Proceedings of the International Conference in Engineering 2005, 23 – 26 May 2005, Lagos, Nigeria

IDENTIFICATION AND CONTROL OF A 45 TON LOSS-IN-WEIGHT RAW CEMENT FEEDER

A. S. Bappah[1] and J. Katende[2]
[1]School of Technology Education,
Abubakar Tafawa Balewa University, Bauchi
asbappah@yahoo.com
[2]Department of Electrical Engineering,
Bayero University, Kano
sempa54@yahoo.com

ABSTRACT

This paper employs a Matlab based system identification toolbox (SID) to obtain a mathematical model of a loss-in-weight (LIW) feeder. In this study a pseudorandom binary sequence (PRBS) was used to excite a loss-in-weight feeder used in a cement manufacturing process. The input and output data were then processed using a Matlab system identification toolbox that yielded an ARMAX model of the process. After model validation a digital velocity PID controller was designed using the control toolbox of the Matlab. Details of system identification, controller design and simulation are presented in this paper.

Keywords: Loss-In-Weight Systems, System Identification, Modeling and PID Control.

Type of Paper: Research

INTRODUCTION

Mathematical models are essential for understanding the dynamic behavior of industrial systems. Modeling enables a mathematical treatment of such systems and facilitates synthesis of appropriate control algorithms for them. A mathematical model is the result mapping the relationships between a system's physical variables onto mathematical structures such as algebraic equations, differential equations or systems of differential equations. Mathematical models can be developed using different approaches. Theoretical approaches employ physical laws of nature. Empirical approaches are based on experimentation with the existing systems. Some other approaches combine both theoretical and empirical methods. Mathematical models for specific materials and equipment can as well be derived from first principles. Whereas simple models are used to obtain the gross feature of the system behavior, the complex models are for a detailed check of the performance of the control system. It is often beneficial, therefore, to combine mathematical model building with experiments.

If a plant cannot be experimented on or if the plant does not exist at all then theoretical modeling has to be employed to arrive at a mathematical model suitable for control algorithm synthesis and system simulation. Experimental analysis (system identification) of a plant requires measurement of input and output signals. The input signals can either be the operating signals of the plant or artificial test signals. This involves careful planning of the inputs to be applied so that sufficient information about the system dynamics is obtained. This is imperative in obtaining good quality data that contains sufficient information about the system behavior [3].

There are two ways to carry out this process; by applying an excitation rich in desired frequencies and directly identifying a parametric model from input-output data or by finding a non-parametric frequency response of plant by performing one or more experiments with periodic inputs. Then finding a parametric model based on frequency response samples.

The measurements are processed in an identification procedure that yields the plant's mathematical model. The result of the identification is an experimental model. The approach is facilitated by wide spread availability of computers and appropriate software. The models so developed can be used for purposes such as:

- better understanding of the process.
- verification of theoretical models.
- controller synthesis.
- optimization of the process.
- computation of inaccessible process variables.
- etc.

In this paper an experimental model is obtained for a loss-in-weight feeder used in a cement manufacturing industry. Input and output data from the plant are processed using the Matlab system identification toolbox to yield an ARMAX model of the system. After model validation a PID controller is designed and used in the simulation of the closed-loop system.

SYSTEM DESCRIPTION

A Loss-In-Weight (LIW) feeder system consists of a hopper and feeder mounted on load cells commonly used to deliver raw cement at a desired feedrate. When operated in a continuous discharge mode, accurate gravimetric operation is achieved by controlling the speed of the feeder in order to provide a constant decrease in the weight of the feed hopper. Feedrate regulation is in the main based on closed-loop control to deliver powders and other bulk solids at a desired feedrate [1]. Many industrial processes must handle bulk solids, often in granular or powdered form. Various devices such as conveyors, screw augers, pneumatic tubes, vibrating platforms, bucket elevators, and so on can be used for transporting and metering these materials. A typical LIW feeder system involves a hopper designed to hold material, and a screw discharge device that will feed material out of the hopper. The screw conveyor shown in Fig. 1 varies the flow by varying the speed of the screw.

