

Adaptive Control of an Integrator with Unknown Dead-Time

Magnus Dagbro



LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6030
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

© 2017 by Magnus Dagbro. All rights reserved.
Printed in Sweden by Tryckeriet i E-huset
Lund 2017

Abstract

An adaptive control system is proposed for an unknown integrator process with unknown dead-time. The process dead-time and gain are estimated simultaneously using an extended regression model and recursive least-squares estimation. The parameter estimates are used to scale the actuator signal, transforming the unknown process into a known reference process, while the dead-time estimate provides dead-time compensation via state prediction.

Acknowledgements

Thank you to my advisor Dr. Rolf Johansson at the Dept. of Automatic Control of Lund University's Faculty of Engineering, and to Ashraf Zarur, Anders Rundberg and Mattias Darmell, my industry advisors at Tetra Pak. Also a big thank you to Magnus Meijer at Tetra Pak for helping me adapt my controller into a PLC program, and for preparing and running the machine during testing.

Thank you to my friends and colleagues Daniel Nilsson and Christoffer Schou Kongstad for allowing me to bounce ideas and for providing valuable insights from their own areas of expertise.

Finally, thank you to my examiner Dr. Karl Erik Årzén at the Dept. of Automatic Control.

Contents

1. Introduction	9
1.1 Background	9
1.2 Problem Formulation	10
1.3 Delimitations	10
1.4 Methodology	10
1.5 Literature	11
1.6 Notation	11
1.7 Outline	11
2. Modeling	13
2.1 Process model	13
2.2 Simulation model	14
2.3 Actuator model	14
3. Combined Parameter and Dead-Time Estimator	16
3.1 Linear regression	16
3.2 Adaptive Estimator for Integrator Dead-Time	17
4. Controller	20
4.1 Structure of the Adaptive Control System	20
4.2 The PID Controller	20
4.3 Delay Compensation	22
4.4 Complete Controller	23
5. Simulation and Testing	25
5.1 Simulation	25
5.2 Real-World Tests	28
6. Conclusion	31
7. Discussion	32
7.1 Future Work	33
7.2 Summary	33
Bibliography	34

1

Introduction

1.1 Background

Tetra Pak is the world's largest supplier of packaging solutions for food and drink products in carton packages. Tetra Pak have requested an adaptive control system be designed for an integrator process where unknown and time-variable gain and dead-time causes problems in tuning new versions of the system.

Tetra Pak currently uses empirically tuned PID controllers to control these processes. The actuator providing input to the integrator process is non-linear and time-invariant with dead-time, but any given machine will only work within a narrow range of the nonlinear actuator, where a working point linearisation gives sufficient performance. Tetra Pak's current approach is to have an engineer manually tune all parameters of the PID controller with every new hardware change, until the closed-loop system achieves sufficient performance around the new linearisation point. Currently no effort is made to try and compensate for the significant dead-times found in the process; instead the PID parameters are chosen sufficiently slow to keep the system stable. Even so, process changes often occur over time, eventually bringing these systems out of stability.

Tetra Pak now wants to streamline this process, so that instead of manually tuning a PID loop, the machine commissioner inputs a few basic system parameters into an adaptive controller that is then able to identify and control the process. The proposed adaptive controller uses the given system parameters as starting point for a recursive least squares estimator, estimating a linearisation of the process gain and dead-time. The gain estimate is used to scale the control signal, transforming the process into a reference system with previously tuned PID parameters. The dead-time estimate is used to provide a state prediction based on past control inputs.

1.2 Problem Formulation

The system to control is a simple inflow-outflow integrator process with an unknown integrator gain, unknown dead-time, and process changes to both gain and dead-time. The outflow is governed by the (constant) machine speed and the inflow via a non-linear actuator. The inflow actuator is, however, only working in a small part of its entire range, allowing a simple linearisation to give adequate performance. If the linearised process is known, the control signal can be scaled and biased to transform the system into any given integrator. Using this approach, only one tuning is necessary across all machines.

Thus the goal of this thesis is to design an estimator for the actuator gain and dead-time, and to also design a PID controller that is able to accept the identified parameters and update its response such that the pair form an adaptive control loop. The control system should be capable of self-tuning to a new machine as well as adapting to process changes that may occur over time.

1.3 Delimitations

Due to the small working range of the input actuator, actuators are assumed to be linear aside from their quantization nonlinearity.

The process estimator will be designed mainly for an integrator process and not as a general gain and dead-time estimator between any two linearly correlated and delayed signals.

1.4 Methodology

In the first step, a simulation model is created to allow testing of the controller without using a real machine. The model parameters should be closely matched to a typical machine and a reasonable non-linearity added at the input. Tetra Pak has kindly provided measurements and control data from a range of different machines to ease the modeling process.

With the simulation model done, a reference process is created as a standard integrator with unity gain on the input. A PID controller is then tuned to the desired response on the reference model. The PID controller is then modified to accept inputs with knowledge of the process, allowing any integrator process to be controlled using the reference tuning. A process estimator is then designed capable of extracting the required knowledge from a similar process, and the output of this estimator sent to the 'knowledge' input of the PID controller.

The completed adaptive control system is tested first using the simulation model, and finally tests are performed on a real Tetra Pak machine.

1.5 Literature

The general structure of the proposed dead-time estimator was based on the overparameterised dead-time estimator by Kurz and Goedecke (1981). Their approach used a least-squares estimator with a regressor made up of all possible delayed inputs, of which the index of the largest coefficient is used as a basis for the delay estimate.

The adaptive reduced-order overparameterised dead-time estimator was derived from the adaptive dead-time estimator designed by De Keyser (1986).

For adaptive control methods and the structure of the least-squares estimator, Rolf Johansson's course book on adaptive control was used (2013). General best practices in adaptive control were gained from Åström and Wittenmark's *Adaptive Control* (2008). Finally, valuable insights into Digital Signal Processing, sampling and more were found in Tan and Jiang's academic literature on DSP. (2014)

1.6 Notation

Throughout the thesis, time derivatives are expressed either using Leibniz's notation as $dx(t)/dt$ or Newton's dot notation as \dot{x} .

