System Identification and Estimation

Master of Science in Aerospace Engineering

Identification of Linear System from Sampled Data

Authors:

Rajashree SRIKANTH rajashree.srikanth@student.isae-supaero.fr
Nivetha SRINIVASAN nivetha.srinivasan@student.isae-supaero.fr

Tutor:
Yves BRIERE

December 8, 2023

2023/2024 – 3rd Semester

1 Introduction

This report highlights the results of the system identification done for a given linear system. The model was identified first for a discrete time sample using Recursive Least Squares and Extended Recursive Least Squares Method. The identification process was then attempted using continuous-time model, using State Variable Filter and Instrumental State Variable Filter methodologies.

For the model identification, we use numerical data, a part of which is used for the identification, while the other half is utilized for verification of the estimated model.

2 Identification of a Discrete Time Model

The identification of a discrete time model is done using the Recursive Least Squares Algorithm (RLS) and Extended Recursive Least Squares Algorithm (ELS). The key difference between the two is RLS assumes that the measurement error is modeled as white noise. On the other hand, ELS is more generic and can account for white as well as colored noise. These algorithms exploit the idea of the Least Squares algorithm and are called recursively over time. The advantage of using a recursive algorithm is that it is computationally inexpensive compared to the Ordinary Least Squares (OLS) algorithm.

2.1 Recursive Least Squares Method

2.1.1 Theory

The estimated model $\hat{\theta}$ from the RLS algorithm is

$$\hat{\theta}(t+1) = \hat{\theta}(t) + D(t)\frac{\phi(t)}{1+\phi(t)D(t)\phi(t)^T}$$

with regressor ϕ, apriori error $\epsilon^0(t)$

$$\epsilon^0(t) = y(t+1) - \hat{\theta}(t)\phi(t)$$

and $D = (\phi^T\phi)^{-1}$

$$D(t+1) = D(t) - \frac{D(t)\phi^T\phi D(t)}{1+\phi^T D(t)\phi^T}.$$

This algorithm, like OLS, aims to minimize the cost function

$$J = \sum_{i=1}^{t} \epsilon^2(t)$$

Similarly, we can also include a component λ, called forgetting factor, which uses more recent data for the minimization of the cost function and progressively ignores the later or initial error terms. Usually, this λ is taken as 0.99, and the new cost function J becomes

$$J(t) = \lambda^0\epsilon(T)^2 + \lambda^1\epsilon(T-1)^2 + ... + \lambda^T(1)^2$$

The matrix D is modified to incorporate the forgetting factor as:

$$D(t+1)^{-1} = \lambda D(t)^{-1} + \phi(t)\phi(t)^T$$

$$D(t+1) = \left[D(t) - \frac{D(t)\phi^T\phi D(t)}{\lambda + \phi^T D(t)\phi^T}\right]\frac{1}{\lambda}.$$

The fit of the data is obtained through regression, and the accuracy of the fit is determined by the R^2 factor.

2.1.2 Finding n_a and n_b

The template given for this lab includes a data set that is given as input to the algorithm for identification. As mentioned previously, the first half of the data is used for identification. The signal input here given here is a PRBS signal over $120s$.

The algorithm results in a model that is generated by first establishing the number of poles and number of zeros that best fits the data given.

A loop was run with n_a (number of poles) and n_b (number of zeros) with the values varying from 1 to 10 and the RLS %fit for all iterations of these values are obtained. From the plots 1 and 2, it can be seen that n_a, n_b values between 4 and 8 for na and nb was found to give the low values of the cost function which would give the least error. We choose $n_a = 6$ and $n_b = 4$ for the choice of our discrete transfer function, since it gives an appreciably high %fit value, while maintaining low model complexity.