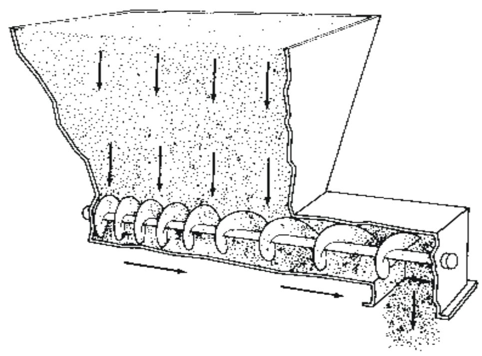

Fig. 1: The Screw Conveyor
***SOURCE: Balchen et al* [2]**

A variable speed motor drives the feed screw that discharges the material. As the material is discharged from the hopper, a feeder controller keeps track of the weight; constantly comparing it to how much the weight should have decreased based on the set-point. If the weight is not decreasing fast enough, the controller increases the control signal to speed it up. If the weight is decreasing too fast on the other hand, the control signal will be decreased accordingly.

To measure the feedrate, the hopper is mounted on load cells. The hopper is supported in such a way that the weight of the hopper and its contents is sensed by three load cells. An analog electrical signal from the load cells which corresponds to the mean weight on the load cells is used for feedback purposes. Two screw feeders actuated by thyristor driven variable speed motors are used to transfer material from the hopper at a specified rate. The advantage of this method is that there is no lag between the time the material is weighed and when it actually leaves the weighing device. Fig. 2 is the sectional schematic diagram of a 45ton LIW system used at Ashaka Cement factory in Gombe state, Nigeria.

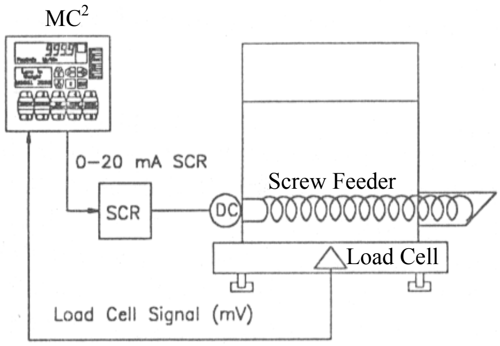

Fig. 2: Sectional View of a 45T LIW Feeder System.
SOURCE: Merricks Industries, Inc. **[1]**

Automatic control requires that the transport device be controllable over a reasonable range by means of some type of control variable. As material leaves the hopper, the weight will go down, showing a loss-in-weight to the controller. A microprocessor-based controller MC^2 is employed in the control loop's error channel. It compares the feedrate set-point value with the actual feedrate and employs a digital PID control algorithm to generate the control signals that actuate the thyristor regulating the speed of the motors.

The paper addresses the issue of designing and simulating a LIW process. A model of a LIW feeder is developed from actual open loop LIW process data using the Matlab system identification toolbox (SID). The problem generally consists of data acquisition, estimation, characterization, and verification [4]. The procedure was followed to acquire input-output data necessary for system identification from a 45ton loss-in-weight raw cement feeder. Simulation results based on the identified process are then presented.

DATA ACQUISITION

The first and most important step is to acquire the input - output data of the system to be identified. Open loop input-output data relating PV to CO was extracted for the purpose of this study. The average operating SP for the feeder under investigation, Feeder 1 (main) was 90TPH. At this constant SP, the CO was shifted between two levels by ±20% for the period of this experiment. The procedure was facilitated by the microprocessor based MC^2 process controller in the "manual %" mode of operation. This is the first setpoint mode available in the MC^2 feeder controller. This mode allows the user to control the controller output directly. There is no closed loop control of feedrate in this mode. The output to the motor speed controller can simply be regulated by setting this parameter which is adjustable from 0 to 100%. The method was both safe and realizable. The following algorithm shows the order in which identification data was obtained.

