intervals for *fischer8* with input signal Steps. SNR = 1. Ideally the residuals should be Gaussian and the residuals vs. time and fitted value should be within a horizontal band and be "structureless" [Mon97].

Figure A.9. Residual standard deviation versus factor levels for ANOVA and confidence intervals for *fischer8* with input signal Steps. SNR = 1. Ideally the standard deviation for all factor levels should be equal [Mon97].