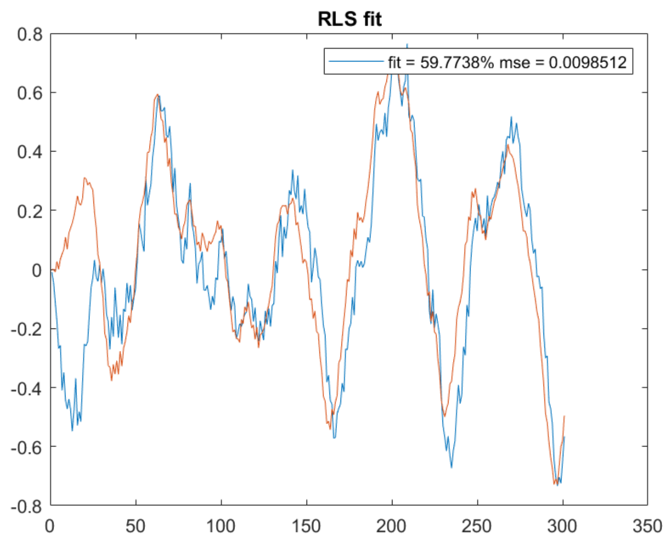

Figure 1: RLS fit

To find the best na and nb values from the range concluded above, another factor to be considered is the fit of the model. A higher value of na and nb, although produced lesser error, the fit of the model was compromised. Thus, through validation a balance was struck between precision and order of model. From this, by trial and error, the $na = 6$ and $nb = 4$ values were chosen.

The final transfer function identified using the RLS algorithm is:

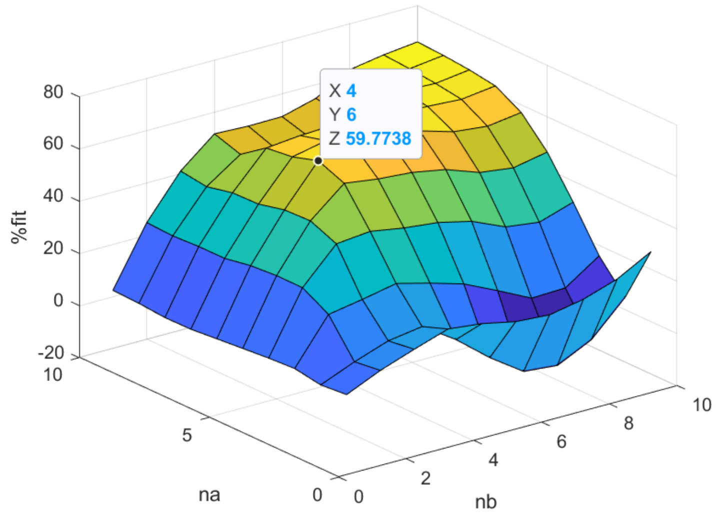

Figure 2: Plot of the %fit for different na and nb values

$$F_{ls} = \frac{-008803z^{-2} + 0.02376z^{-3} + 0.02004z^{-4} + 0.0271z^{-5}}{1 - 0.6451z^{-1} - 0.496z^{-2} - 0.1405z^{-3} - 0.01478z^{-4} + 0.2001z^{-5} + 0.1354z^{-6}} \tag{1}$$

2.1.3 Incorporating forgetting factor

Including the forgetting factor (λ) in the Recursive Least Squares Algorithm, to reduce the weight of the past observations in the updating process, the model is generated again. The value for λ is taken to be 0.99. With this the updated values of na and nb are found to be 6 and 4 respectively as before. The plot for the same is given below in figure 3.

2.1.4 Validation

The second half of the data was used to validate the generated model using a step response. These are represented in the plots 4 and 5.