In the Laplace frequency domain, s is used to express complex frequency and $F(s)$ to denote Laplace domain functions of the complex frequency s (e.g. block transfer functions).

For discrete-time difference equations the subscript notation is used, where x_k denotes the value of x at the discrete time k .

Estimated values are denoted with the addition of a 'hat' symbol (as in \hat{x}), while a minor revision or change of the value x will be denoted with an added 'prim' symbol (as in x').

1.7 Outline

Chapter 2: Modeling

This chapter describes the process and simulation models created for the thesis.

Chapter 3: Combined Parameter and Dead-Time Estimator

The theory behind the estimator is described in Chapter 3.

Chapter 4: Controller

In this chapter, the modified PID controller for adaptive systems is described.

Chapter 5: Simulation and Testing

This chapter presents data from simulations and real-world testing.

Chapter 6: Conclusions

A summary of results and conclusions drawn from the work.

Chapter 7: Discussion

Discussion of results and conclusions.

2

Modeling

2.1 Process model

The process is modeled as an inflow-outflow integrator with the storage level x . The rate of change of the storage variable x is given by the difference between inflow rate and outflow rate - if the inflow is greater than the outflow, the stored level will rise with a rate proportional to the difference in flows. This dynamic is described using the following expression:

$$\frac{dx(t)}{dt} = \dot{x}(t) = b[q_i(t) - q_o] \quad (2.1)$$

Where $dx(t)/dt$ or $\dot{x}(t)$ is the rate of change of the storage level. $q_i(t)$ is the (controlled) rate of inflow and q_o the (constant) rate of outflow. b is the proportionality coefficient relating the input flow to the storage level change rate.

By feedforward of the outflow to the inflow, the system can be seen as a single-input single-output (SISO) integrator in the input signal $u(t)$.

$$q_i(t) = u(t) + q_o \quad (2.2)$$

\Leftrightarrow

$$\dot{x}(t) = b \cdot u(t) \quad (2.3)$$

Now, by scaling $u(t)$ with a chosen coefficient, any SISO integrator can be seen as any other SISO integrator.

$$u(t) = \frac{c}{b} \cdot u'(t) \quad (2.4)$$

\Leftrightarrow

$$\dot{x}(t) = c \cdot u'(t) \quad (2.5)$$

2.2 Simulation model

A simulation model was created in Simulink based on the integrator process model with inflow and outflow rates, with added white noise disturbances and with measurement sampling and quantisation similar to what is in a real process. A Simulink block diagram of the model is shown in Fig. 2.1.

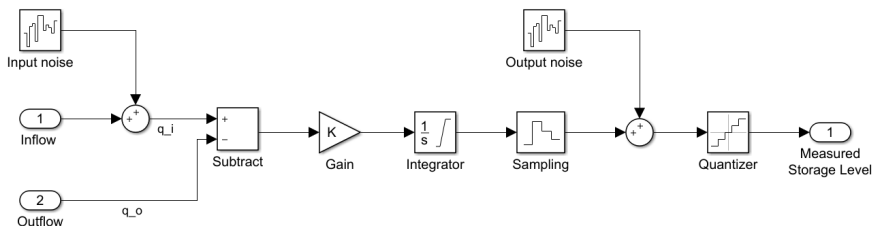


Figure 2.1 The simulation model created for the process

2.3 Actuator model

The non-linear actuator has a "well-behaved" non-linearity, in the sense that the transfer from actuator value to input flow is time-invariant and continuous in its transfer and in its first derivative. Adding to that, at any given constant production rate, the actuator is working within a small enough range that a simple linearisation is sufficient. Thus the actuator can be adequately modeled as a simple line function. A mockup of the actuator transfer function is shown in Fig. 2.2. The figure also illustrates the limited working areas of two different machines. In this case machine 1 sees a process with higher gain than machine 2 despite using the same actuator.

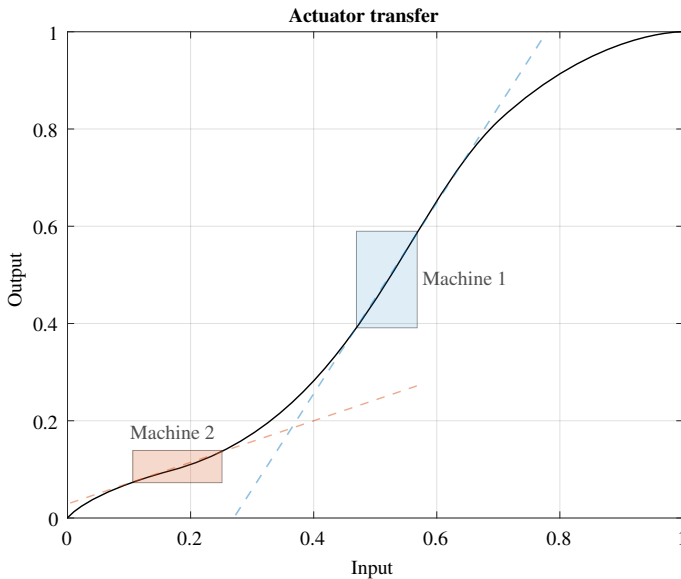


Figure 2.2 Example of an actuator transfer function. Different machines may work in different areas of the non-linear actuator and may thus see very different responses.

Since the actuator is digitally controlled in a number of discrete steps, the actuator also has a quantisation non-linearity. The resolution of the actuator is such that a typical machine may work within approximately 50 to 100 steps of actuation.

The actuator also has an input-output delay, in the range of between 3 and 30 time steps in different machines, and process variations such that the system may over time drift out of stability.