Step 1: Log on SP @ 90TPH
Step 2: Set to manual % Set-point mode'
@ ±20% of CO
Step 3: While time < specified duration
Read PV
Read CO
Step 4: Read SP
Read clock
go to step 1.

Fig. 3 below shows the LIW Feeder identification data set when imported and preprocessed. The upper sub - plot being the measured feedrate in tons per hour (output signal) and the lower sub - plot represents the input signal as % change in CO. The output data vector is a scaled measured feedrate in tons per hour (PV) while the input data vector is in form of percentage change in CO.

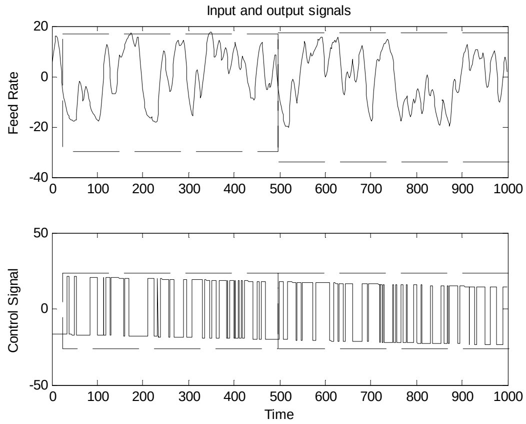

Fig. 3: Data Views with SID Toolbox.

That measured data set was exported into the Matlab workspace as *Fedat.sid* and examined to determine the time delay. The data set is then preprocessed by detrending the data to remove offsets. This detrended working data, *Fedatdd*, is shared into two equal places through the action of select range prop-ups. The first range 0 to 500 samples, *Fedatdde* is the final working data for system identification. While the second range 501 to 1000 samples, *Fedatddv* is the validation data.

MODEL ESTIMATES

The second step is estimation, which involves determining the numerical values of the structural parameters, which minimize the error between the system to be identified, and its model. Parameter estimation can be formulated as an optimizing problem, where the best model is the one that best fits the data according to the given criterion. There is a large number of different estimation methods. Interestingly, however, the choice of method is not crucial as there is no method that is universally the best [5]. Common approaches to parameter estimation are the prediction error method (PEM), maximum likelihood (MLE) and instrument-variable (IV). Parameterizations in the Matlab toolbox are preferably done with the PEM estimation method, which consistently estimates models from the data set. Although called conventional by many researchers, the PEM method will consistently estimate a system if the data set is informative and the model set contains the true system, irrespective of whether or not data have been collected under feedback [4, 6]. All these technical provisions are therefore on the side of experiments in open loop.

MODEL STRUCTURE

The third step defines the structure of the system, for example, type and order of the differential equation relating the input to the output. The set of systems from which a model is chosen is usually defined by a model structure. Once the structure is chosen, an appropriate excitation is applied to the true plant and input-output data is measured [4]. Appropriateness here refers to the ability to outweigh the effect of disturbances and to excite the desirable system dynamics.

The measured input - output data is then mapped into a model which best explains this data. Let $\{e(t)\}$ be a pseudorandom sequence which is similar to white noise in the sense that

$$\frac{1}{N}\sum_{t=1}^{N} e(t)e(t-\tau) \to 0 \quad \text{as } N \to \infty \; (\tau \neq 0) \qquad \ldots(2)$$

This relation is to hold for τ at least as large as the dominating time constant of the unknown system. Let $y(t)$ and $u(t)$ be scalar signals and consider the model structure

$$A(q^{-1})y(t) = B(q^{-1})u(t) + C(q^{-1})e(t) \qquad \ldots(3)$$

where

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{na} q^{-na}$$
$$B(q^{-1}) = 1 + b_1 q^{-1} + \ldots + b_{nb} q^{-nb}$$
$$C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_{nc} q^{-nc}$$