2.2 Recursive Extended Least Squares Method

2.2.1 Theory

The algorithm for ELS resembles very closely to RLS, with the incorporation of an additional coefficient c, with the model represented as

$$y(t) = [-y(t-1)\, u(t-1)][a\, b]^T + \epsilon(t) + c\epsilon(t-1)$$

$$Ay = Bu + c\epsilon$$

The magnitude of the error

$$\epsilon = \sqrt{\frac{e^T e}{N}}$$

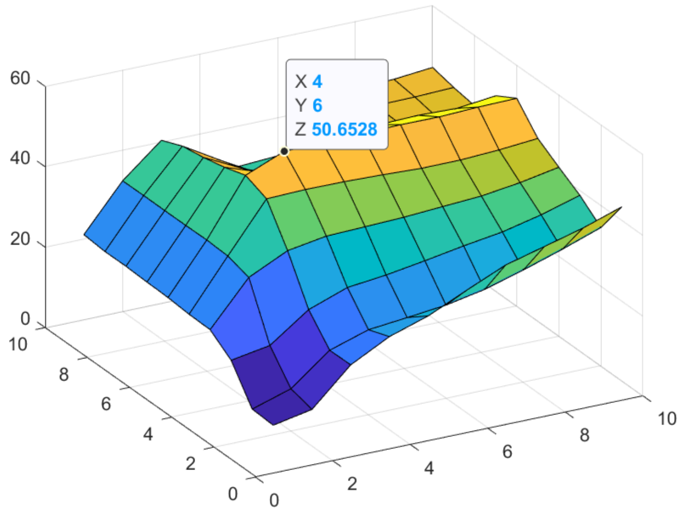

Figure 3: Plot of the %fit for different na and nb values (on incorporating the forgetting factor)

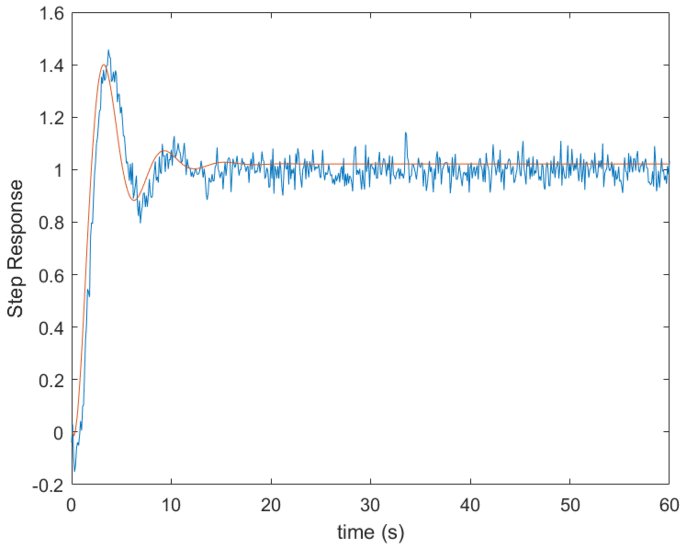

Figure 4: Step Response using the RLS algorithm

where the coefficient c is embeded into the nature of the noise. For this algorithm, the same procedure as in RLS was followed as well.

2.2.2 Finding n_a and n_b

The same procedure as with RLS was used to find the values of n_a and n_b using the ELS algorithm. The values for n_a and n_b were found to be 6 and 4 respectively. This is done using figure 6, and the closeness to the fit is represented in figure 7.

On comparing with RLS, we can see that there is not much improvement in the closeness of the fit for the case of ELS, and the reasoning behind this observation requires to be explored.

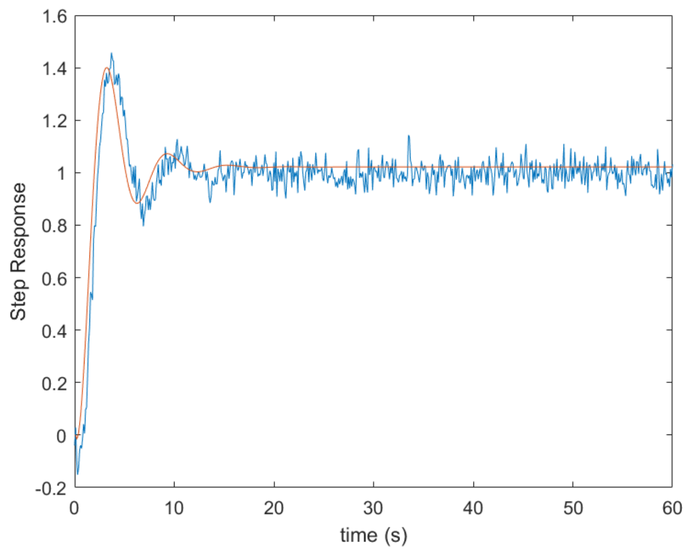

Figure 5: Step Response using the RLS algorithm (with λ)

