Delay time estimation for linear systems by using Kalman filter

N. Nabavi 1, B. Moaveni 2

1 Research and science center, Islamic Azad University, Sayamon blv, Ashrafi Isfahani blv., Tehran, Iran, niloofar.nabavi@gmail.com
2 Research and science center, Islamic Azad University, Sayamon blv, Ashrafi Isfahani blv., Tehran, Iran, b_moaveni@eetd.kntu.ac.ir

Abstract
To compensate the delay and to design the predictive controller for time delayed system we need to know the delay time, so delay time estimation is very important stage to control the time delayed systems. In this paper, we first review some important methods of delay time estimation and then a new solution for delay time estimation problem is proposed. This new method is based on Kalman-Filter approach. Our proposed method is an offline delay time estimation method that can be used for SISO, MIMO and all other linear systems.

Keywords: Time-delayed systems, delay time estimation, Kalman filter

1. Introduction
Time delay is commonly encountered in various engineering systems, such as chemical processes, long transmission lines in pneumatic and hydraulic systems, economic systems and rolling mill systems. It usually results in unsatisfactory performance and is frequently a source of instability, so identifying the delay time and control of time-delay systems is practically important, especially in state feedback control. One of the most important steps to control a system is its modeling and identification. So, we should be able to model and estimate delay time in different systems. Time delays cause more complexity, especially if they were time varying. One major group of systems, which held this kind of complexity, is hybrid systems. In hybrid systems according to various states with different time delays, controlling the system will be more difficult. So, necessity of estimating delay times is more obvious in this kind of systems. Knowing the delay time help us to control the system easily and improve the performance of the system. In the other word, if we know the time delay we can apply the control action in proper time to compensate the time delay and to achieve better performance.

In next section we’ll define time-delay systems, which are followed by preliminary works done for delay-time estimation. In section 2, offline delay time estimation by Kalman filter will be described. In section 3, simulation results are presented for better realization.

1.1 Time-delayed system definition

A time delay may be defined as the time interval between the start of an event at one point in a system and its resulting action at another point in the system. Time delays are also known as transport lags, dead times or time lags; they arise in physical, chemical, biological and economic systems, as well as in the process of measurement and computation. Time delays are also common in signal processing applications.

There are different ways to model time-delayed systems. The state-space model of a typical discrete time-delayed system, can be presented as (1):

\[
\begin{align*}
\dot{x}(n+1) &= Ax(n) + Bu(t-d) \\
y(n) &= c \cdot x(n)
\end{align*}
\] (1)

Where A is the state vector; B is the input vector; d is a constant presenting delay.

1.2 Preliminary works in delay-time estimation

There are two major ways to estimate the delay-time, online delay estimation and offline delay estimation. Each of them is broadly divided as follows [1]:

Offline delay estimation techniques:
(a) Methods using known process characteristics
(b) Closed loop methods
(c) Experimental open loop methods
(d) Methods based on using a relay compensator

Online delay estimation techniques which this kind of estimation requires recursive algorithms:
(a) Methods that use rational approximations for the time delay, with recursive identification of the model parameters.
(b) Over-parameterization of the process in the discrete time domain.
1.2.1 Some of offline methods [1]

First researchers were working on offline delay time estimation are Unbehauen and Rao (1987). They described a method for estimating simultaneously the unknown delays and parameters of a SISO or MIMO process using the 'iterative shift algorithm' (ISA). The delay is chosen in the neighborhoods of its true value based on a priori knowledge, the vectors of unknown parameters (excluding time delay) are found recursively (using the RLS, RIV or recursive maximum likelihood (RML) estimation algorithms) for a number of subintervals of time T, where T is greater than the time delay; minimizing the sum of the norm of the difference in the parameter vectors identified, as the time delay is varied, gives the actual time delay value.

The Laguerre approximation for the time delay is used by De Souza et al. (1987), (1988) and by Salgado et al. (1988); all of these methods were using the standard (off-line) least squares estimation method to identify the parameters of the resulting model. After them, Nagy and Ljung (1991) define a state space model for the process and discuss a maximum likelihood estimate for the process parameters and the time delay based on this parameterization. In 1992, Kristinsson and Dumont discuss the use of genetic algorithms for process identification including the identification of the time delay; In 1993, Lublinsky and Fradkov, design both an adaptive control algorithm and an optimal, non-adaptive control algorithm based on uncertain knowledge of the process parameters. The parameters of a high order process, including time delay, are estimated; the time delay is estimated by finding the delay that minimizes the difference between the process and model outputs.

After them, Habermayer and Keviczky (1985) and Habermayer (1986) maximize the windowed sum of the square of the numerator parameters identified divided by the corresponding covariance matrix terms, as the time delay is varied, to find the optimum value of the time delay; simulation results show the method working in both noise-free and noisy environments. The maximum value of the time delay must be known a priori. Laakso et al. (1996) consider how to model a fractional delay in the sampled data domain. He presented a comprehensive review of finite impulse response (FIR) and all-pass filter design techniques for the band limited approximation of a fractional delay.

