

# A linear regression variable time delay estimation algorithm for the analysis of hydraulic manipulators

O. L. R. Albrecht and C. J. Taylor

Engineering Department, Lancaster University, UK

Email: o.albrecht@lancaster.ac.uk, c.taylor@lancaster.ac.uk

**Abstract**—An algorithm to investigate nonlinear systems with time delay variability is proposed. It automatically cycles through segments of an open loop experiment, capturing the directional change in the output via a pair of regression equations, and estimates the time delay accordingly. The method is a black-box, statistical approach that does not require a dynamic model nor any knowledge of the systematic causes of the time delay variability. It is designed for systems with approximately integrating behaviour, hence it is used in this article for the analysis of hydraulically actuated robotic manipulators with this characteristic and time-varying delays. In the context of control, the new algorithm provides insight into the variable time delay behaviour and hence can guide control design decisions e.g. by uncovering state dependencies.

**Index Terms**—uncertain time delay, time-varying delay, dead-band, hydraulic manipulators

## I. INTRODUCTION

Many physical systems have time delays and these are most typically observed and subsequently modelled as time-invariant delays [1, 2]. Delays are sometimes referred to as an after-effect or a dead-time, and the system itself can be referred to as a hereditary system. Time delays typically manifest as a mixture of communication delays [3], calculation delays, and delays that occur due to mechanical or other dynamics. These include systems with, for example, actuators, sensors, field networks involved in feed-back loops, and networked control systems [4]. Well-known identification and control methods for time delay systems include functional differential equations, fractional derivation equations, transport equations, the Smith Predictor, among many others.

Particular difficulty is introduced, however, by systems with time delays that are relatively long, unknown and potentially time-variable. As discussed by [5], there are a number of methods for modelling time-varying delays e.g. [6–9]. Examples of such systems can be found across science and engineering, from e.g. insulin delivery control in medicine [10] to hydraulically actuated robotic manipulators. With regard to the latter, in some scenarios the time-delay (in seconds) between implementing a change in the applied voltage and observing the associated angular velocity response, can change over time. Such variations in the delay may be caused by the internal dynamics of the system and other nonlinear characteristics, such as fluid compressibility, varying pressure

dynamics, dead-band of the pump, valve flow properties and friction characteristics [5]. These variable time-delays greatly complicate the process of system identification. Indeed, without knowledge of the delay, many methods can become wildly inaccurate. Furthermore, the presence of variable time delays in control systems can lead to instability, reduced robustness, a sluggish response and, in general, detuned control [11].

The present article develops an algorithm to help investigate the behaviour of a system with a variable time delay. The method is a black-box, statistical estimation approach that does not require a dynamic model nor any knowledge of the systematic causes of the time delay variability. In the context of control, the algorithm can provide insight into the time delay behaviour and hence guide design decisions. For instance, the algorithm provides information to help decipher whether the time delay is state dependent, in which case it can be predicted and used for state-dependent control [12], or whether it is dependent on multiple conditions such that it appears stochastic, in which case, it might be estimated online using existing methods e.g. via neural networks [6] or model comparison [7].

The algorithm has been developed and initially evaluated for the analysis of hydraulically actuated robotic manipulators, for which prior work has tended to utilise an integrating model with time delays [12–14]. In such cases, the Refined Instrumental Variable (RIV) algorithm and associated statistical criteria, were used to estimate linear models with time-invariant delays for a range of operating conditions. Alternatively, nonlinear models were obtained by directly estimating the state dependent parameters from experimental data [15]. Both these approaches are limited to time-invariant delays. However, recent research by the present authors has investigated how to obtain improved manipulator control performance [16], and this has motivated a re-examination of the experimental data to highlight the scenarios in which the time-delay varies.

Graphs of how the delay varies with operating condition can potentially be used to develop new improved state-dependent models. However, the RIV based approach alluded to above is necessarily used in conjunction with classical step experiments with steady state initial conditions, and does not fully capture the nonlinear dynamic behaviour manifested in longer experiments with multiple input magnitudes. By contrast, the new algorithm automatically cycles through segments of experimental data, capturing the directional changes in the

output associated with each input change, and updating the time delay accordingly.

