Comparison of two B-polynomial methods: application to the identification of time delayed processes

Ruiyao Gao
Technological University Dublin

Aidan O'Dwyer
Technological University Dublin, aidan.odwyer@tudublin.ie

Proceedings of the Irish Signals and Systems Conference, NUI Maynooth, Ireland, June, pp. 105-111.

Comparison of Two B-polynomial Methods

Application in the Identification of Time Delayed Processes

Ruiyao Gao, Aidan O'Dwyer

School of Control System & Electrical Engineering
Dublin Institute of Technology,
Kevin St., Dublin 8,
Ireland

Abstract

An extensive literature exists on the estimation of the model parameters of time delayed processes. The objective of this paper is to review the main B-polynomial approaches to the delay estimation problem. More specifically, two algorithms are discussed and compared theoretically and in simulation. The algorithms involve various procedures to assess how the model relates to the observed data, to a priori knowledge, and to the algorithms' intended use. Simulation results indicate that both the algorithms can suitably identify the model parameters, including the time delay. This paper also discusses advantages and disadvantages of the methods and gives some guidelines for choosing the appropriate algorithm.

1. INTRODUCTION

Many industrial plants have inherent time delays. They can be found in chemical processes, biomedical processes, or even in a simple pressure control loop. The correct estimation of delay can be exploited to enhance closed-loop performance and is of importance for controller design, which motivates research of simple and reliable extended B-polynomial identification algorithms. The method of B-polynomial is widely used to estimate the process parameters (including unknown time delay). It involves subsuming the time delay term (in the z domain) into an extended numerator polynomial. The corresponding parameters are estimated using a recursive estimation scheme, and the time delay is calculated based on the parameters identified [1]. The best time delay is the one that yields the smallest value of the quadratic loss function. Vogel and Edgar [2] adopt the extended B-polynomial method to avoid an explicit on-line estimate of the discrete delay in an adaptive pole/zero placement control strategy: they simply retain the whole estimated B-polynomial as the desired closed-loop transfer function. Clough and Park [3] apply the extended B-polynomial method to determine the optimal prediction step (which may not coincide with the true delay d) in a standard minimum-variance self-tuning controller, while Chien et al [4] replace completely the d step ahead predictor with a Smith predictor, based on the extended B-polynomial.

Two promising methods are proposed by Kurz and Goedecke [5] and Teng and Sirisena [6]. The method of Kurz and Goedecke (RLSVT for short) finds the maximum value of the numerator parameters and declares that the time delay index, which is the time delay divided by the sample time, is less than or equal to this value. A quadratic loss function for each value of the time delay index is then calculated recursively, based on a priori knowledge of process order. The minimum value of the quadratic loss function corresponds to the time delay estimate. This robust method for estimating the SISO model parameters is equivalent to determining the best match between the impulse response of an over-parameterized model and the impulse response of a non-over-parameterized model with a pure delay. However, the method is computationally intensive.

Teng and Sirisena present an alternative cost function, which calculates the error function of the sum of the numerator parameters between the over-parameterized model and the estimated model, for each value of the time delay index. The method (ENLSE for short) offers an interesting trade-off between robustness and computational simplicity.
The rest of this paper will follow the statement of the RLSVT and ENLSE algorithms in Section 2. Simulation work was performed to identify a theoretical one-order system with long delay, two-order system with long delay and a dynamic process high order system, respectively, in Section 3. Section 4 gives the conclusions.

2. PRELIMINARIES

2.1. Kurz algorithm RLSVT

Assuming that the process can be described by a linear difference equation (2-1) of order n with constant parameters

$$y(k) = -a_1 y(k-1) - \ldots - a_n y(k-n) + b_1 u(k-d-1) + \ldots + b_m u(k-d-m) + w(k) \qquad (2\text{-}1)$$

u(k) and y(k) are the sampled process input and output signals, and w(k) is a white noise signal. The discrete process time delay index is d=Tt/T, where d is non-negative integer, Tt is time delay, T is sampling time. The z transformation of equation (2-1) results in

$$y(z) = \frac{B\left(z^{-1}\right)}{A\left(z^{-1}\right)} z^{-d} u(z) + w(z) \qquad (2\text{-}2)$$