…(4)

The parameter vector is taken as

$$\theta = (a_1 \ldots a_{na} \quad b_1 \ldots b_{nb} \quad c_1 \ldots c_{nc})^T$$

…(5)

The model in Eqn. (3) can be written explicitly as the difference equation

$$y(k) + a_1 y(t-1) + \ldots + a_{na} y(t-na) = b_1 u(t-1) + \ldots + b_{nb} u(t-nb) + e(t) + c_1 e(t-1) + \ldots + c_{nc} e(t-nc)$$

…(6)

but the previous form using the polynomial formalism will be more convenient. This model is called an ARMAX model, which is short for an autoregressive moving average with an exogenous signal (a control variable $u(t)$). When all $b_i = 0$, it is called an MA (moving average) process, while for an autoregressive (AR) process all $c_i = 0$.

The ARMAX model structure is chosen and Matlab is required to select the best model order in the range of 1 – 10. The result indicates that a second order ARMA model best describes the system under investigation. Using a sampling interval of 1secs, the state space model for the identified process is:

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k + Du_k$$

…(7)

where

$$\mathbf{A} = \begin{bmatrix} -0.8754 & -1.4060 \\ 1 & 0 \end{bmatrix}; \qquad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix};$$

$$\mathbf{C} = \begin{bmatrix} 0.0830 & 1.2654 \end{bmatrix}; \qquad \mathbf{D} = \begin{bmatrix} 0 \end{bmatrix};$$

u is the control signal, y is the federate, and x is the state vector.

The corresponding pulse transfer function is:

$$\frac{PV}{CO} = \frac{Y(z)}{U(z)} = \frac{0.4750z + 0.2750}{z^2 - 0.5833z + 0.4167}$$

…(8)

The process model can be re-written in the difference equation form as

$$y(k) - 0.5833\,y(k-1) + 0.4167\,y(k-2) = 0.4750\,u(k-1) + 0.2750\,u(k-2)$$

…(9)

Fig. 4 shows the uncompensated step response of the simulated plant model with a maximum overshoot of 20% and settling period of about 0.80secs at known sampling rate.

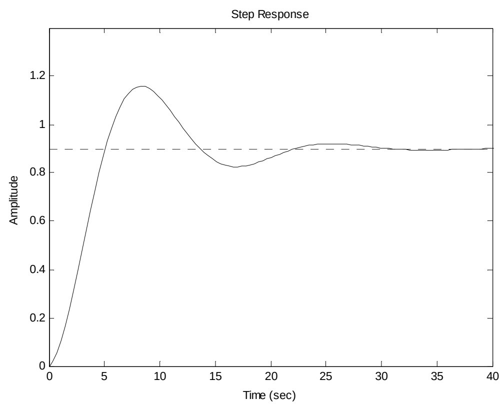

Fig. 4: Time Response of the Identified Process.

This configuration results into a steady state offset of about 0.1. These are indicators to an open loop stable second order process with trivial delay. The process is therefore validated with the remaining one half of the data that were not used for model estimation (i.e. *Fedatddv*). Standard validation tools are residual analysis and cross – validation [7].

MODEL VALIDATION

The final step, verification, consists of relating the system to the identified model responses in time or frequency domain to instill confidence in the obtained model. Residual (correlation) analysis, Bode plots and cross-validation tests are generally employed for model validation. In the residual analysis models are statistically evaluated by way of correlation tests. The auto–correlation and cross-correlation functions of the errors with the outputs do not go significantly outside the 99% confidence bounds as shown in Fig. 5 below;

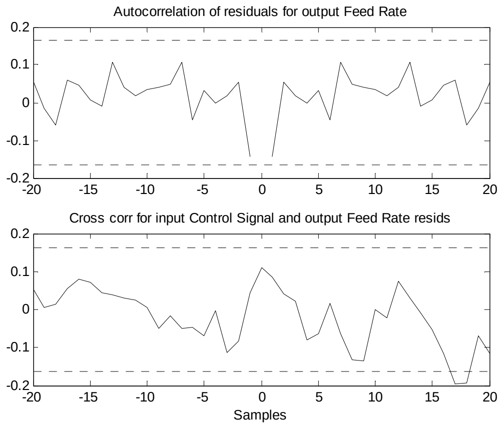

Fig. 5: The Auto-correlation and Cross-correlation Functions for Open-loop Feeder Model.