3

Combined Parameter and Dead-Time Estimator

3.1 Linear regression

The idea of control system identification is not new. There have been many contributions to the field, but common to most of them is that they are based around regression analysis. The general idea is that measurements are decomposed into a linear matrix function with known *predictor variables* and unknown *response variables*. For linearly correlated data $y = kx + m$ with a known x , the decomposition looks as follows:

$$y = [x \quad 1] \begin{bmatrix} k \\ m \end{bmatrix} \quad (3.1)$$

where $[x \quad 1]$ is the known predictor and $[k \quad m]^T$ the unknown response. Using this decomposition it is possible to find a best estimate for the response variable from a set of linearly correlated but noisy data. One common method is to find the response variable minimising the sum of squared errors for all data points.

In system identification, the predictors will typically consist of past inputs and outputs of a discrete-time system, and the unknown response variable will contain the system's impulse response coefficients.

On-line Estimation

Recursive techniques exist that enable on-line estimation of the response variable, where only a small part of the estimation problem is solved each time step and new best estimates are available every step. This project uses a recursive Least Squares (RLS) identification technique from Rolf Johansson's course book on adaptive and predictive control. (2013)

Johansson's algorithm is

$$\hat{\theta}_k = \hat{\theta}_{k-1} + P_k \phi_k \varepsilon_k \quad (3.2)$$

$$\varepsilon_k = y_k - \phi_k^T \hat{\theta}_{k-1} \quad (3.3)$$

$$P_k = P_{k-1} - \frac{P_{k-1} \phi_k \phi_k^T P_{k-1}}{1 + \phi_k^T P_{k-1} \phi_k} \quad (3.4)$$

where $\hat{\theta}_k$ is the parameter estimate, ε_k the prediction error and the matrix P_k is an estimate of the parameter covariance at time step k save for a factor σ_k . The initial covariance matrix P_0 is chosen by heuristic, often as an identity matrix multiplied by some large number. A 'large' P_0 may cause large transients in the first steps of estimation while a 'small' matrix generally gives slow but steady initial convergence. Another method is to use regular 'off-line' least-squares estimation for the first few values to provide initial values for P_0 and $\hat{\theta}_0$.

It may sometimes be beneficial to weigh old data lower than recent data to make the system respond to changes in the process dynamics. To allow this, a *forgetting factor* λ can be included in the calculation of the P_k matrix. The 'forgetful' formula for P_k is

$$P_k = \frac{1}{\lambda} \left(P_{k-1} - \frac{P_{k-1} \phi_k \phi_k^T P_{k-1}}{\lambda + \phi_k^T P_{k-1} \phi_k} \right)$$

The number of 'remembered' data points is roughly given by interpreting the forgetting factor λ as a time constant: $1/(1 - \lambda)$ [Johansson, 2013]

3.2 Adaptive Estimator for Integrator Dead-Time

The proposed estimator uses an adaptive extended least-squares model inspired by the dead-time estimators proposed by Kurz and Goedecke (1981) and De Keyser (1986). The discrete-time integrator's impulse transfer function

$$\begin{aligned} x_{k+1} &= x_k + bu_{k-\hat{d}} \\ &\Leftrightarrow \\ x_{k+1} - x_k &= bu_{k-\hat{d}} \end{aligned}$$

is extended with two terms; one with *less* dead-time and one with *more* dead-time than the current dead-time estimate \hat{d} , signified by the coefficients \hat{b}_m and \hat{b}_p .

$$x_{k+1} - x_k = \hat{b}_m u_{k-\hat{d}+1} + \hat{b} u_{k-\hat{d}} + \hat{b}_p u_{k-\hat{d}-1}$$

This can be decomposed into predictor-response format with (known) past control signals and unknown \hat{b} coefficients:

$$x_{k+1} - x_k = \begin{bmatrix} u_{k-\hat{d}+1} & u_{k-\hat{d}} & u_{k-\hat{d}-1} \end{bmatrix} \begin{bmatrix} \hat{b}_m \\ \hat{b} \\ \hat{b}_p \end{bmatrix}$$

If the current dead-time estimate \hat{d} is correct, the largest estimated coefficient will be the "middle" coefficient \hat{b} . If, however, the estimated dead-time is smaller or larger than the actual dead-time, either the \hat{b}_m or the \hat{b}_p coefficient (respectively) will yield a larger estimate than the \hat{b} coefficient. In this case, the dead-time estimate \hat{d} is decreased or increased one step (thus moving the entire dead-time "window" left or right), the estimated coefficients are moved to stay with their respective dead-time estimate, and the covariance matrix is reset as this is essentially a new system. This approach works well for estimating a dead-time of an integer number of samples, but cannot handle a dead-time that is a non-integer number of sample times.

Approximation of Sub-Sample Dead-Time using Coefficient Magnitudes

By empirical observation, a sub-sample dead-time with the pure integrator process appears to "split" the gain estimate over the two closest coefficients. This is seen in the estimate array as a larger coefficient on *one* of the edge coefficients than on the other, seemingly in proportion to the sub-sample delay. A naïve argument can be made that if the true dead-time is in the middle of two integers, the estimated coefficients for the two integer dead-times above and below the true dead-time would be expected to have similar estimated coefficients. Thus a proposed naïve approximation of the sub-sample dead-time is to use a weighted average of the \hat{b}_p and the \hat{b}_m coefficients, divided by the larger "middle" coefficient \hat{b} - This way, if the "plus" estimate and the "middle" estimate are similar in magnitude while the "minus" estimate is (close to) zero, the approximated sub-sample delay will be in the middle of the "plus" dead-time and the "middle" dead-time. Using this approximation, the sub-sample delay estimate is thus:

$$\hat{d}' = \hat{d} + \frac{(1) \cdot \hat{b}_p + (-1) \cdot \hat{b}_m}{\hat{b}}$$

Finding the actuator gain parameter

The "gain" parameter should theoretically be given by the magnitude of the middle coefficient \hat{b} while the coefficients representing less and more dead-time should tend to zero. However, given that a sub-sample dead-time will split the gain estimate over two coefficients (as established in the previous section), the middle coefficient

alone cannot be assumed to give a good estimate of the gain parameter. A better estimation of the gain parameter is in this case given by the *sum* of the three estimated coefficients.

$$\hat{b}' = \hat{b} + \hat{b}_m + \hat{b}_p$$

Decimation of signals

If the estimation block is to be run at a lower sampling rate than the controller block, any signals used in the parameter estimator must be low-pass filtered to attenuate any frequency content above this lower sample frequency - otherwise, the high frequency content may cause aliasing artefacts in the decimated signal such that high-frequency content appears as false images at lower frequencies. [Tan and Jiang, 2014] A filter designed to combat this is known as an *anti-aliasing* (AA) filter and is usually a low-pass filter with a very sharp cut-off at the Nyquist frequency of the new sample rate (or lower).