RLSVT is an algorithm based on RLS [5], in which equation (2-2) can be rewritten in a modified structure as follows

$$y(z) = \frac{B^{*}\left(z^{-1}\right)}{A\left(z^{-1}\right)} u(z) + w(z) \qquad (2\text{-}3)$$

with $B^{*}(z^{-1}) = b_1^{*} z^{-1} + b_2^{*} z^{-2} + \ldots b_{m+d_{\max}}^{*} z^{-(m+d_{\max})}$ (2-4)

with d_{max} being the upper limit of the process time delay index which has to be assumed known a priori. For a process with the time delay index d, the following equations hold

$b_i^{*} = 0$, i=1,2…d;

$b_j^{*} = b_{j-d}$, j=1+d,…m+d;

$b_i^{*} = 0$, i=m+1+d,…m+d_{max}..

The basis of RLSVT algorithm is to use this modified structure for the parameter estimation; the unified recursive parameter estimation algorithm is used:

$$\hat{q}(k+1) = \hat{q}(k) + g(k)e(k+1) \qquad (2\text{-}5)$$

$$g(k) = u(k+1)P(k)y(k+1) \qquad (2\text{-}6)$$

$$e(k+1) = y(k+1) - y^{T}(k+1)\hat{q}(k) \qquad (2\text{-}7)$$

With the following conditions:

$$\hat{q}(k) = [\hat{a}_1(k), \ldots \hat{a}_n(k), \hat{b}_1^{*}(k), \ldots \hat{b}_{m+d_{\max}}^{*}(k)] \qquad (2\text{-}8)$$

$$y(k) = \begin{bmatrix} -y(k-1) \\ \vdots \\ -y(k-n) \\ u(k-1) \\ \vdots \\ u(k-m-d_{\max}) \end{bmatrix}^{T} \qquad (2\text{-}9)$$

$$u(k+1) = \left[l + y^{T}(k+1)P(k)y(k+1)\right]^{-1} \qquad (2\text{-}10)$$

$$P(k+1) = \left[I - g(k)y^{T}(k+1)\right]P(k)/l \qquad (2\text{-}11)$$

Applying (2-8) and (2-9) in the recursive parameter estimation algorithm, hence the modified process model

$$\hat{G}^{*}(z) = \frac{\hat{B}^{*}(z^{-1})}{\hat{A}(z^{-1})} = \frac{\hat{b}_1^{*} z^{-1} + \ldots + \hat{b}_{m+d_{\max}}^{*} z^{-(m+d_{\max})}}{1 + \hat{a}_1 z^{-1} + \ldots + \hat{a}_n z^{-n}} \qquad (2\text{-}12)$$

is estimated.

Simulations based on the algorithm above have shown that it is difficult in some cases to estimate the model time delay from the first numbered parameters of $\hat{B}^{*}(z^{-1})$ which are small in comparison to the succeeding parameters of this polynomial. Therefore the following procedure is used for the estimation of $\hat{B}(z^{-1})$ and $\hat{d}$:

Step1: Determine the maximal parameter $\hat{b}_{d_{\max}}^{*}$ of $\hat{B}^{*}(z^{-1})$, i.e.

$\hat{b}_{d_{\max}}^{*} = \max\left\{\hat{b}_i^{*}, i = 1, \ldots, m + d_{\max}\right\}$. Then $0 \le \hat{d} \le d_{\max}$ is valid.

Step 2: Calculate the error functions

$F(d) = \sum_{k=1}^{N} \Delta g_d^2(k)$; d =0… d_{max}, where $\Delta g_d(k) = \hat{g}^{*}(k) - \hat{g}_d(k)$ with $\hat{g}^{*}(k)$…weighting coefficients of the model

$\hat{G}^{*}(z) = \frac{\hat{B}^{*}\left(z^{-1}\right)}{\hat{A}\left(z^{-1}\right)} = \sum_{i=0}^{\infty} \hat{g}^{*}(i) z^{-i}$, $\hat{g}_d(k)$ … weighting coefficients of the model

$$\hat{G}_d(z) = \frac{\hat{B}\left(z^{-1}\right)}{\hat{A}\left(z^{-1}\right)} z^{-d} = \sum_{i=0}^{\infty} \hat{g}_d(i) z^{-i} (\hat{g}_d(i) = 0, i \le d)$$

Step 3: Determine the minimal value $F(\hat{d})$ of $F(d)$, d =0, …, d_{max}, i.e.