In cross–validation the percentage of fit between the predicted and the measured output are evaluated in the SID toolbox. The cross-validation analysis in Fig. 6 shows that the simulated outputs follow the measured output closely, since relatively very high percentage of fit was obtained.

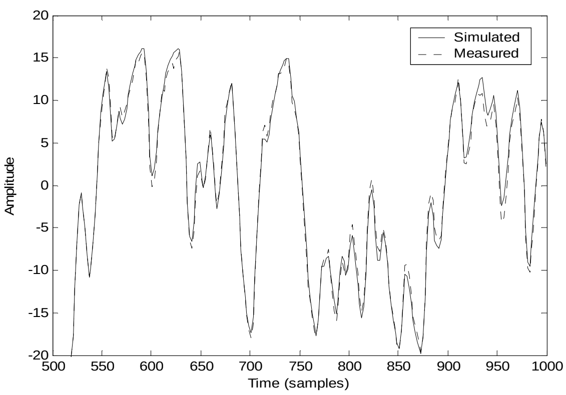

Fig. 6: Cross Validation Test for the Feeder Model.

THE PID CONTROLLER

The proportional plus integral plus derivative (PID) controller is a standard building block in industrial control systems. For many processes it gives good performance with reasonable robustness to incorrect process model assumptions and limited process parameter changes. The controller is also easy to understand amongst non-specialist plant operators, with tuning rules that have been validated in a wide variety of practical cases. It has been reported that in process control applications, more than 95% of the controllers are of PID type [8]. The PID controller may be implemented in continuous or discrete time, in a number of controller structures [7]. Fig. 7 represents a simplified block diagram of a typical Loss-in-Weight control loop.

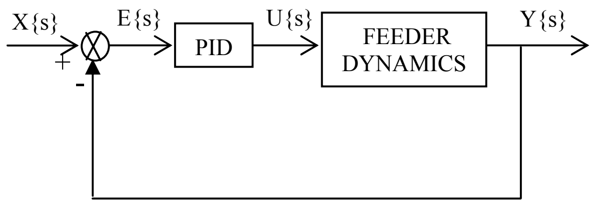

Fig. 7: Control of a LIW Feeder System.

The ideal continuous time PID controller is expressed in Laplace form as follows:

$$G_c(s) = \frac{U(s)}{E(s)} = k_c\left[1 + \frac{1}{sT_i} + sT_d\right] \quad (10)$$

with k_c = proportional gain, T_i = integral time constant and T_d = derivative time constant. If $T_i = \infty$ and $T_d = 0$ (i.e. P control), then the closed loop measured value will always be less than the desired value for processes without an integrator term, as a positive error is necessary to keep the measured value constant, and less than the desired value. A proportional term, k_c will have the effect of reducing the rise time and will reduce, but never eliminate the steady-state error. With proportional control a steady-state error always exists after a setpoint change or a load disturbance and this phenomenon is termed offset.

The introduction of integral action facilitates the achievement of equality between the measured value and the desired value, as a constant error produces an increasing controller output. This will have the effect of eliminating the steady-state error, but it may make the transient response worse. . An integral control term Integral windup describes one vital aspect about integral action when an increase in control effort does not reduce the deviation. Suppose a proportional plus integral (Pl) controller is in use in a particular loop and something happens whereby the controller cannot eliminate the deviation. For example a control valve could stick, or the steam supply pressure to a heater could be too low and so the steam valve opens fully. Under these circumstances the controller output will grow due to the integral action, but without effect. When the cause of the trouble is eliminated, the deviation will drop rapidly, but because of the build up of integral action the controller output cannot change until there is a deviation of the opposite sense and of sufficient duration to cancel the historical build up of control effort. This build up is termed reset windup. The result is a large overshoot and a considerable delay before the system is brought under control once more. Controllers which employ techniques for limiting integral windup are said to incorporate anti-reset windup.