The method is applicable to open-loop experiments in which the input signal switches between different magnitudes at random intervals in time, and the response of the nonlinear system possesses integrator-like properties (see laboratory example below). In contrast to non-integrator systems, the integrator-like response is a particularly unique case for time delay analysis in that it has no steady state. Thus, while the RIV based approach is suitable for non-integrator systems, it yields unsatisfactory results for the integrator-like response. The proposed algorithm, hence, provides a unique solution to the given problem. Finally, in contrast to some of the neural network based approaches alluded to above, the proposed algorithm has a low computational cost, is straightforward to implement and is not linked to a specific dynamic model or control method.

The algorithm is introduced in the following Section II. The laboratory example is described in Section III. This is followed by the discussion and conclusions in Sections IV and V.

## II. TIME DELAY ESTIMATION METHODOLOGY

To illustrate the concept of a varying time delay, consider the following generic model for manipulator angle  $y_k$ ,

$$y_k = \text{fn} \{u_{k-\tau_k}\} \quad (1)$$

where  $\text{fn}$  represents a discrete-time dynamic model, linear or otherwise,  $u_k$  is the input and  $\tau_k$  is the pure time delay at sample  $k$ . For the manipulator,  $u_k$  is an applied voltage, scaled so that the sign represents the direction of joint movement.

The proposed algorithm assumes  $\tau_k$  is the number of samples between each input change and the beginning of the associated directional (or velocity) change in the output response. It is estimated using discrete-time open-loop experimental data. An illustrative experiment for one joint of a HYDROLEK-7W manipulator in the laboratory [14] is shown in Fig. 1. Here, the pseudo-random input signal was designed to encompass various step changes in magnitude, activated at random intervals in time, some close to the manipulator dead-band limits of  $\approx \pm 1.2$ , in addition to some larger magnitude step changes up to the saturation limits of  $\approx \pm 2$ .

The proposed algorithm estimates the time delay  $\tau_k$  directly from the input and output signals without reference to equation (1). An outline of the approach is given by Algorithm 1. It operates on the input-output data set,  $\mathbf{u} = [u_1, u_2, \dots, u_N]$  and  $\mathbf{y} = [y_1, y_2, \dots, y_N]$ , where  $N$  is the total number of samples in the experiment.

### A. Identification of Input Step Indices

The first stage of Algorithm 1 is to determine the subset of sample numbers  $p_i$  ( $i = 1, \dots, n$ ) at which the input changes, where  $n$  is the total number of input changes that occur within an experiment. This is illustrated by Fig. 2 where, for example, the second and third step changes in the input occur at samples  $p_2 = 89$  and  $p_3 = 116$ , respectively. In practice, the elements

---

### Algorithm 1: Estimation of time-varying delay $\tau_k$

---

**Input:**  $\mathbf{u} = [u_1, u_2, \dots, u_N]$ ,  $\mathbf{y} = [y_1, y_2, \dots, y_N]$

#### Identification of Input Step Indices

$i = 1$

**for**  $k \leftarrow 2$  **to**  $N$  **do**

**if**  $u_k \neq u_{k-1}$  **then**

$p_i = k$

$i = i + 1$

**end**

**end**

**Result:**  $\mathbf{p} = [p_1, p_2, \dots, p_n]$

#### Output Data Pre-Processing

**Result:**  $\mathbf{y}_s = \text{low-pass filter}(\mathbf{y}, \omega_{pb})$

#### Estimation of Time Delays

**for**  $i \leftarrow 1$  **to**  $n$  **do**

**if**  $q_{i-1}$  is known **then**

        Estimate pairs of regression lines within

        segment  $\bar{\phi}_i = [q_{i-1}, q_{i-1} + 1, \dots, p_{i+1} + \alpha]$ ,

        where user coefficient  $\alpha \leq \tau_{min}$

**else**

        Estimate pairs of regression lines within

        segment

$\bar{\phi}_i = [p_i - \alpha, p_i - \alpha + 1, \dots, p_{i+1} + \alpha]$

**end**

**end**

Reduce data resolution in such a way that pairs of distributed data points remain to estimate candidate regression lines. The resolution is defined by user coefficient  $\beta$  and is achieved by the  $\beta \times M_i$  matrix  $\Phi_i$  in equation (4).

**for**  $f \leftarrow 1$  **to**  $M_i - 1$  **do**

    Calculate candidate pairs of regression lines of the form  $\gamma_{i,f,1} = m_{i,f,1}\bar{\phi}_{i,1} + b_{i,f,1}$  and

$\gamma_{i,f,2} = m_{i,f,2}\bar{\phi}_{i,2} + b_{i,f,2}$ , where

$\bar{\phi}_{i,1} = [\phi_{i,1}, \phi_{i,2}, \dots, \phi_{i,f\beta}]$  and

$\bar{\phi}_{i,2} = [\phi_{i,f\beta+1}, \phi_{i,f\beta+2}, \dots, \phi_{i,M_i\beta}]$

The pairs of lines are concatenated to produce

$\gamma_{i,f} = [\gamma_{i,f,1}, \gamma_{i,f,2}]$

Find the Sum of Squared Errors (SSE) of each set of lines.