$F(\hat{d}) = \min\{F(d), d = 0, \ldots d_{max}\}$, then $\hat{d}$ is the estimated model time delay.

Step 4: Calculate the parameters $\hat{b}_i$ of the process model using the following equations:

$$\begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \vdots \\ \hat{b}_m \end{bmatrix} = \begin{bmatrix} \hat{g}^*(1+\hat{d}) \\ \hat{g}^*(2+\hat{d}) + \hat{a}_1 \hat{g}^*(1+\hat{d}) \\ \vdots \\ \hat{g}^*(m+\hat{d}) + \hat{a}_1 \hat{g}^*(m-1+\hat{d}) + \ldots \hat{a}_{m-1} \hat{g}^*(1+\hat{d}) \end{bmatrix} \qquad (2\text{-}13)$$

Equation (2-13) results from equating the values $\hat{g}_d(k)$, k=1+ $\hat{d}$,…, $m+\hat{d}$ with the values $\hat{g}_d^*(k)$, k=1+$\hat{d}$,…, $m+\hat{d}$.

The estimation of parameters of $\hat{A}(z^{-1})$ and $\hat{B}^*(z^{-1})$ corresponding to the model time delay index $\hat{d}$ (step1-3) and the calculation of the parameters of $\hat{B}(z^{-1})$ (step 4) are done in every sampling interval.

As a result, the model can be given like

$$\hat{G}(z) = \frac{\hat{B}(z^{-1})}{\hat{A}(z^{-1})} = \frac{\hat{b}_1 z^{-1} + \ldots + \hat{b}_m z^{-m}}{1 + \hat{a}_1 z^{-1} + \ldots + \hat{a}_n z^{-n}} z^{-\hat{d}} \qquad (2\text{-}14)$$

2.2. Teng algorithm ENLSE:

Teng proposed a simple yet effective method, which gives the correct parameter estimates for a system with time delay controlled by the self-tuning PID controller. Assuming that the process is also defined by equation (2-1), the algorithm used can be divided into two stages as follows:

Stage A has four steps:

1) Use over-parameterized model (2-3) for RLS estimator.
2) Determine the maximum parameter $\hat{b}^*_{d_{max}}$ of $\hat{B}^*(z^{-1})$, $\hat{b}^*_{d_{max}}$ =max ($\hat{b}_i^*$, i=1,…, m+d_{max}), then $0 \le \hat{d} \le d_{max}$ is valid.
3) Calculate the error function

$F(d) = \left|\hat{B}^*(d) - \hat{B}_d(d)\right|$ with $\hat{B}^*(d) = \Sigma \hat{b}_i$, i=1, …, m+d_{max};

$\hat{B}_d(d) = \sum_{i=1}^{m} \hat{b}_{i+d}, d = 0,\ldots,d_{max}$.

4) Determine the minimal value $F(d)$ for d=0,… , d_{max}, and time delay $\hat{d}$ is obtained when F (d) is minimum.

Stage B:

Applying the knowledge of time delay from stage A, the estimated model [Equation (2-14)] may be obtained.

3. SIMULATION

Both RLSVT and ENLSE algorithms are applied to identify processes with time delay in the discrete domain. They require an assumption of the sum of the maximal order of the process and maximal delay parameter, which is assumed as 8 in all simulation. Many simulation results were collected. Some of them are provided below.

3.1. First order lag plus delay (FOLPD) model

Considering the model, whose transfer function is $G(s) = \frac{1}{1+Ts} e^{-ts}$ (T=1, t = 1). Convert the transfer function to discrete time domain using zero order hold (ZOH) method through modified Z Trans-formations [6]. When a sampled-data control system contains a time delay index Tt= dTs, assuming n is an integer and l is a fraction, Tt can be rewritten as Tt=dTs=(n-1)Ts+ l Ts, 0< l <1. For the above simulated process, its discrete time transfer function has denominator order na=1, numerator order nb=1 (when l is integral) or 2 (when l is fractional). Using modified z transform, it is clear that nb=na (l is integral) or nb=na+1 (l is fractional). Assuming a priori knowledge of system order na and nb, estimate the model as previously described in section 2.1 & 2.2 (Figures showing is based on sampling timeTs=0.4).