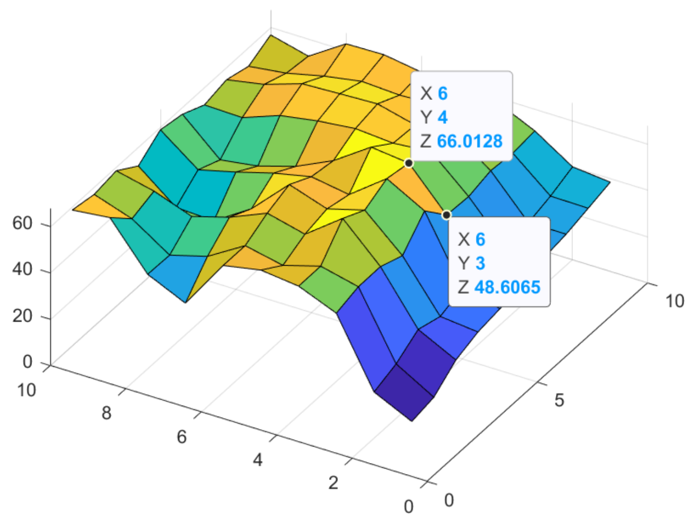

Figure 6: Plot of the %fit for different na and nb values for ELS

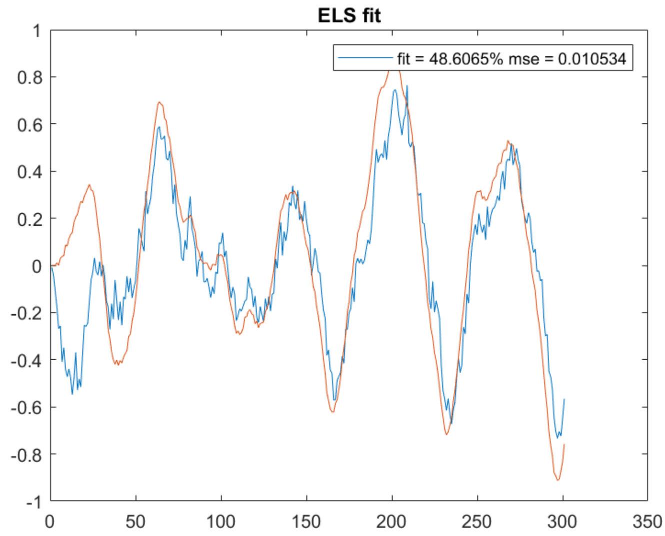

Figure 7: ELS fit

2.2.3 Incorporating Forgetting Factor

As done previously, we incorporate the forgetting factor by introducing $\lambda = 0.99$ to study and compare the betterness of fit. These curves are indicated in figure 8. Of all the cases previously seen, we can see that ELS with λ has the highest closeness to fit at 66.02%, indicating that this algorithm, could in fact, provide a better estimate of the model.