**end**

Find best fit pair of regression lines i.e.  $f_{opt}$  is the value of  $f$  that yields the lowest SSE.

Determine  $j_{opt} = f_{opt}\beta$

Identify  $q_i = r_i + j_{opt}$  and hence  $\delta_i = q_i - p_i$ .

Evaluate result against conditions.

**end**

**Result:**  $\mathbf{q} = [q_1, q_2, \dots, q_n]$ ,  $\boldsymbol{\delta} = [\delta_1, \delta_2, \dots, \delta_n]$ ,

$\mathbf{m} = [m_1, m_2, \dots, m_n]$

**Variable time delay:**  $[\tau_1, \tau_2, \dots, \tau_N]$  for samples

$k = 1 \rightarrow N$  are obtained by expansion of  $\boldsymbol{\delta}$  using  $\mathbf{p}$ .

---

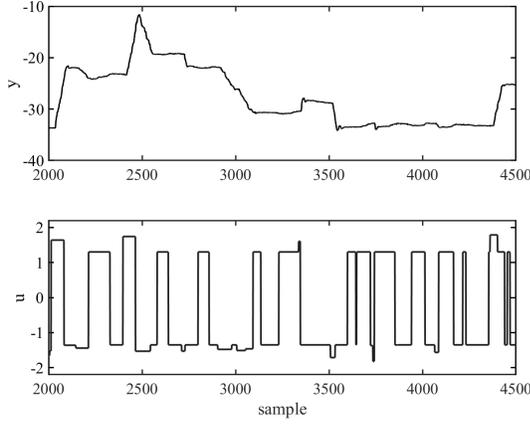


Fig. 1. HYDROLEK-7W experimental data, showing the joint angle (upper subplot) and open loop input (lower) plotted against sample number (0.01 s).

of  $\mathbf{p} = [p_1, p_2, \dots, p_n]$  are found via a straightforward for-loop over the data set e.g. using MATLAB. In the pseudo-code shown by Algorithm 1, a counter  $i$  is used as an index to save a given sample number to the array.

### B. Output Data Pre-Processing

The output data are pre-processed to reduce noise. This is achieved by applying a low-pass filter to the response  $y$ , with user defined passband frequency  $\omega_{pb}$  obtained via trial and error experimentation. The smoothed response is denoted  $y_s$ .

### C. Estimation of Time Delays: Overview

The next and main part of Algorithm 1 is a loop for  $i = 1 \rightarrow n$  to estimate the time delays. The first stage of this loop is to extract a segment of data from which to estimate the delay associated with the input change that occurred at sample  $p_i$ . Each segment of the response is approximated by a pair of linear regression lines, illustrated by the thick solid traces in Fig. 2. Each segment is defined so as to encompass the delayed output response associated with a particular change in the input, whilst avoiding the inclusion of any directional output changes associated with the previous, or the following, input change. In Fig. 2, the vertical dashed lines at samples 79 and 126 define the borders of one such segment (details explained in the following II-D). For each segment, a set of candidate regression line pairs are fitted, and the pair with the best fit in terms of the Sum of Squared Errors (SSE) is identified.

The sample number associated with the switch from the first to the second of the pair of optimised regression lines (the pair with the lowest SSE) is denoted  $q_i$ . This is assumed to be the sample at which the output starts to respond to the input change. For example, in Fig. 2, the input signal drops to a value close to zero at sample  $p_2 = 89$  and, following successful application of the algorithm, sample  $q_2 = 104$  is identified as the moment when the output has started to respond to this change. Hence, the time delay associated with this part of the experiment is  $\delta_2 = 15$  samples.