RLSVT algorithm:

This algorithm always presents the desirable estimated model when we have exact a priori knowledge of na and nb. Otherwise, zeros may appear in the estimated model. It is interesting that in some cases, hidden zero-pole cancellation are shown in the Zero-pole plots when the orders are bigger than the true value. An example is shown in Figure 2. After the zero-pole cancellation command is performed, the desired model is obtained.

ENLSE algorithm:

Simulation results shows the best model is estimated when we have a priori knowledge of exact value of na and nb. When nb=na or nb=na+1, simulation results are the same as those from the RLSVT algorithm. Hidden zero-pole cancellation is also a feature of the results (Figure 2).

Analysis of simulation results:

The limitation of the two algorithms is that both of them need a priori knowledge of the process order. When na and nb are equal to or greater than their real value, Kurz algorithm gives exact integer delay time. However, the performance of Teng algorithm depends on a priori knowledge of system order. And if it is assumed that nb=na or nb=na+1, both algorithms give the same exact integer delay time, but the estimated model has hidden zeros to cancel poles if na and nb are greater than their real values.

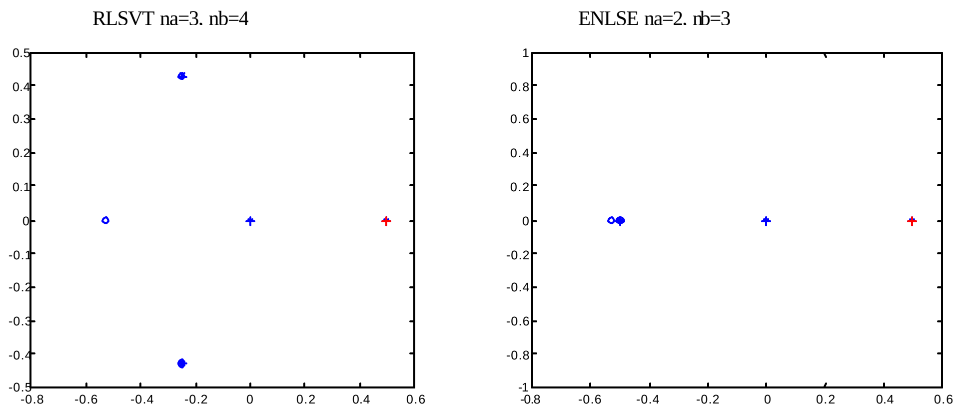


Figure 2: zero-pole cancellation are shown in the figures. '.' and '+' are zeros and poles from the the model, 'o' and '*' are zeros and poles from the estimated models.

Step 3: Estimation in the presence of noise.

In order to analyze the sensitivity of the estimation algorithms to the presence of noise corrupting the data, white noise signals with zero mean value and standard deviation m =0.2 was added to the model output. The RLSVT and ENLSE algorithms are performed, in the case that there is no a priori knowledge of na and nb.

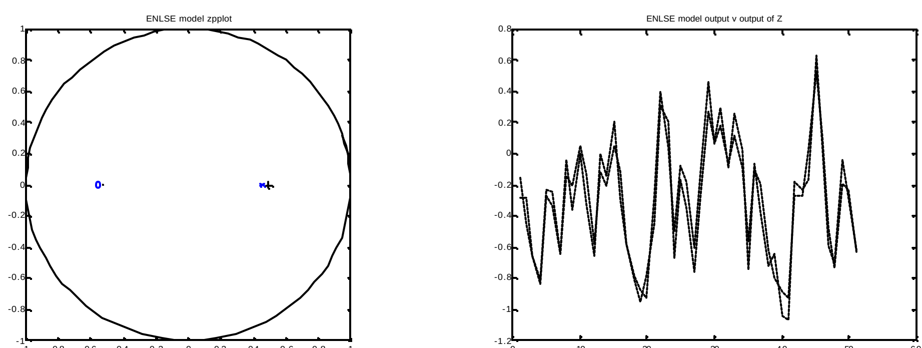


Figure 3: noise u=0.2, na=1,nb=2. '.'and '+' are the zero and pole without noise. 'o' and '*' are zero and pole with noise. The solid line is from the model; the dashed line is from estimated model.