The introduction of derivative action means that changes in the desired value may be anticipated, and thus an appropriate correction may be added prior to the actual change. It will have the effect of increasing stability of the system, reducing the overshoot, and improving the transient response. Sudden changes in the desired value may cause very large changes to the control effort, which are referred to as setpoint kicks. One way of avoiding a setpoint kick is to filter the setpoint signal so allowing only gradual changes to the setpoint. However, this affects the other actions of the controller where there is no problem. Another way of overcoming setpoint differentiation is to apply the derivative action only to the measured value. This is the one adopted by many commercial controllers [9].

In practical implementations the PID controller ensures bumpless transfer from manual to automatic mode. Thus, in simplified terms, the PID controller allows contributions from present, past and future controller inputs.

TUNING RULES

Because it is often difficult to obtain accurate transfer function models for some processes and their determination can also be time consuming, most of the industrial controllers are tuned based on the plant's frequency response data. Process reaction curve tuning rules are based on calculating the controller parameters from the model parameters determined from the open loop process step response. This method was originally suggested by Ziegler and Nichols, who modeled the SISO process by a first order process plus dead time (FOPDT) model, estimated the model parameters using a tangent and point method and defined tuning parameters for the P, PI and PID controllers. Other process reaction curve tuning rules of this type are also described, sometimes in graphical form, to control processes modeled by a FOPDT model [9]. The advantages of such tuning strategies are that only a single experimental test is necessary, a trial and error procedure is not required and the controller settings are easily calculated; however, it is difficult to calculate an accurate and parsimonious process model and load changes may occur during the test which may distort the test results.

Ultimate cycle tuning rules are calculated from the controller gain and oscillation period recorded at the ultimate frequency (i.e. the frequency at which marginal stability of the closed loop control system occurs). The first such tuning methods was defined by Ziegler and Nichols for the tuning of P, PI and PID controller parameters of a process that may or may not include a delay.

Ziegler and Nichols realized that if the gain margin could be estimated quickly then it should be possible to find good controller settings for many practical situations from this information. They therefore suggested that one way to do this practice would be by setting a PID controller into the P mode and adjusting the gain until an oscillation took place. Since nearly all the PID controllers then were pneumatic, the values of controller gain k_c , and the oscillation frequency, ω_c were easily measured. Assuming a linear plant model then the frequency of oscillation ω_c on the Nyquist plot of the plant frequency response has a phase shift of -180° and the gain margin is $1/k_c$ expressed in decibels (dB), the desired optimal PID controller parameter settings are recommended as [10];

$$k = 0.6k_c$$
$$T_i = k\omega_c/\pi$$
$$T_d = k\pi/4\omega_c$$

…(11)

The controller settings are easily calculated; however, the empirical nature of the method means that uniform performance is not achieved in general, several trials must typically be made to determine the ultimate gain, the resulting process upsets may be detrimental to product quality and there is always a danger of misinterpreting a limit cycle as representing the stability limit.

A variety of methods have been proposed for gaining the information required for tuning but must require the plant to be disturbed in some way. A procedure that has received much attention recently is relay autotuning where the controller switches in the tuning mode to operate as an on-off relay and obtains data from the resulting limit cycle. The resulting limit cycle data can be used to estimate the critical point of an assumed plant transfer function, that is, where its frequency response has a phase slight of 180° and thus may be regarded as an automated Ziegler-Nichol's test. Direct synthesis tuning rules result in a controller that facilitates a specified closed loop response. These methods include pole placement strategies and frequency domain techniques. Performance (or optimization) criteria, such as the minimization of the integral of absolute error in a closed loop environment, may be used, sometimes in graphical form, to determine a unique set of controller parameter values.