4

Controller

4.1 Structure of the Adaptive Control System

The adaptive control system is based around the estimator outlined in chapter 3. The estimated gain and dead-time parameters are sent to the controller block which then updates its response accordingly. A block diagram of the entire control system is drawn in Fig. 4.1.

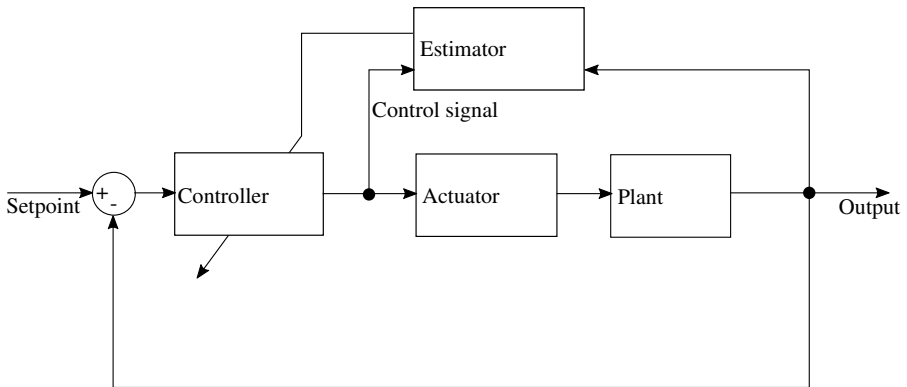


Figure 4.1 Block diagram of the controller structure

4.2 The PID Controller

The integrator process

$$\dot{y}(t) = bu(t) \quad (4.1)$$

can be stabilised using a control signal that is proportional to the error

$$e(t) = y(t) - r(t)$$

where $r(t)$ is the wanted reference value at time t . The error $e(t)$ is fed back to the input using the control law $u(t) = -K_p e(t)$ for some factor K_p . Using this control law the closed-loop system will act like a first-order low pass filter with cut-off frequency $K_p b$. The transfer function from reference to output is then:

$$\frac{Y(s)}{R(s)} = \frac{1}{\frac{s}{K_p b} + 1}$$

If there are static disturbances, an integrator must be added to the control law to eliminate static errors. The integrator however shifts the phase response 90 degrees in the positive direction, which may give poor phase response. Adding a derivative term to lift the phase margin back up, we arrive at the standard PID control law:

$$u(t) = -K_p e(t) - K_i \int_0^t e(\tau) d\tau + K_p \frac{de(t)}{dt} \quad (4.2)$$

Variable PID controller

The proposed *variable* PID controller has an additional input for the estimated input gain \hat{b} seen in equation (4.1). As shown in Chapter 2, any integrator process can be transformed into a unity integrator $\dot{y}(t) = u(t)$ via control signal scaling:

$$\begin{aligned} u'(t) &= \frac{u(t)}{\hat{b}(t)} \\ &\Leftrightarrow \\ u'(t) &= -\frac{K_p e(t)}{\hat{b}(t)} - \frac{K_i}{\hat{b}(t)} \int_0^t e(\tau) d\tau + \frac{K_p}{\hat{b}(t)} \frac{de(t)}{dt} \end{aligned}$$

However, using this scaling on the output of the PID controller in Eq. (4.2) will not give bumpless control when updating the \hat{b} value, because the integrator state would then be scaled directly. Instead a modified control law is applied that scales the *integrand* rather than the integrator state.

$$u(t) = -\frac{K_p e(t)}{\hat{b}(t)} - K_i \int_0^t \frac{e(\tau)}{\hat{b}(\tau)} d\tau + \frac{K_p}{\hat{b}(t)} \frac{de(t)}{dt}$$

Quantisation noise

The actuator's deterministic quantisation noise is dealt with by adding rectangular pseudo-random noise in the range of $[-0.5, 0.5]$ units to the output signal, before rounding to the actuator's discrete values. The dither de-correlates the deterministic noise from the input signal and instead creates a random noise floor. [Blinn, 1994] A comparison between dithered and non-dithered signals is shown in Fig. 4.2

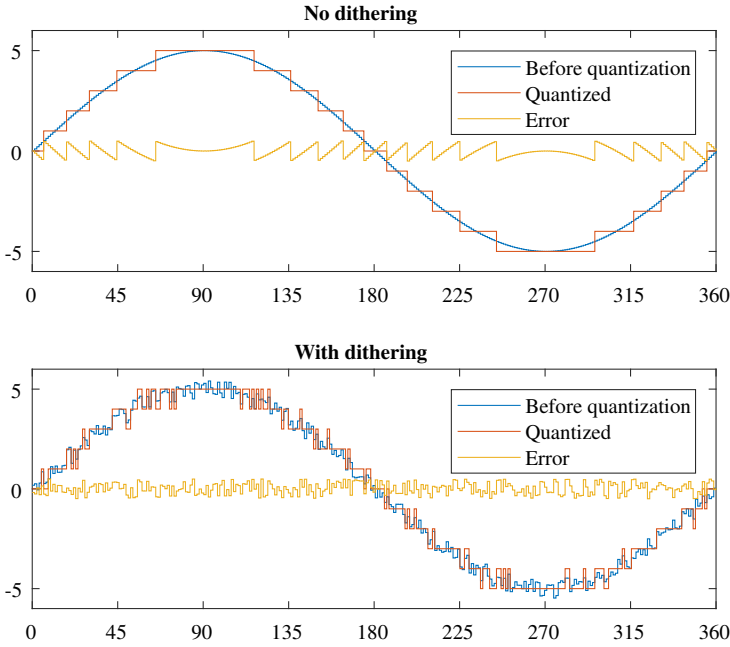


Figure 4.2 Comparison between dithered and non-dithered quantized signals

4.3 Delay Compensation

The input-output dead-time is compensated for with a simple state predictor using past control inputs. For a continuous-time integrator with dead-time d and gain b , the expected output $y(t)$ at time t is