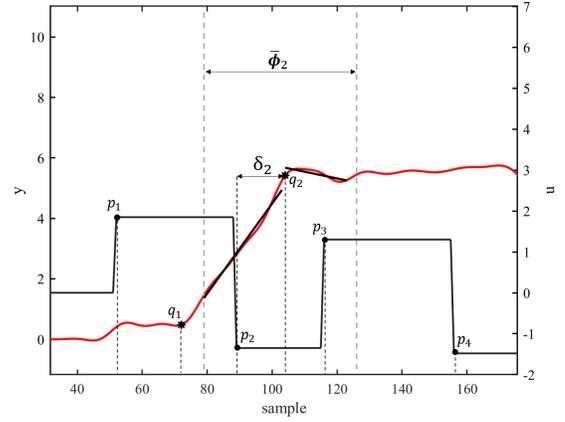


Fig. 2. Diagram showing how a pair of regression lines is fitted to one segment of data. The square-wave-like signal is the input,  $u_k$ , whilst the response,  $y_k$ , is represented by the red trace. The two black lines overlaying the response are illustrative regression lines estimated in the range  $\bar{\phi}_2$ . Here,  $p_1$  to  $p_4$  represent sample numbers at which the input signal changes. For the case of  $p_1$  and  $p_2$ , the estimated sample numbers at which the output starts to respond are denoted  $q_1$  and  $q_2$ .

To summarise,  $p_i$  and  $q_i$  represent the known sample number for a step change in the input and the *estimated* sample number when the output starts to respond, respectively. Hence,  $\delta_i = q_i - p_i$  ( $i = 1, \dots, n$ ) is the time delay associated with that particular input change.

### D. Estimation of Time Delays: Segments

As shown by Algorithm 1, the time delay estimation loop starts by defining a list of sample numbers spanning the segment associated with the  $i$ -th change in the input, i.e.

$$\bar{\phi}_i = [r_i, r_i + 1, r_i + 2, \dots, p_{i+1}, p_{i+1} + 1, \dots, p_{i+1} + \alpha] \quad (2)$$

where  $r_i$  and  $p_{i+1} + \alpha$  are the first and final sample numbers of the  $i$ -th segment of data over which a pair of regression lines will be fitted, respectively, and  $\alpha$  is a user defined coefficient. There are two options for the first element,  $r_i$ , expressed using an if statement in Algorithm 1, as follows.

If the previous estimate of time delay is known,  $q_{i-1}$  is recorded and can be used to define the start of the segment i.e.  $r_i = q_{i-1}$ . In other words,  $r_i$  is defined as the moment (sample number) that the time delay associated with the previous segment has worked its way through the system. In the case of Fig. 2,  $q_1 = 71$  and so choice of  $r_2 = 71$  would ensure that the time delay estimate for the second segment is not effected by the response of the first, whilst maximising the quantity of data used to estimate the associated regression line.

However, if the previous estimate of the time delay does not exist, either because it is the first segment of the series, or because the algorithm failed to converge to a satisfactory solution, then  $q_{i-1}$  is not available and  $r_i = p_i - \alpha$  is used as the first element of  $\bar{\phi}_i$  instead. Fig. 2 illustrates the concept for the case that  $p_2 = 89$  and user defined  $\alpha = 10$ . Hence,

$r_2 = p_2 - \alpha = 79$  is identified as the start of the segment, as indicated by the vertical dotted line over sample number 79.

For both the above options, the final sample number of  $\bar{\phi}_i$  in equation (2) is defined as  $p_{i+1} + \alpha$ . Again,  $\alpha$  is the user-defined variable chosen to ensure that any previous or subsequent directional output changes are not included in the current segment. In Fig. 2,  $p_3 = 116$  and user chosen  $\alpha = 10$ , hence the final element of  $\bar{\phi}_2$  is sample number  $p_3 + \alpha = 126$ , as indicated by the vertical dotted line over sample number 126. The second segment is, therefore, defined as follows,

$$\bar{\phi}_2 = [79, 80, \dots, 116, 117, \dots, 126] \quad (3)$$

Normally,  $\alpha \leq \tau_{min}$  where  $\tau_{min}$  is an assumed minimum time delay for the system. Hence, if the minimum time delay takes a very low value, the associated small value of  $\alpha$  will ensure relatively few samples of data within the segment and may yield poor estimates of  $\delta_i$ .

### E. Estimation of Time Delays: Regression Lines

Candidate pairs of regression lines are fitted to the output response associated with each segment, i.e.  $y_k$  for  $k = \bar{\phi}_i$ . Referring to equation (2), the first and second regression lines of each pair are fitted to  $y_k$  over samples  $r_i \rightarrow r_i + j - 1$  and  $r_i + j \rightarrow p_{i+1} + \alpha$ , respectively. Here,  $j > 1$  represents the switch from one of the regression lines to the other and its optimised value will be used to determine the time delay. If the regression lines provide a satisfactory fit to the output data, then  $q_i = r_i + j$  is the sample at which the output has started to change direction or velocity.