Tuning rules are easy to use, even in the absence of an accurate process model. These design methods are suitable for the achievement of a simple performance specification, for a compensated process with a non-dominant delay. PID implementation is often recommended for the control of processes of low to medium order, with small delays, when controller parameter setting must be done using tuning rules.

SIMULATION RESULTS

A discrete-time PID controller was developed using the well known Ziegler-Nichols tuning rules and applied to the computed SISO transfer function. The task is to achieve a desired feedrate with minimum control effort. The design specifications for unit step inputs are as follows;

- Peak overshoot, $P_o \leq 5\%$
- Rise time, $t_r \leq$ 2secs
- Settling time, $t_s \leq$ 5secs
- Zero steady-state error to step input
- Zero steady-state error due to step disturbance.

Fig. 8 below shows the process variable, Y{s} following a load disturbance under the digital PID controller. The result confirms the versatility of a simple PID algorithm when applied to a practical second-order cement process.

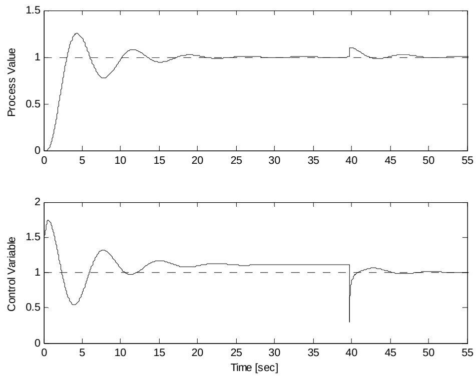

Fig. 8: Step and Load Disturbance Response.

The PID controller parameters thus obtained are as follows:

$K_p = 1.4588,$

$T_i = 2.6043,$

and

$T_d = 0.6511.$

CONCLUSION

This paper presented details of the determination of an experimental model a 45 ton loss-in-weight raw cement feeder. The complete hardware setup and the algorithm followed for data acquisition have been explained. Based on the model identified a PID controller was designed. Simulation results indicate a great deal of flexibility and straightforward realization.

REFERENCES

1. Operation and Maintenance Instructions: MC^2 Merricks Micro-computer Controller Software 30.00. HP. H & H Services Ltd, Kirkby-in-Ashfield. Notts.

2. Balchen, J G. and Mummé (1988) Process Control: Structure and Applications. Van Nostrand Reinhold, NY USA.

3. Ljung, L. (1999) System identification Theory for the User, 2nd ed. Prentice Hall. New Jersey, USA.

4. Åström, K J. and Eykhoff, P. (1971) System Identification – A Survey. Automatica, 7: 123-62.

5. Åström, K J. and Wittenmark, B. (1990) Computer Control Systems: Theory and Design. Practice-Hall International Inc. NJ, USA.

6. Gustavsson, I., Ljung, L. and Söderström, T. (1977) Identification of Process in Close Loop – Identifiability and Accuracy Aspects. In: The Analysis of Time Series – An Introduction (ed. Chris Chatfield) Chapman and Hall. London.

7. Isermann, B, Lachmann, K-H and Matko, D. (1992) Adaptive Control System Practice-Hall International Inc.UK.

8. Bialkowski, W.L. {1996} The Control Handbook Series Editor W.S. Levine, CRC/IEEE Press, Boca Rantom, Florida – USA.

9. Golten, J., and Verwer, A. {1992} Control System Design and Simulation McGraw-Hill Book Company, International Edition, Singapore.

10. Atherton, D. P., and Majhi, S. {1998} Tuning of Optimum PI-PD Controllers 3rd Portuguese Conference on Automatic Control, Controlo '98, 9th – 11th September 1998, Coimbra, Portugal, pp 549 – 555