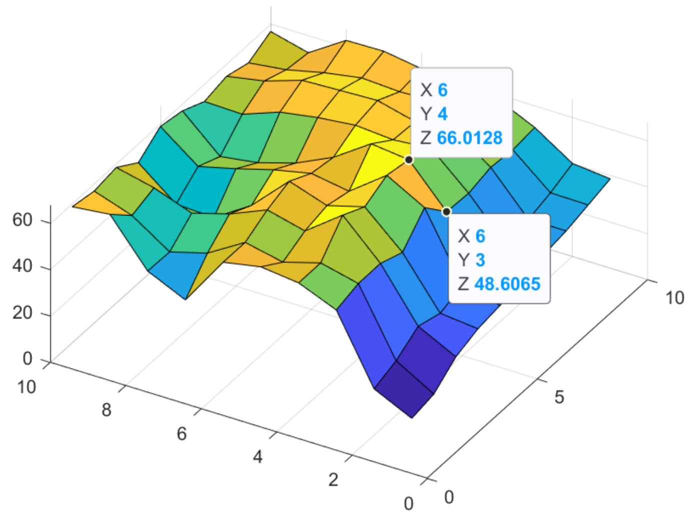

Figure 8: ELS fit with Forgetting Factor

2.2.4 Validation

The identified transfer function models (with and without forgetting factor) are validated by plotting the step response output y_{step} along with the step response from the identified model $\hat{y} = \frac{B}{A}u_{step}(t)$. These response curves are indicated in figures 9 and 10. Even though the % closeness to fit was found to be highest for the case of ELS with forgetting factor, however, the identified model doesn't seem to follow the measured data as accurately as the models from the other algorithms do.

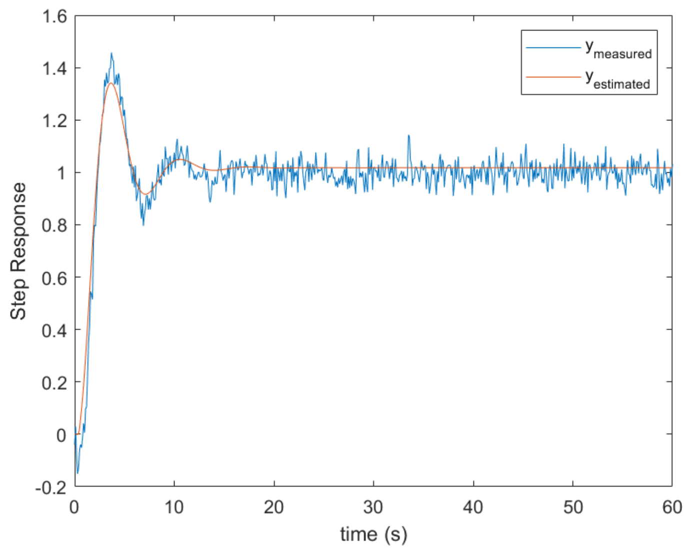

Figure 9: Step Response using the ELS algorithm

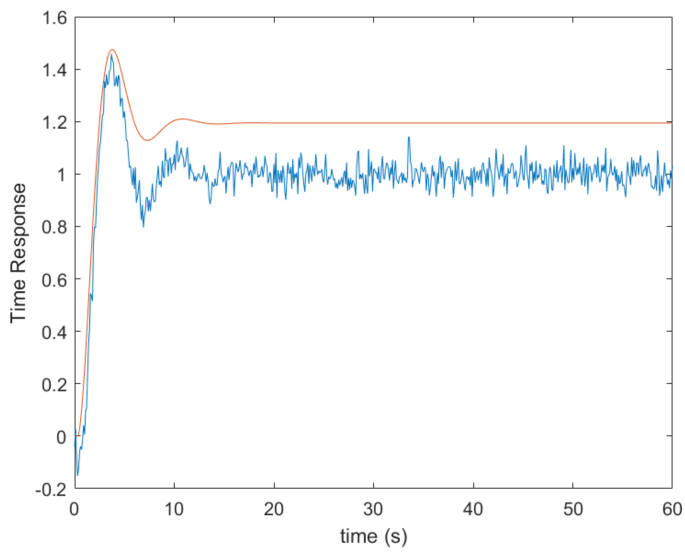

Figure 10: Step Response using the ELS algorithm (with forgetting factor)

2.3 Residual Analysis and Extent of Whiteness of Noise

Residual plots for both recursive algorithms are presented in Figures 11, 12 and 13. The expectation E of the noise is bounded by the confidence interval, indicated by the red lines. A notable observation is that the number of points within the confidence interval around $E = 0$ is significantly higher for RLS compared to ELS. This difference may be attributed to the "whiteness" of noise in the case of RLS, as white noise has an expectation $E = 0$. The lack of any discernible pattern in the auto-correlation for RLS supports this conclusion.