$$y(t) = \int_0^{t-d} b \cdot u(\tau) d\tau \quad (4.3)$$

Thus the result of any control actions given after $t - d$ have not yet reached the output at time t . To avoid acting multiple times on the same disturbance, the controller should act upon a *predicted* error $\hat{e}(t + d)$ rather than the current error $e(t)$.

For the integrator without dead-time, the expected output at time t would be:

$$y(t) = \int_0^t b \cdot u(\tau) d\tau \quad (4.4)$$

The difference between the non-delayed and the delayed system is:

$$\int_0^t b \cdot u(\tau) d\tau - \int_0^{t-d} b \cdot u(\tau) d\tau = \int_{t-d}^t b \cdot u(\tau) d\tau$$

Thus a reasonable prediction of the future state at time $t + d$ is given by adding this difference to the latest measurement $y(t)$.

$$\hat{y}(t + d) = y(t) + \int_{t-d}^t b \cdot u(\tau) d\tau \quad (4.5)$$

Finally the predicted error is given by

$$\hat{e}(t) = \hat{y}(t) - r(t)$$

Discrete Time Predictor

For the discrete-time process sampled with sample time T_s with n steps of delay, the integrated control signal in Eq. (4.5) reduces to a simple sum of past inputs. The predicted non-delayed output is thus:

$$\hat{y}_k = y_k + \frac{1}{T_s} \sum_{i=k-n}^k b \cdot u_i$$

Similarly to the continuous-time case, the predicted error is

$$\hat{e}_k = \hat{y}_k - r_k$$

4.4 Complete Controller

The state predictor provides a prediction of the control error using past inputs, the estimated dead-time \hat{d} and the estimated integrator gain \hat{b} . The PID controller acts on the predicted error and adjusts its response based on the estimated gain \hat{b} . Finally, the control signal is dithered using rectangular noise before it is rounded for quantization. A Simulink block diagram of the complete controller is shown in Fig. 4.3

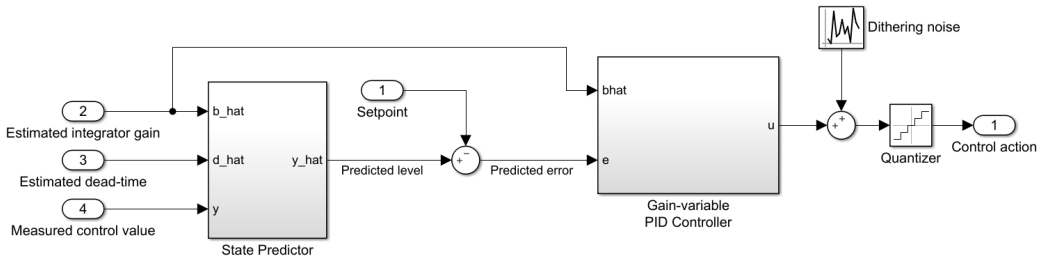


Figure 4.3 The complete PID controller with dead-time and gain compensation

5

Simulation and Testing

5.1 Simulation

This section contains results from the simulated system. The gain and dead-time of the process have been normalised to 1.0 in all plots and units have been omitted in order to protect Tetra Pak's intellectual property. During estimation an acceptable amount (for the real-world system) of white noise was added to the control signal to provide the estimation system with sufficient excitation.

The initial covariance matrix P_0 was chosen as a unit matrix multiplied by a large constant. The forgetting factor λ was chosen to 0.995, giving a memory time constant of approximately 200 samples. The sample time for the estimator was 8 times longer than that of the control loop. A sharp-cutoff anti-aliasing filter was used to stop high frequency content from causing aliasing artefacts in the resampled signal.

Convergence

Adaptive control was started using crude initial guesses for the gain and delay estimates. The gain and delay estimates quickly converged to within 90% of the correct values as seen in Fig. 5.1. The delay estimate in this case converged with a constant error of 1 step size.

Gain disturbance

After an initial period of convergence, the actuator gain was artificially increased from 1.0 to 1.5. After some time the gain was decreased to 0.75. The estimated gain during this experiment is plotted in Fig. 5.2.

Delay disturbance

After convergence the I/O delay was increased with 0.7 units to 1.7. The estimator quickly reacted to the change and converged to the new I/O delay. Fig. 5.3 shows the effect on the gain and delay estimates and Fig. 5.4 the effect on the response

variable $\theta = [\hat{b}_m \hat{b} \hat{b}_p]^T$. Here the \hat{b}_p and \hat{b}_m estimates seem to trade places in the plot while the center \hat{b} estimate stays largely the same. Near the end of the plot there is also a window switch triggered by the \hat{b}_p estimate becoming larger than the center \hat{b} estimate.