Operating at its highest resolution, the algorithm estimates pairs of regression lines in a loop for  $j = 2, 4, 6, \dots, p_{i+1} + \alpha - r_i - 1$ . Again referring to the second segment  $\bar{\phi}_2$  of Fig. 2, equation (3) shows that the first pair of lines would be for samples 79–80 and 81–126, the second 79–82 and 83–126, and so on through to 79–124 and 125–126, i.e. 23 pairs in total. In practice, promising results are obtained using the algorithm with a reduced resolution and hence reduced computational time. Another user coefficient  $\beta \geq 2$ , representing the number of data points in the shortest regression line under consideration, is introduced to define the new resolution.

For coding purposes, and to identify the data points for which to calculate the regression lines, the first  $M_i\beta$  elements of  $\bar{\phi}_i = [\phi_{i,1}, \phi_{i,2}, \dots, \phi_{i,p_{i+1} + \alpha - r_i + 1}]$  from equation (2) are rearranged into a matrix,  $\Phi_i$ , of dimensions  $\beta \times M_i$ ,

$$\Phi_i = \begin{bmatrix} \phi_{i,1} & \phi_{i,2} & \dots & \phi_{i,\beta} \\ \phi_{i,\beta+1} & \phi_{i,\beta+2} & \dots & \phi_{i,2\beta} \\ \phi_{i,2\beta+1} & \phi_{i,2\beta+2} & \dots & \phi_{i,3\beta} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{i,(M_i-1)\beta+1} & \phi_{i,(M_i-1)\beta+2} & \dots & \phi_{i,M_i\beta} \end{bmatrix} \quad (4)$$

where,

$$M_i = \text{floor} \left\{ \frac{p_{i+1} + \alpha - r_i + 1}{\beta} \right\} \quad (5)$$

Here,  $p_{i+1} + \alpha - r_i + 1$  represents the number of elements in  $\bar{\phi}_i$  from equation (2). Floor is used to ensure an integer value, with the omitted data at the end of the segment generally found to have a negligible impact on the results.

Having optionally reduced the data resolution via choice of  $\beta$ , the next step of Algorithm 1 is the time delay estimation loop for  $j = \beta, 2\beta, 3\beta, \dots, (M_i - 1)\beta$ . For each segment  $i$ , standard linear least squares is used to estimate a candidate set of regression line pairs of the form,

$$\gamma_{i,f,1} = m_{i,f,1}\bar{\phi}_{i,1} + b_{i,f,1} \quad (6)$$

and

$$\gamma_{i,f,2} = m_{i,f,2}\bar{\phi}_{i,2} + b_{i,f,2} \quad (7)$$

where  $f$  is a counter that increments by unity for each step in the loop i.e.  $f = 1, 2, \dots, M_i - 1$ , while  $m_{i,f,1}$  and  $b_{i,f,1}$  are the slope and intercept coefficients of the first line, and  $m_{i,f,2}$  and  $b_{i,f,2}$  are similar for the second line. Here,

$$\bar{\phi}_{i,1} = [\phi_{i,1}, \phi_{i,2}, \dots, \phi_{i,f\beta}] \quad (8)$$

and

$$\bar{\phi}_{i,2} = [\phi_{i,f\beta+1}, \phi_{i,f\beta+2}, \dots, \phi_{i,M_i\beta}] \quad (9)$$

For  $f = 1$ , the sample numbers over which to estimate the first line are given by the first row of  $\Phi_i$  and the sample numbers for the second line are formed by a vectorization of the remaining rows i.e. samples  $\phi_{i,1}, \dots, \phi_{i,\beta}$  and  $\phi_{i,\beta+1}, \dots, \phi_{i,M_i\beta}$ , respectively. For  $f = 2$ , the first two rows of  $\Phi_i$  are the sample numbers for the first line and a vectorization of the remaining rows forms the basis of the second line. The loop proceeds through to  $f = M_i - 1$ , for which the regression pair is formed using a vectorization of the first  $M_i - 1$  rows and the final row of  $\Phi_i$ .