Conversely, in the case of ELS estimation, there is a finite number of points lying outside the interval. Since the noise is not necessarily white, there are samples with noise such that $E \neq 0$. These samples exhibit a certain degree of orderliness or reduced randomness compared to their RLS counterparts.

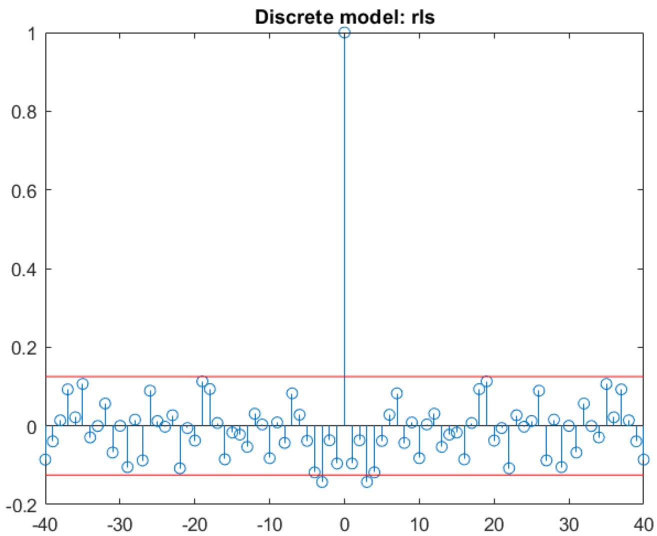

Figure 11: Auto-correlation for RLS

3 Identification of Continuous Time Model

Identification of a model in continuous time domain is more convenient since the coefficients have physical meaning, and are more intuitive and easier to work with. Additionally, precision for discrete-time models is essential, since it requires to take into account significant digits. Most importantly, discretisation of continuous transfer function with a very high sampling rate (or small Ts) contains poles that are very close to the boundary of a unit circle, which might lead to system instability due to protection issues.

3.1 State Variable Filter (SVF)

Suppose we have a dataset represented by $t = t_1, t_2, t_3, \ldots, t_N$ and seek to solve $Y'' = \Phi\theta$ through a linear regression approach employing the least squares algorithm. This method requires the estimation of $y''(t)$, $y'(t)$, and $u'(t)$. It is essential to note that this approach may