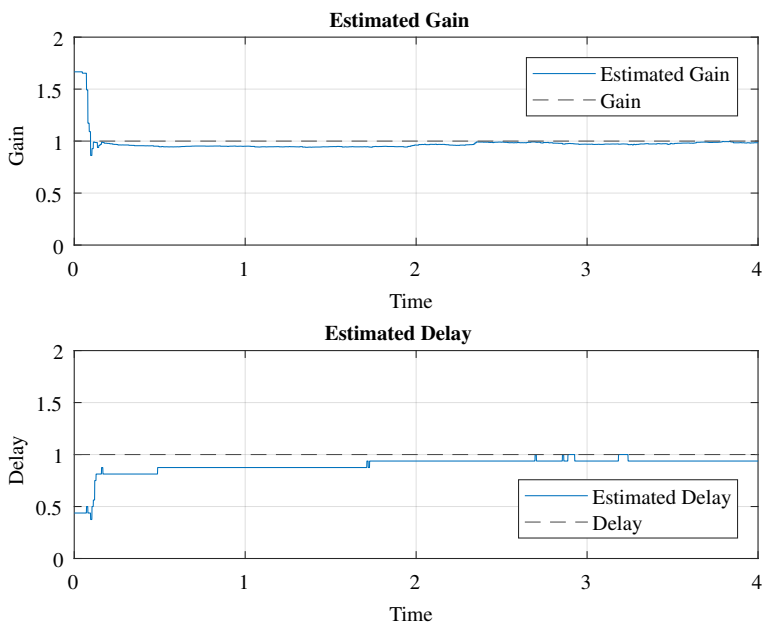


Figure 5.1 Convergence at start of adaptive control

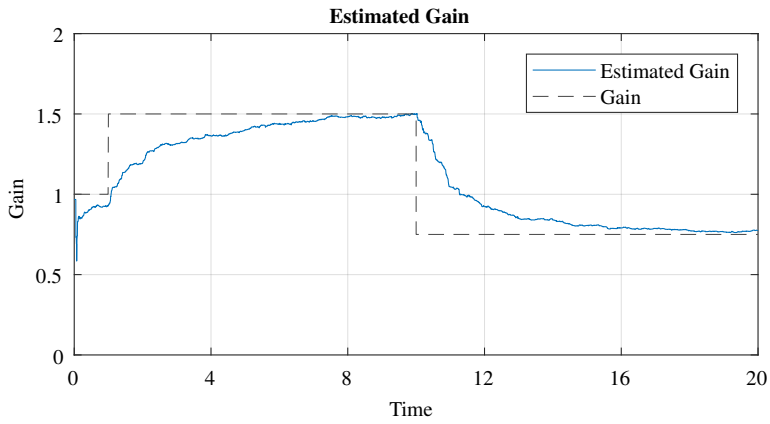


Figure 5.2 Estimated actuator gain with multiplicative gain disturbance

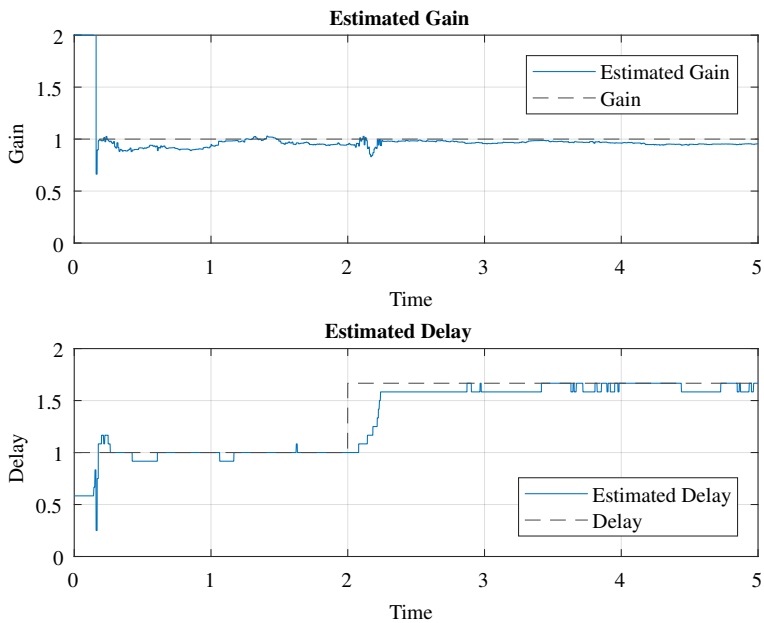


Figure 5.3 Estimation with step delay disturbance at time $t = 2$

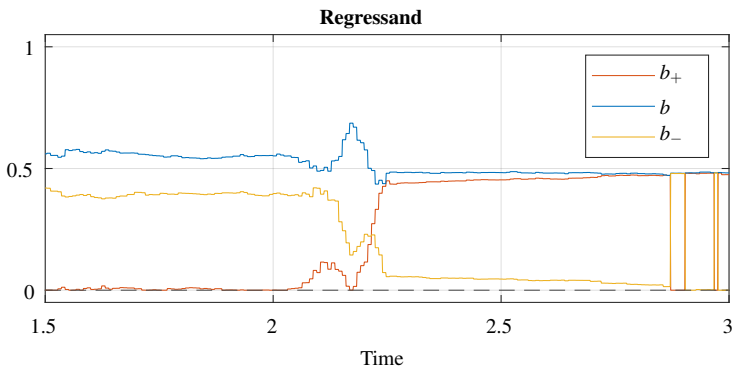


Figure 5.4 Response variables during step delay disturbance

5.2 Real-World Tests

Estimation Convergence and Tracking

The estimator and the controller were both implemented in machine code using Simulink PLC Coder. This is a Matlab toolbox that allows generation of a single Simulink block to a PLC routine with the same inputs and outputs. To generate a single PLC routine for the entire adaptive controller, a subsystem was created containing all of the controller's Simulink blocks and the requisite inputs and outputs.

The generated code was imported to a real machine's PLC and tested in the real world, again using a unit matrix-based covariance matrix P_0 , a forgetting factor λ of 0.995, and an 8 times longer sample time for the estimator than for the control loop. A sharp-cutoff anti-aliasing filter was again used to reduce high frequency content before decimation.