Each pair of regression lines are concatenated,

$$\gamma_{i,f} = [\gamma_{i,f,1}, \gamma_{i,f,2}] \quad (10)$$

and the SSE is used as the metric to select the best fit, where the error is based on  $\gamma_{i,f}$  and  $y_k$  for the relevant sample numbers. Define  $f_{opt}$  as the value of  $f$  associated with the regression pair that yields the lowest SSE. This is used to determine  $q_i = r_i + j_{opt}$ , where  $j_{opt} = f_{opt}\beta$  and  $\delta_i = q_i - p_i$  is the time delay associated with the input change  $p_i$ . Finally, it is assumed that  $\tau_k = \delta_i$  for  $k = p_i$  through to  $k = p_{i+1}$ .

### III. LABORATORY EXAMPLE

The laboratory system consists of two HYDROLEK-7W manipulators, each a 6-degrees-of-freedom (6-DOF) articulated arm, with a seventh actuator for the gripper. The manipulators show evidence of various types of nonlinearity, but generally yield an integrating type of response, albeit sometimes poorly damped with oscillations [12–14]. Although such oscillations complicate the time delay estimation process, they appear to be satisfactorily handled via the proposed algorithm. The system has recently been reconfigured, with the latest hardware framework and control software described by [5]. Photos are available at: <http://wp.lancs.ac.uk/cjtaylor/brokk/>.

Again illustrating using the second segment  $\bar{\phi}_2$  of Fig. 2, with  $p_2 = 89$ ,  $p_3 = 116$  and  $\alpha = 10$  as before,  $r_2 = 79$  and the segment consists of  $p_3 + \alpha - r_2 + 1 = 48$  samples: see equation (3). With  $\beta = 2$ , then  $M_2 = 48/2 = 24$ ,

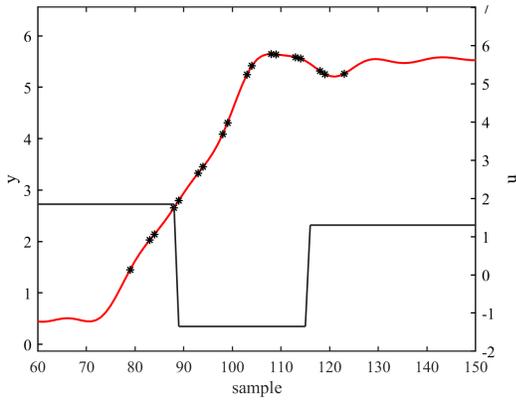


Fig. 3. Example of regression pair sample numbers over one segment, showing the input signal (black, square wave), the smoothed output (red) and the key sample numbers for  $\beta = 5$  (asterisks).

$j = 2, 4, \dots, 46$  and there are 23 iterations in the loop. If instead the resolution  $\beta = 5$ , then  $M_2 = \text{floor}(48/5) = 9$ ,  $j = 5, 10, \dots, 40$  and there are 8 iterations. Focusing on the case with  $\beta = 5$ , equation (4) takes the following form,

$$\Phi_2 = \begin{bmatrix} \phi_{2,1} & \phi_{2,2} & \phi_{2,3} & \phi_{2,4} & \phi_{2,5} \\ \phi_{2,6} & \phi_{2,7} & \phi_{2,8} & \phi_{2,9} & \phi_{2,10} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{2,36} & \phi_{2,37} & \phi_{2,38} & \phi_{2,39} & \phi_{2,40} \\ \phi_{2,41} & \phi_{2,42} & \phi_{2,43} & \phi_{2,44} & \phi_{2,45} \end{bmatrix} \quad (11)$$

in which the numerical values are  $\phi_{2,1} = 79$  through to  $\phi_{2,45} = 123$ . The final three samples of  $\bar{\phi}_2$  are omitted from the matrix  $\Phi_2$ . As discussed above, this array provides a convenient framework to estimate the pairs of regression lines. In the case of Fig. 2,  $f_{opt} = 5$  yields the lowest SSE, hence  $j_{opt} = 25$ ,  $q_2 = 104$  and  $\delta_2 = 15$  samples, as previously noted. Finally, it is assumed that  $\tau_k = 15$  for  $k = 89 \rightarrow 116$  i.e. the time delay of this segment. Alternatively, a linear (or other) interpolation could be used to connect the estimates of time delay for each segment.

For the same segment, the beginning and end points of the candidate regression pairs for  $\beta = 5$  are plotted in Fig. 3, while Fig. 4 exemplifies a regression pair fitted to part of an experiment in which the input signal moves across the dead-band. Here, the manipulator is moving during the first part of the experiment but slows almost to a stop once the input is close to the dead-band. Finally, although illustrative results have been omitted for brevity, Algorithm 1 has been successfully applied to data such as Fig. 1, in order to provide a series of time delay estimates over longer experimental series.