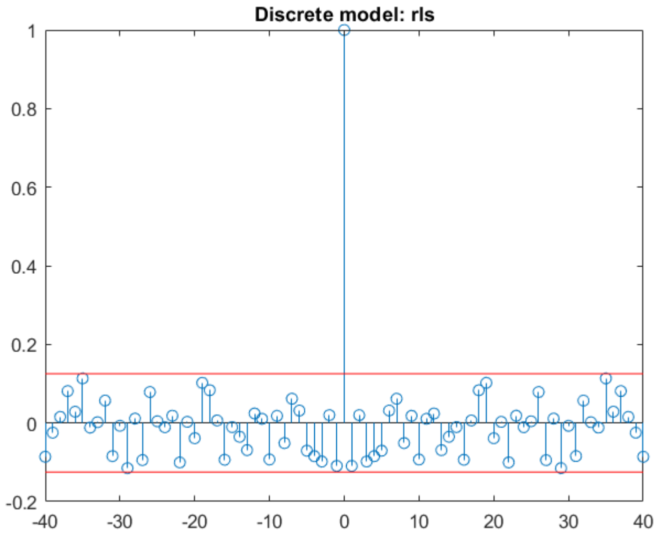

Figure 12: Auto-correlation for RLS with forgetting factor

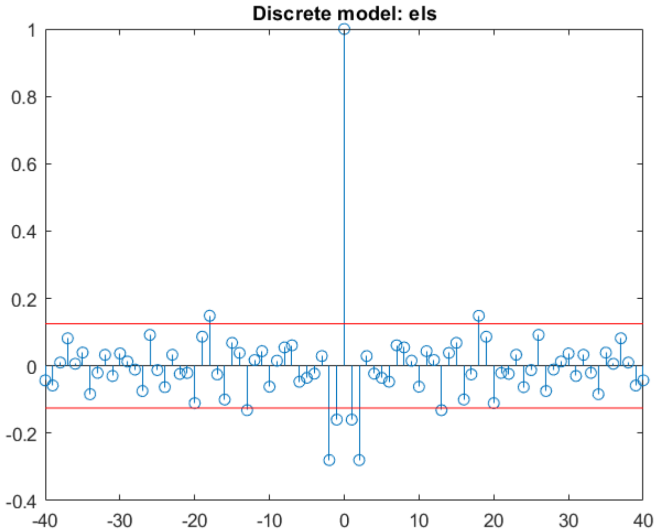

Figure 13: Auto-correlation for ELS

face challenges in signal processing applications. The estimation of derivatives involves the use of finite difference schemes, leading to increased noise with each derivative computation. As the order of the derivative rises, the noise amplification becomes more pronounced, posing a limitation in the accuracy of the results.

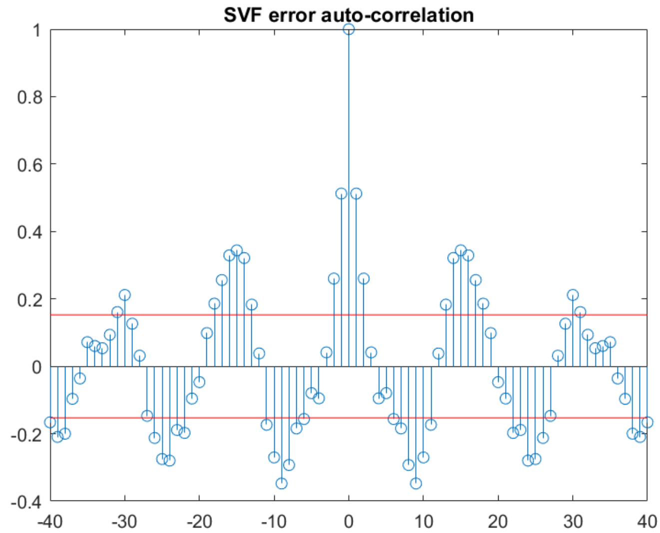

Figure 14: Autocorrelation using SVF

In the context of solving this problem, it is approached as a pure linear regression problem to obtain θ through the least squares algorithm, provided that estimations for $y''(t)$, $y'(t)$, and $u'(t)$ are available. However, it's worth noting that this approach encounters challenges in signal processing due to the necessity of computing derivatives (using Finite Difference Schemes) from the estimated y. The higher the derivative, the more susceptible the value is to noise amplification at each derivative computation.
An alternative solution to address this issue is the State Variable Filter (SVF).

3.1.1 Algorithm

$$s^2y(t) + a_1sy(t) + a_0y(t) = b_1su(t) + b_0u(t)$$

The SVF introduces a modification by pre-multiplying the above differential equation in the Laplace domain by $\frac{1}{(s+\alpha)^2}$, where α represents the filter bandwidth.

$$\frac{s^2}{(s+\alpha)^2}y(t) + \frac{a_1s}{(s+\alpha)^2}y(t) + \frac{a_0}{(s+\alpha)^2}y(t) = \frac{b_1s}{(s+\alpha)^2}u(t) + \frac{b_0}{(s+\alpha)^2}u(t)$$

The transfer function estimated is as follows:

$$G_{svf} = \frac{0.2044s^4 - 19.74s^3 + 62.16s^2 - 409.5s + 1195}{s^6 + 2.074s^5 + 97.26s^4 + 104.8s^3 + 1361s^2 + 711.6s + 1185}$$

It is evident from Figure 14 that a pronounced autocorrelation exists in the error term. This stems from the fact that ε is not white noise, making the application of the Least Squares

(LS) method inappropriate. The term $\frac{\varepsilon(t)}{(s+\alpha)^2}$ is correlated with $y_1(t)$. Even when expressed as $y_2 = \phi \cdot \theta^T + \varepsilon$, where ϕ and θ maintain their previous definitions, the LS method is still unsuitable. This is due to the nonzero mean of the last term in $\phi^T\phi\theta^T + \phi^T\epsilon$, resulting from the non-white noise nature of ε.