The simulation result shows that the ENLSE algorithm works as well as the RLSVT algorithm, so the figure 3 using the ENLSE algorithm is given as an example. Zero and pole from the ENLSE deviate from the model zero and pole when adding the noise, but they are still very close. We can see the goodness of the matching from the comparison figure above. The algorithms also present the correct time delay index estimates, which are not affected by the noise added, although the noise causes some errors in the estimates of the model parameters.

3.2. Second order system plus delay (SOSPD) model

The following two systems are inspired by the results of biological data encountered in reference paper [7], in which they are applied as models to describe the relationship of respiration, blood pressure and heart rate.

Z1: $H_1(Z) = \dfrac{-z^{-3}}{1-1.2z^{-1}+0.35z^{-2}}$

Z2: $H_2(Z) = \dfrac{1z^{-1}-1.3z^{-4}}{1-1.2z^{-1}+0.35z^{-2}}$

$H_1(Z)$ is a delay system, but $H_2(Z)$ is not a delay system. Simulation results for estimating the model are encouraging. The simulation also compares the RLSVT and ENLSE algorithm with the ARX algorithm, for the use of the ARX algorithm has become very common practice in modeling linear, time invariant systems using their input/output data [7].

There are still some important observations to make regarding the above results. First, it should be noted that the goodness of fit improved when the order of the selected model increased, and the fit value is almost constant when the selected order is greater than the real system order. This is true for all the three algorithms. Second, the RLSVT and ENLSE simulation results always present the desirable estimated model when we have exact a priori knowledge of na and nb, and both algorithms provide a much better model than the ARX algorithm. For estimating $H_2(Z)$, both the RLSVT and ENLSE algorithms work better than the ARX algorithm. Figure 4 shows an example in the case na=2, nb=4.

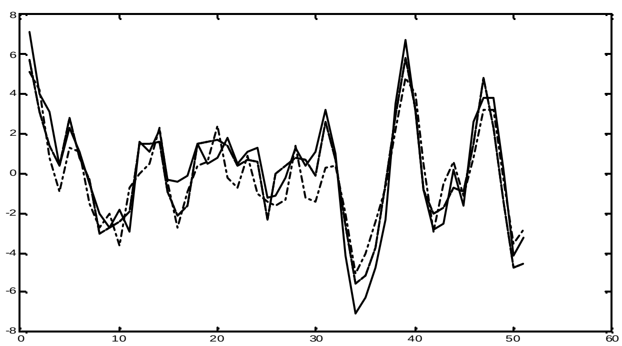


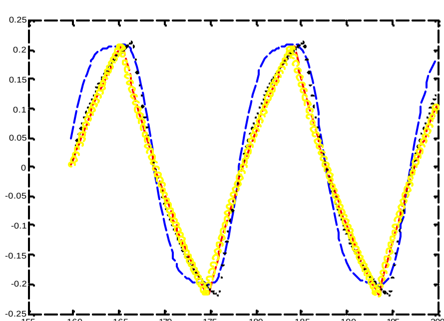
Figure 4: output comparison. Solid line is from system model; '-.' dash-dot line is from RLSVT algorithm; '.' dot line is from ENLSE algorithm; and '- -' dashed line is from ARX algorithm.

3.3. A physical high order system

It is well-known that most high order physical systems can be represented by a first order or second order system with time delay, in which the delay estimate is a combination of an actual delay and contributions due to high order dynamic terms in the process transfer function [8,9]. So the possibility of modeling a high order process, which has no physical delay, by a low order time delayed model, is examined. The process considered is the PCS327 process Simulator from Feedback Instruments Limited, configured as one lag plus a distance-velocity lag circuit. Analysis reveals that the process has the following transfer function $G(s)$.

$$G(s) = \frac{1}{1.106s^4 + 5.882s^3 + 13.39s^2 + 17.98s + 9.36}$$

The simulation results are shown in figure 6, which is the comparison of the model output and the measured output from the Process Simulator, we can see the goodness of fit increases when na and nb increase. The fit value improves when na and nb are greater than 2. This point can be seen in figure 7, which is the standard deviation function of the error between the model output and the measured output. Furthermore, when the system order (na and nb) is greater than 4, the model from the ARX algorithm matches the real system output as well as the models from the RLSVT and ENLSE algorithms.