In 1998, Kulikov used a modified recurrent least squares algorithm to estimate the parameters and time delay of a process model on line in closed loop. The parameters of the process model are estimated in the z domain with the delay estimated in the s domain.

2. Offline delay time estimation using Kalman Filter

In the previous section we review some of the delay estimation techniques. Here we propose a new method to estimate the delay time by using Kalman filter. Here we use discrete-time Kalman filter and so discrete-time system models. First of all, we will predict the future value of system without considering the time-delay. Then we compare the actual and predicted signal value in step n. If the actual value and predicted value were equal in step n, then we don’t have delay in our system, otherwise, if the actual value was not equal to predicted value in step n, we will repeat this comparison for predicted value in step n and actual value in next steps. The difference between the steps, where predicted and actual values were equal in that, is the time-delay. With this method we can even estimate the time-varying time delays in various systems, including SISO and MIMO.

Thus, we have the following algorithm, shown in Fig.1, based on the new proposed method for delay time estimation. According to suggested algorithm, we predict the output signal in m step further with Kalman Filter by using the present data. Then we compare the actual value of the signal and predicted signal by computing the error like (2):

$$ e(n) = y(n) - y(n) $$  \hspace{1cm} (2)
Where \( e(n) \) is error in step \( n \), \( y(n) \) is the actual value of output signal in step \( n \). \( n(y\hat) \) is the predicted value in step \( n \). (\( n = 1, 2, \ldots \)) We consider \( \pm 0.001 \) as the accuracy. So, if
\[
(3) \quad \left| e(n) - y(n) \right| < 0.001
\]
Then, we can accept the predicted value is equal to actual value and delay is equal to \( x \).

### 3. Simulation results

Here we consider SHELL Heavy Oil Fractionators problem (Prett and Garcia 1988). The MIMO system is presented in [2], [3]. For simplicity, we consider the SISO model. The SISO transfer function of this system is (4):
\[
G(z) = \frac{4.05}{50z + 1} z^{-27}
\]
The state space equations can be presented as (5), (6):
\[
x(n) = -\frac{1}{50} x(n-1) + \frac{4.05}{50} u(n-d) + w(n) \quad (5)
\]
\[
y(n) = x(n) + v(n) \quad (6)
\]
Where \( w(n) \) is a white zero mean measurement noise with covariance matrix \( Q = 0.00015 I \), and \( v(n) \) is a white zero mean process noise with covariance matrix \( R = 0.0003 I \).

Now we use Kalman filter relations to estimate the future value of \( y(n) \). Then we have to compute the error defined in (3). For error estimation we must compare predicted value, \( \hat{y}(n) \), with the actual value in step \( n \), \( y(n) \). If they weren’t equal, we have to repeat the comparison for \( y(n) \) and \( \hat{y}(n+1) \) this time, and so on till \( y(n) \) and \( \hat{y}(n+x) \) be equal according to defined accuracy in (3). Therefore, the delay time is equal to \( x \). In this example, first we compute (7):
\[
(7) \quad e(1) = \hat{y}(1) - y(1)
\]
But, 
\[
(8) \quad e(1) \geq 0.001
\]
so according to our proposed flowchart, we must compute (9):
\[
(9) \quad e(2) = \hat{y}(2) - y(1)
\]
After taking 27 steps, we’ll have (10):
\[
(10) \quad e(27) = \hat{y}(28) - y(1)
\]
This time error satisfies relation (8). So, estimated delay time is equal to 27.

Some of the equalities between predicted values and actual values, for 62 data, are shown in table 1.

### 4. Conclusion

One of the most important plants to control is time-delayed plant and so one of the ways to improve the time-delayed system’s performance is to estimate the delay time. In this paper we proposed a new offline delay estimation method based on Kalman-filter. This method can be used for SISO and MIMO systems with time varying delay times.

### 5. Reference

[3] STAR® - Easy Model Predictive Control Implementation
Table 1. Shows the equality between actual and estimated values for delay time estimation. Accuracy=0.001