#### IV. DISCUSSION

Previous research has investigated a state dependant parameter (SDP) dynamic model for joint angle, which is based on the relationship between angular velocity and voltage [12–14]. In this regard, Fig. 5 shows how the estimated time delay can be plotted against, for example, the manipulator angular

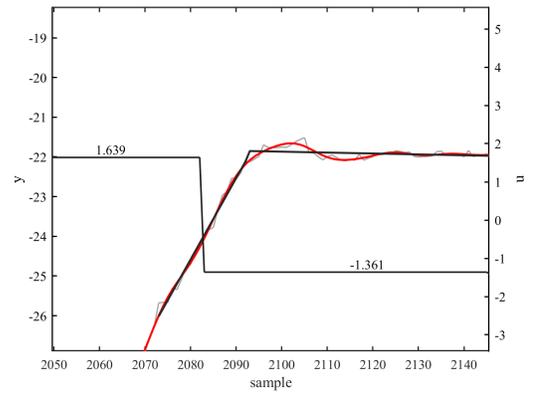


Fig. 4. Example of the Algorithm 1 applied to part of an experiment in which the input changes to a value close to the dead-band, showing the input signal (thick black trace), experimental data (grey), smoothed output (red) and optimised regression line pairs (black traces overlaying the output signal).

velocity, in order to highlight potential state-dependencies. Unfortunately, the example given by Fig. 5 is inconclusive. However, upon accruing more data via further experimental work, the present authors aim to identify the dependency of the time delay on measured variables or combinations of these e.g. angular velocity and hydraulic pressure.

In this context, the new algorithm helps to clarify the types of experiments that will be needed. These should be implemented with a predefined, finite array of randomly chosen input values, which are repeatedly used within one experiment, allowing for deeper analysis into the system's time delay inconsistencies. Identifying a correlation between the delay and other variables will give insight into whether the time delay can be predicted, or whether an online identification method is required for control. For example, [7] develop a multi-model controller, which uses multiple partial models to predict a range of responses for a range of possible time delays. The time delay is subsequently estimated by comparing the predictions to the current measured response. Reference [16] builds on this idea in the context of time delay variations.

Fig. 5 also demonstrates the significant *range* of time delays for the manipulators under consideration here. The estimated time delay varies from 1 sample to 80 samples, equivalently 0.01 s to 0.8 s, a comparatively very large range that is ignored by the existing SDP models based on time-invariant delays. Although beyond the scope of the present article, which focuses on the new time delay estimation routine *per se*, the authors are investigating the development of improved SDP models and control systems to account for these results.

Finally, analysis of the laboratory results reported above have facilitated development of several algorithmic conditions for which the delay might not be recorded (in which case a NaN variable is used in MATLAB):

- 1) The input being considered must not be the first or last in the experiment. The first and last input changes generally result in a comparatively poor evaluation of the response.
- 2) The time delay will not be recorded if any portion of the

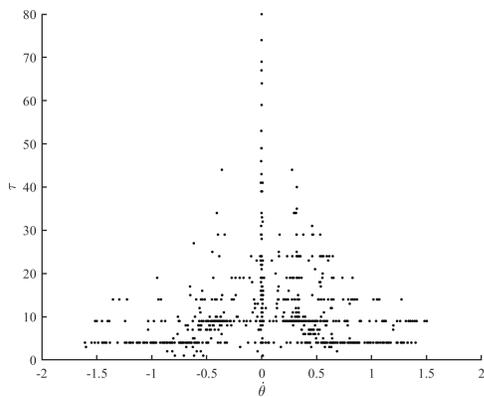


Fig. 5. Estimated time delay plotted against average angular velocity  $\dot{\theta}$ .

output data in the segment surpasses or reaches system limits, such as joint angle hardware limits.

- 3) The time delay will not be recorded if it is estimated as a sample that exists prior to  $p_i$ , i.e. if  $q_i < p_i$ .
- 4) There must be a qualitatively sufficient minimum difference in slope for the optimised pair of regression lines.
- 5) A time delay value may not be recorded if the number of samples between  $p_i$  and  $p_{i+1}$  is shorter than  $\tau_{min}$ .
- 6) Amongst the SSE values calculated for a set of candidate regression lines, the lowest SSE must be less than a threshold value.