The experiment was performed by starting the process and allowing the adaptive control system to control and estimate the process. The process estimation was allowed to converge for some time, after which an artificial gain disturbance of 1.5x was introduced in software. The gain disturbance was then removed. On introduction of the disturbance, the estimated value started converging toward the new value, and upon removal of the disturbance the estimated gain quickly returned to the first value. Unfortunately it was found post experiment that the gain disturbance had been removed too quickly, as seen by the disturbance being reset before the gain estimation could fully converge to its asymptotic value. The gain and delay estimations during this test are plotted in Fig. 5.5. Note the very quick convergence back to 1.0 as the disturbance was removed. This coincided with a small bump in the

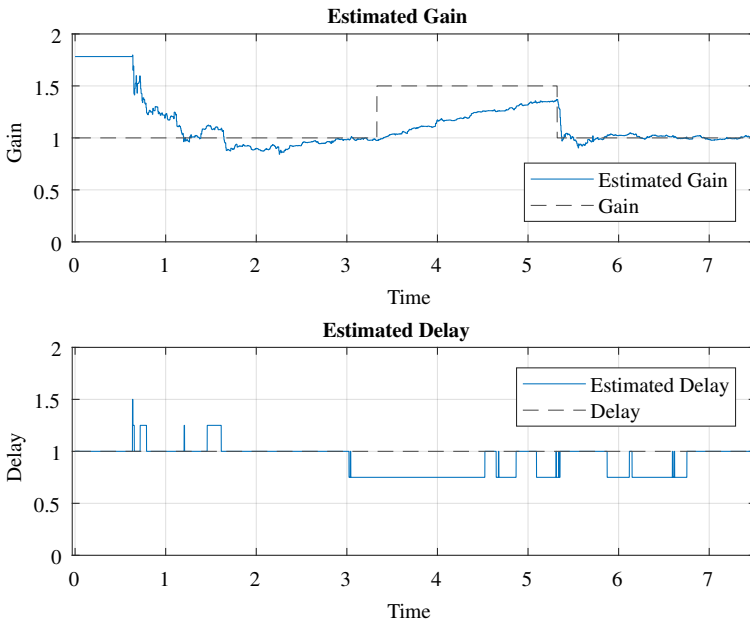


Figure 5.5 Estimated actuator gain and I/O delay in the real machine

controlled level created by the non-bumpless integrator.

Input-Output Comparisons

Comparisons were made between the estimated actuator action u and the measured level's change rate dx/dt . The estimated actuator action was created by scaling and delaying the control signal using the estimator's estimated values for gain and dead-time - thus a 1:1 correspondence is expected between the actuator action and the measured level change rate. A time series plot is shown in Fig 5.6, where values in between measurements (filled circles) have been interpolated using spline interpolation to increase readability. A scatter plot relating estimated input vs measured output is shown in Fig. 5.7. The two plots suggest that the process model is quite good as there is a high degree of correlation between the estimated actuator action and the measured level change rate. The scatter plot of Fig. 5.7 shows that the linear approximation made for the input actuator is very accurate in this machine, since the scatter is nicely linear with no obvious nonlinearity. The apparent 1:1 correlation between output and corrected input also shows that the estimator is doing a good job at estimating the process parameters.

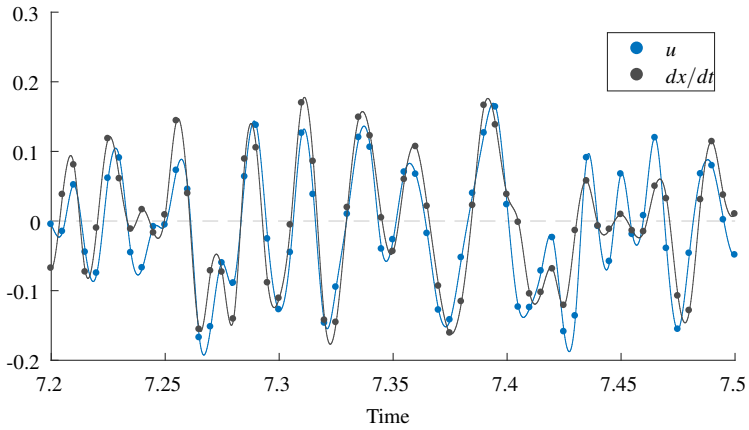


Figure 5.6 Time series comparison between estimated actuator action u and measured storage level derivative dx/dt in the real machine

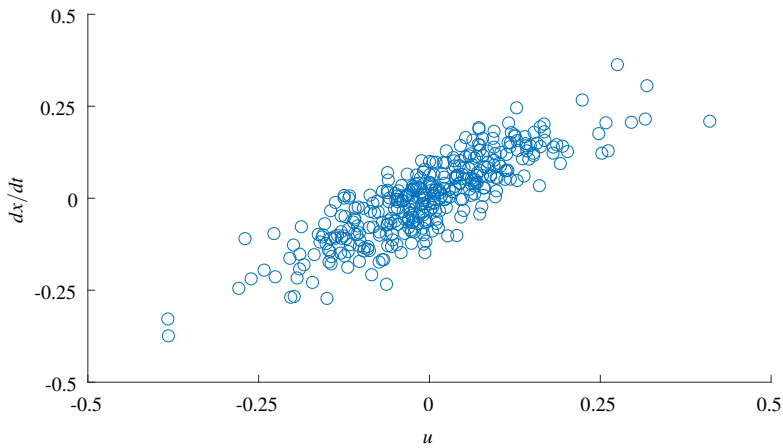


Figure 5.7 Scatter plot of estimated actuator action u and measured storage level derivative dx/dt in the real machine

6

Conclusion

On the whole, this project has been successful in achieving the goals defined from the start. The simulation model created for the project seems to provide an accurate representation of the real machine as seen in Figs. 5.7-5.6. The scatter plot shows a clear linear correlation between delayed control signal and storage level derivative, as predicted by the delayed integrator model. The scatter plot also hints that the region of the non-linear actuator that this particular machine operates in is adequately modeled as a line function.

The parameter estimates provided by the process estimator quickly converge from reasonably poor initial conditions to good estimates of the actual process parameters. Using the estimated parameters, the variable PID controller is able to compensate for the estimated process gain and dead-time and controls the system well using the pre-set PID parameters chosen for the reference model and the prediction-based dead-time compensation.