One effective approach to mitigate this problem is the utilization of the Instrumental Variable solution for SVF, hereafter referred to as "IVSVF."

3.2 Instrumental Variable solution for SVF

This algorithm is to be implemented such that

$$z^T y_2 = z^T \phi\theta + z^T \varepsilon \tag{2}$$

An "instrument" z must satisfy the above equation. The choice of z should ensure that $z^T\phi$ is invertible. Additionally, it is crucial to select z in such a way that the correlation between z and ε is minimized. For example, $z = \phi$ is acceptable only if z is uncorrelated with ε.
During the implementation of this algorithm for the previously estimated transfer function, it was observed that a satisfactory fit did not exist. This anomaly could have resulted from an inaccurate estimation of the transfer function's order or a potential error in the implementation itself. Nevertheless, to underscore the significance of IVSVF in eliminating correlation, the algorithm was executed for a nominal 2^{nd} order transfer function. The auto-correlation of the error for this case is depicted in Figure 15.

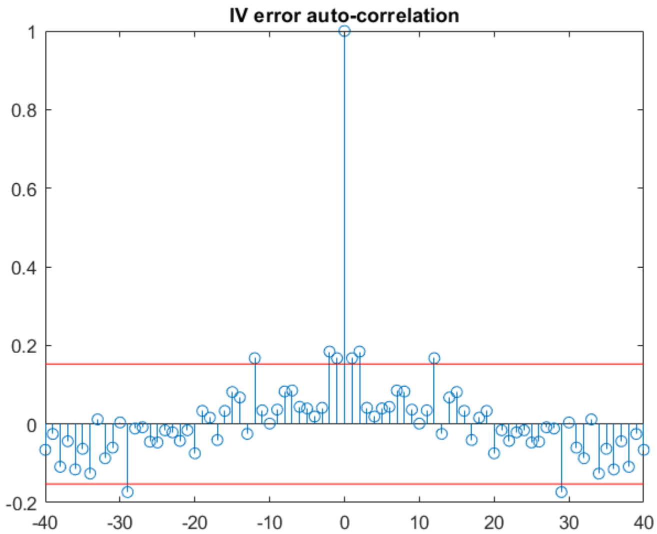

Figure 15: IVSVF auto-correlation

4 Overall Model Validation

A combined plot of all the different algorithms for both discrete and continuous time models are plotted in the figure 16.

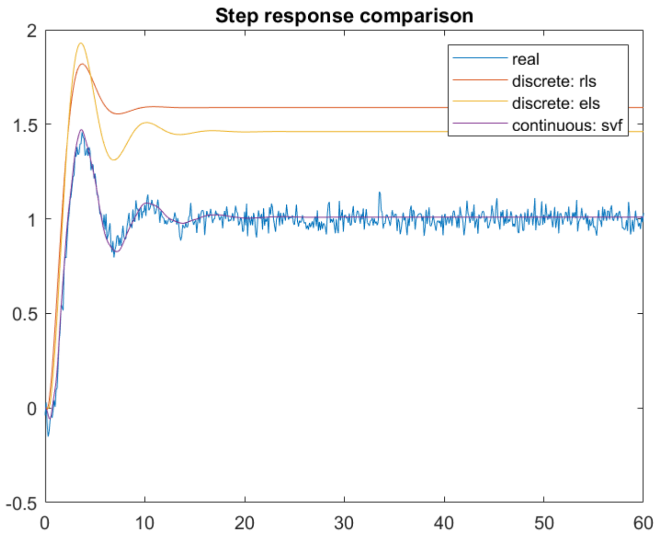

Figure 16: Step Response for different algorithms