<table>
<thead>
<tr>
<th>Item</th>
<th>actual</th>
<th>estimated</th>
<th>Item</th>
<th>actual</th>
<th>estimated</th>
<th>Item</th>
<th>actual</th>
<th>estimated</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>0.072556</td>
<td>0</td>
<td>63</td>
<td>0.077171</td>
<td>0.074524</td>
<td>84</td>
<td>-0.079067</td>
<td>0.086378</td>
</tr>
<tr>
<td>43</td>
<td>0.078322</td>
<td>0</td>
<td>64</td>
<td>0.084893</td>
<td>0.07147</td>
<td>85</td>
<td>-0.08304</td>
<td>0.08019</td>
</tr>
<tr>
<td>44</td>
<td>0.079521</td>
<td>0</td>
<td>65</td>
<td>0.075286</td>
<td>0.073639</td>
<td>86</td>
<td>-0.078657</td>
<td>0.091345</td>
</tr>
<tr>
<td>45</td>
<td>0.082288</td>
<td>0</td>
<td>66</td>
<td>0.078898</td>
<td>0.09273</td>
<td>87</td>
<td>-0.075758</td>
<td>0.068168</td>
</tr>
<tr>
<td>46</td>
<td>0.077814</td>
<td>0</td>
<td>67</td>
<td>0.076864</td>
<td>0.087281</td>
<td>88</td>
<td>-0.085668</td>
<td>0.07977</td>
</tr>
<tr>
<td>47</td>
<td>0.08195</td>
<td>0</td>
<td>68</td>
<td>-0.084225</td>
<td>0.074497</td>
<td>89</td>
<td>-0.083948</td>
<td>0.072963</td>
</tr>
<tr>
<td>48</td>
<td>0.084369</td>
<td>0</td>
<td>69</td>
<td>-0.074266</td>
<td>0.077659</td>
<td>90</td>
<td>-0.081083</td>
<td>0.080449</td>
</tr>
<tr>
<td>49</td>
<td>0.083662</td>
<td>0</td>
<td>70</td>
<td>-0.07969</td>
<td>0.079076</td>
<td>91</td>
<td>-0.079125</td>
<td>0.083148</td>
</tr>
<tr>
<td>50</td>
<td>0.087881</td>
<td>0</td>
<td>71</td>
<td>-0.076773</td>
<td>0.080198</td>
<td>92</td>
<td>-0.079209</td>
<td>0.073562</td>
</tr>
<tr>
<td>51</td>
<td>0.078507</td>
<td>0</td>
<td>72</td>
<td>-0.080248</td>
<td>0.08511</td>
<td>93</td>
<td>-0.080344</td>
<td>0.079041</td>
</tr>
<tr>
<td>52</td>
<td>0.077788</td>
<td>0</td>
<td>73</td>
<td>-0.07882</td>
<td>0.074947</td>
<td>94</td>
<td>-0.081354</td>
<td>0.087356</td>
</tr>
<tr>
<td>53</td>
<td>0.077037</td>
<td>0</td>
<td>74</td>
<td>-0.079917</td>
<td>0.078908</td>
<td>95</td>
<td>-0.080922</td>
<td>-0.09644</td>
</tr>
<tr>
<td>54</td>
<td>0.084461</td>
<td>0</td>
<td>75</td>
<td>-0.082441</td>
<td>0.08577</td>
<td>96</td>
<td>-0.074955</td>
<td>-0.065683</td>
</tr>
<tr>
<td>55</td>
<td>0.079933</td>
<td>0.067659</td>
<td>76</td>
<td>-0.070779</td>
<td>0.084227</td>
<td>97</td>
<td>-0.079939</td>
<td>-0.07773</td>
</tr>
<tr>
<td>56</td>
<td>0.076971</td>
<td>0.08486</td>
<td>77</td>
<td>-0.079672</td>
<td>0.080005</td>
<td>98</td>
<td>-0.082763</td>
<td>-0.073212</td>
</tr>
<tr>
<td>57</td>
<td>0.081885</td>
<td>0.079053</td>
<td>78</td>
<td>-0.078729</td>
<td>0.076583</td>
<td>99</td>
<td>-0.083868</td>
<td>-0.083143</td>
</tr>
<tr>
<td>58</td>
<td>0.077978</td>
<td>0.087572</td>
<td>79</td>
<td>-0.077077</td>
<td>0.080051</td>
<td>100</td>
<td>-0.078546</td>
<td>-0.08198</td>
</tr>
<tr>
<td>59</td>
<td>0.08105</td>
<td>0.073946</td>
<td>80</td>
<td>-0.075816</td>
<td>0.078118</td>
<td>101</td>
<td>-0.075397</td>
<td>-0.078118</td>
</tr>
<tr>
<td>60</td>
<td>0.074923</td>
<td>0.086175</td>
<td>81</td>
<td>-0.080774</td>
<td>0.082261</td>
<td>102</td>
<td>-0.077276</td>
<td>-0.08403</td>
</tr>
<tr>
<td>61</td>
<td>0.079197</td>
<td>0.083104</td>
<td>82</td>
<td>-0.085716</td>
<td>0.078126</td>
<td>103</td>
<td>-0.087253</td>
<td>-0.075171</td>
</tr>
<tr>
<td>62</td>
<td>0.082613</td>
<td>0.072386</td>
<td>83</td>
<td>-0.08216</td>
<td>0.075899</td>
<td>104</td>
<td>-0.080076</td>
<td>-0.087911</td>
</tr>
</tbody>
</table>