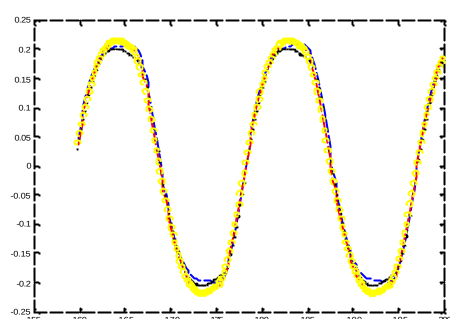


Figure 6: the first figure is obtained when a priori of na=1,nb=2,and the second one is obtained when na=4,nb=5.The solid line is model output; '- - 'dashed line is ENLSE algorithm model output; '-.'dash-dot line is RLSVT algorithm model output; and '.' dot line is ARX algorithm output.

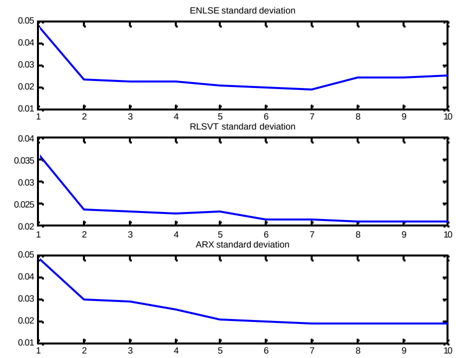


Figure 7: standard deviation funtion from na=1:10 and nb=2:11.

4. CONCLUSION:

Two types of extended B-polynomial identification methods in the discrete time domain, both of which are applied for estimating the system delay and other parameters, are described. Summarizing the simulation, there are some points to mention. First, in the case there is no a priori knowledge of system order, matching goes up when the model order goes up. When the model order is lower than the true value, RLSVT shows better robustness than ENLSE. When the assuming system order is equal to or greater than the true value, these two algorithms get the same results. If the system order is known a priori, both the algorithms can give desired estimated model, although ENLSE has much less computational load. For real processes, the RLSVT and ENLSE work better to systems with delay than the ARX algorithm. Whether the RLSVT or ENLSE algorithm is good enough as identification tools depends on the purpose of the identification algorithm. Further work is going to explore the approaches in formulating the controllers for time delayed processes based on the above identification methods.

REFERENCES:

1. A.O'Dwyer, J.V.Ringwood(1999), " A Classification of Techniques for the estimation of the model parameters of a time delayed process." Modern Applied Mathematical techniques in circuits, Systems and control, World scientific and Engineering Society Press. ISBN:960-8052-05-X,PP 187-198.
2. E. F. Vogel and T. F. Edgar, "Application of an adaptive Pole-Zero Placement Controller to Chemical Processes with Variable Dead Time" Proceedings, American Control Conference, pp. 786-791 ,1982.
3. D. E. Clough and S. J. Park, "A Novel Dead Time Adaptive Controller", IFAC Workshop on Adaptive Control of Chemical Processes, pp. 19-24,1985.
4. I. L. Chien, D. A. Mellichamp and D. E. Seborg, "A self-tuning controller for sys-tems with unknown or varying time delays", Proceedings, American Control Conference, pp. 905-912, 1984.
5. H. KURZ and W. Goedecke, "Digital Parameter-adaptive Control of Process with Unknown Dead Time." Automatic, Vol. 7, No.1, pp. 245-252, 1981
6. Teng Fong-Chwee and H. R. Sirisena, "Self-tuning PID Controllers for Dead Time Process". IEEE Transactions on Industrial Electronics, , Vol.35, PP. 119-125, 1988.
7. Michael H. Perrott and Richard J. Cohen, " An efficient approach to ARMA modeling of biological systems with multiple inputs and delays".IEEE Transaction on Biomedical Engineering, VoL 43, No.1, January, PP 1-14, 1996.
8. A.B. RAD, "Self-tuning Control of Systems with Unknown Time Delays: A Continuous –time Approach", Control Theory and Advanced Technology, Vol 10, No.3, PP 479-497, 1994.
9. Karl J. Åström and Björn Wittenmark, 'Adaptive Control'. Addison-Wesley Publishing Company,Inc.Second Edition,1995.