## V. CONCLUSIONS

An algorithm to investigate nonlinear systems with time delay variability has been developed. It automatically cycles through segments of an open loop experiment, capturing the directional change in the output via a pair of regression equations, and estimates the time delay accordingly. In this manner, the proposed algorithm is used to uncover the time delay range, potential state-dependencies, and to identify what methods are most appropriate for subsequent system identification and control. The new approach is not limited by a requirement for steady state initial conditions and linear dynamics, nor for the assumed fixed delay implicit in previously identified nonlinear SDP models. However, its development has been motivated by systems with approximately integrating behaviour, such as for hydraulically actuated robotic manipulators, as discussed in the laboratory example. The method has been successfully used to estimate the wide range of delays observed in the experimental data, and is providing insight into the variable time delay behaviour. In future work, the authors are developing new SDP models and control systems to exploit these results [16].

## REFERENCES

- [1] J. P. Richard. "Time-delay systems: An overview of some recent advances and open problems". In: *Automatica* 39.10 (2003), pp. 1667–1694.
- [2] J. E. Normey-Rico and E. F. Camacho. "Dead-time compensators: A survey". In: *Control Engineering Practice* 16 (2008), pp. 407–428.
- [3] C. I. Aldana, R. Munguía, E. Cruz-Zavala, and E. Nuño. "Pose consensus of multiple robots with time-delays using NN". In: *Robotica* 37 (2019), pp. 883–905.
- [4] Y. Mao and H. Zhang. "Exponential stability and robust H-inf... non-linear systems with time-varying delays..." In: *Int. J. Systems Science* 45 (2014), pp. 1112–1127.
- [5] O. L. R. Albrecht and C. J. Taylor. "Unknown and time-varying time delays in the modelling and control of hydraulic actuators: literature review". In: *4th Australian and New Zealand Control Conf. Gold Coast*, 2020.
- [6] Y. Tan. "Time-varying time-delay estimation for nonlinear systems using neural networks". In: *Int. J. Applied Math and Computer Science* 14.1 (2004), pp. 63–68.
- [7] S. Ben Atia, A. Messaoud, and R. Ben Abdennour. "An online identification algorithm of unknown time-varying delay and internal..." In: *Mathematical and Computer Modelling of Dynamical Systems* 24 (2017), pp. 26–43.
- [8] Y. Zhao, A. Fatehi, and B. Huang. "A data-driven hybrid ARX and Markov Chain modeling approach to process ident. with time-varying time delays". In: *IEEE Trans. Industrial Elect.* 64 (2017), pp. 4226–4236.
- [9] Y. Ma, S. Kwak, L. Fan, and B. Huang. "A variational Bayesian approach to modelling with random time-varying time delays". In: *American Control Conference*. 2018, pp. 5914–5919.
- [10] Z. Trajanoski and P. Wach. "Neural predictive controller for insulin delivery using the subcutaneous route". In: *IEEE Trans. Biomed. Eng.* 45 (1998), pp. 1122–1134.
- [11] X. M. Ren and A. B. Rad. "Identification of nonlinear systems with unknown time delay based on time-delay neural networks". In: *IEEE Trans. Neural Networks* 18 (2007), pp. 1536–1541.
- [12] C. J. Taylor and D. Robertson. "State-dependent control of a hydraulically-actuated nuclear decommissioning robot". In: *Cont. Eng. Pract.* 21 (2013), pp. 1716–1725.
- [13] C. West, E. D. Wilson, Q. Clairon, S. Monk, A. Montazeri, and C. J. Taylor. "State-dependent parameter model identification for inverse dead-zone control of a hydraulic manipulator". In: *18th IFAC Symposium on System Identification (SYSID)*. Stockholm, 2018.
- [14] M. Bandala, C. West, S. Monk, A. Montazeri, and C. J. Taylor. "Vision-based assisted tele-operation of a dual-arm hydraulically actuated robot for pipe cutting and grasping in nuclear environments". In: *Robotics* 8.6: 42 (2019), pp. 1–24.
- [15] C. J. Taylor, P. C. Young, W. Tych, and E. D. Wilson. "New developments in the CAPTAIN Toolbox for Matlab with case study examples". In: *18th IFAC Symp. System Identification*. Stockholm, 2018.
- [16] O. L. R. Albrecht and C. J. Taylor. "Pole assignment control design for time-varying time-delay systems using radial basis functions". In: *13th UK Automatic Control Council (UKACC) International Conference*. Plymouth, Apr. 2022.