The adaptive control system tracks changes in process gain and dead-time as they appear, and should provide self-tuning and adaptive control for Tetra Pak's entire range of similar machines.

7

Discussion

The designed adaptive control system is largely successful, but some precautions are given to potential users.

In Fig. 5.5 showing the gain estimation in the real machine, the gain estimate returns very quickly when the gain disturbance is removed. This coincides with a small 'bump' in the controlled level which seems to have helped the estimation converge quickly to the new value. The bump was probably caused either by the method with which the gain disturbance was created or by non-bumpless transfer in the integrator variable I , or by some combination of both of these factors. An improved method of providing bumpless transfer is suggested in Section 7.1 of Future Work.

A common difficulty with identification in closed loop control is that error feedback, say from a PID-type controller, may introduce linear dependencies in the response matrix making the system parameters non-identifiable. This problem is described at length in Åström and Wittenmark's Adaptive Control (2008). For this particular process, the PID controller's derivative term provides a high enough order feedback to break these linear dependencies and thus allow unique estimation of the wanted parameters. The process dead-time also helps since a large part of the control is made via open loop prediction.

There's a common problem with dead-time estimation in that 'blindly cost-optimising' estimation algorithms (such as the Least Squares method used in this project) may find local minima if started with unfortunate initial conditions. The typical situation is when the input signal is (roughly) sinusoidal and the initial condition is such that the dead-time window is more than half of an oscillation period away from the 'correct' dead-time; then the estimation is likely to fall into a local minimum correlating the wrong peak of the input to the output. In this system, such a sinusoidal oscillation may ironically even be created as a limit cycle from an incorrect dead-time compensation. A reasonable precaution to avoid this may be to err on the low side when providing initial values for the dead-time estimate - the

limit cycles created from the incorrect dead-time estimate generally have a longer period time than the dead-time, so using a low initial estimate should ensure that the 'correct' minimum is found first.

7.1 Future Work

More Bumpless Integrator

The method used to provide bumpless control upon changing of the estimated gain input \hat{b} is not perfect - using the method described in Chapter 2, the integrator part does indeed stay the same before and after changing \hat{b} , but since \hat{b} also scales the proportional part there may still be a control discontinuity if the proportional error is nonzero. Better results may be had by recalculating the integrator's I variable to give the same output with the new \hat{b}_k as with the old \hat{b}_{k-1} .

Off-line estimation startup

The P_0 matrix was chosen as a 'large' identity matrix for the simulation and testing. Better start values for the initial estimation variable $\hat{\theta}_0$ and the initial covariance matrix P_0 may be given by performing off-line estimation on data points from a number of time steps before closing the adaptive loop. If used with a process that does not show significant process variation, this offline estimation may even be used to tune a non-adaptive controller.

Generalized Estimator

The estimator could with only a few minor modifications be used to estimate the gain and delay between any two similarly related signals. The way to do this is to redesign the adaptive estimator using a simple gain and delay process like $x_{k+1} = bu_{k-\hat{d}}$.

7.2 Summary

Tetra Pak wanted a self-tuning and adaptive controller to be designed for an unknown integrator process with unknown dead-time. This was accomplished using an extended regression model for simultaneously estimating the process gain and dead-time using a recursive Least-Squares algorithm. The estimated gain is used to scale the process into a reference process, while the dead-time estimate provides dead-time compensation via state prediction.

The adaptive control system is successful in self-tuning on start-up and in tracking process disturbances in both gain and dead-time.

Bibliography

- Åström, K. and B. Wittenmark (2008). *Adaptive Control*. Dover Books on Electrical Engineering. Dover Publications. ISBN: 9780486462783. URL: https://books.google.se/books?id=L0m_CR-IK24C.
- Blinn, J. F. (1994). “Quantization error and dithering”. *IEEE Computer Graphics and Applications* **14**:4, pp. 78–82. ISSN: 0272-1716. DOI: 10.1109/38.291534.
- De Keyser, R. (1986). “Adaptive dead-time estimation”. *Proceedings of the IFAC Adaptive Systems in Control and Signal Processing Conference*, pp. 385–389.
- Johansson, R. (2013). *Predictive and Adaptive Control*. KFS i Lund AB.
- Kurz, H. and W. Goedecke (1981). “Digital parameter-adaptive control of processes with unknown dead time”. *Automatica* **17**:1, pp. 245–252. ISSN: 0005-1098. DOI: [http://dx.doi.org/10.1016/0005-1098\(81\)90099-6](http://dx.doi.org/10.1016/0005-1098(81)90099-6). URL: <http://www.sciencedirect.com/science/article/pii/0005109881900996>.
- Tan, L. and J. Jiang (2014). *Digital Signal Processing (Second Edition)*. Second Edition. Academic Press, Boston, pp. i–. ISBN: 978-0-12-415893-1. DOI: <https://doi.org/10.1016/B978-0-12-415893-1.01001-5>. URL: <http://www.sciencedirect.com/science/article/pii/B9780124158931010015>.

Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden		<i>Document name</i> MASTER'S THESIS	
		<i>Date of issue</i> August 2017	
		<i>Document Number</i> TFRT-6030	
<i>Author(s)</i> Magnus Dagbro		<i>Supervisor</i> Ashraf Zarur, Dept. of Automatic Control, Lund University, Sweden Rolf Johansson, Dept. of Automatic Control, Lund University, Sweden Karl-Erik Årzén, Dept. of Automatic Control, Lund University, Sweden (examiner)	
<i>Title and subtitle</i> Adaptive Control of an Integrator with Unknown Dead-Time			
<i>Abstract</i> <p>An adaptive control system is proposed for an unknown integrator process with unknown dead-time. The process dead-time and gain are estimated simultaneously using an extended regression model and recursive least-squares estimation. The parameter estimates are used to scale the actuator signal, transforming the unknown process into a known reference process, while the dead-time estimate provides deadtime compensation via state prediction.</p>			
<i>Keywords</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316			<i>ISBN</i>
<i>Language</i> English	<i>Number of pages</i> 1-34		
<i>Security classification</i>			