

https://dspace.org/

An adaptive optimizing regulator for an autogenous Mill

Bleloch, Mark John

2015-01-09
http://hdl.handle.net/10539/16306

Downloaded from a DSpace repository. https://dspace.org/

Φ is the regression vector.

This estimator has the problem that once it has converged to the set of model parameters it "falls asleep" and will not converge if the plant parameters change again. This loss of sensitivity is due to M being non-increasing.

The RLS estimator can be modified to avoid the above problem. Various modifications exist (Seborg et al. 1986) and the choice is usually a trade-off between accurate estimates and fast convergence to changing parameters. The convergence theory for the standard RLS can be applied to any modified RLS provided:

1. The covariance matrix P (one of the terms in M) is only increasing in magnitude.

2. There is an upper bound on P.

We look at some of the better known modified RLS estimator algorithms.

RLS with exponential data weighting: (Goodwin and Sin, 1984 :64) For the forgetting factor $\lambda<1$ the estimator gives recent plant data a higher weighting so that old data, that may not be accurate if the plant has changed, is forgotten. This is a popular method because λ can be selected according to how fast the plant is changing or what frequency plant changes the estimator must track. It nevertheless has the following pitfall if λ is not chosen carefully: Consider a plant under regulatory control and in a steady state. If the physical plant does not change (no significant unmeasurable disturbances) and old plant data is being forgotten then there might not be sufficient information content in the plant input to estimate the parameters. This condition of the plant input not being persistently exciting (Goodwin and Sin, 1984 :72) leads to deteri-

oration of the estimates and in adaptive control is referred to as "estimator windup".

RLS with Covariance Resetting: (Goodwin and Sin, 1984 :65). The estimator gain is kept high by periodically resetting the covariance matrix P to some large value. The estimator must be given enough time to converge before P is reset again. Estimates are at their worst just after P is reset and the model should only be used just **before** P is reset. This method is also susceptible to "estimator windup" (discussed in the previous section) but the problem can be overcome by, instead of resetting P periodically, P is only reset when a change in the plant is detected.

Regularised Constant Trace Algorithm: (Goodwin et al, 1986).The covariance matrix is updated using:

$$P(t) = P(t-1).(k_0/\text{trace}(P(t-1))) + k_1 I$$

where k_0, k_1 are positive constants that are chosen to set the minimum size of the P matrix diagonal elements. The trace(P(t-1)) is the product of the P matrix diagonal elements and I is an n × n identity matrix.

By updating the P matrix in this way it ensures that the diagonal elements of P never fall below a value determined by k_0 and k_1 This guarantees a minimum sensitivity for the estimator.

Choosing between the above methods will depend on the plant, a specific plant might even require a combination of the above methods.

3.3 MAKING THE ON-LINE OBJECTIVE FUNCTION MODEL IDENTIFICATION ROBUST

A key feature that will contribute to the practical success of the optimizer is the reliability of the on-line model. Small model inaccuracies may result in a degree of sub-optimality or slower convergence, but an estimator that diverges will result in controller failure. This section looks at how the model identification can be made robust in the light of recent developments in adaptive control.

On a real plant it is not always guaranteed that the estimator will converge. We can identify two areas that need to be looked at before applying the model identification (discussed in previous section) to a real plant. They are:

1. Ensuring the estimator gets only the plant data necessary to model the frequencies of interest. This is to circumvent the following:

   - Bandwidth of model: the assumed model is only valid for a certain frequency range. Care must be taken so the estimator sees only those frequencies that need to be modelled.

   - Deterministic disturbances: Care must be taken to ensure that the estimator models the input-output relationship and not divert all it's energy into modelling uncontrollable and possibly unobservable modes.

   - Measurement noise: It must be assumed that white Gaussian bandlimited noise is present on the plant variables. The noise seen by the estimator must be carefully processed to ensure reliable estimator operation.

2. Modelling errors within the bandwidth of interest: the assumed model structure may not be an accurate enough representation of the plant. The model order may be too low and there are unmodelled dynamics, or the steady state of the model may not fit the plant accurately. The latter is usually due to trying to fit a linear model to a highly non-linear plant. If the modelling error that drives the estimator is comparatively large then the estimator may diverge.

In the past dynamic model identification for practical adaptive controllers (eg. ASEA Novatune) has been fraught with the above problems. To overcome these problems ad-hoc methods or "safety-nets" have been used to constrain the range of operation of the estimator. This ensures stable operation of the estimator but has undesirable and unpredictable effects on the quality of the estimation. This detracts from the advantages gained by adaptive control over conventional control.

Recent developments in adaptive control have led to theoretical and more systematic ways of overcoming the above problems. Some of these developments are as yet unknown and are presented here in some detail. The material for the next two sections comes mainly from ideas put forward by Prof. I.M. MacLeod and partly from very recently published papers.

3.3.1 COPING WITH DETERMINISTIC DISTURBANCES AND PLANT NOISE

Elimination of deterministic disturbances and plant noise outside the bandwidth of interest can be done by careful digital filtering. The theory is presented below and an example implementation is given in appendix A.

For simplicity with notation a SISO linear model is assumed. This analysis is easily extended for a single output multi-input linear or non-linear model (as described in a previous section on model selection), but the notation is very cumbersome.

Assume the plant objective function can be modelled approximately by:

$$y = \frac{B}{A}u + \frac{C}{A}z + \xi + r \quad \ldots\ldots\ldots \quad (5)$$

where the discrete variables are:

- $y \equiv$ predicted value of the plant objective function
- $u \equiv$ plant inputs and/or regulator setpoints
- $z \equiv$ measurable disturbance input
- $r \equiv$ unmeasurable deterministic disturbance
- $\xi \equiv$ represents the modelling error and noise

Note that y is not the plant output and u includes the lower level regulator setpoints.

The q^{-1} operator polynomials are:

$$A(q^{-1}) \equiv a_0 + a_1 q^{-1} + \ldots\ldots + a_n q^{-n}$$
$$B(q^{-1}) \equiv q^{-d}(b_0 + b_1 q^{-1} + \ldots\ldots + b_m q^{-m})$$
$$C(q^{-1}) \equiv q^{-d}(c_0 + c_1 q^{-1} + \ldots\ldots + c_r q^{-r})$$

and are assumed to have the following properties:

i) A is monic

ii) A and B are co-prime

iii) $\partial_A = n$

$$\partial_B \leq \partial_A - 1 = m$$
$$\partial_C \leq \partial_A .$$

This is a DARMA model but has an extra term for bounded noise.

A deterministic disturbance r can be modelled by:

$$O(q^{-1})r = 0 \quad \ldots\ldots\ldots (6)$$

(Goodwin and Sin, 1984 :156)

Now multiply (1) by the polynomial operator $A(q^{-1})\ O(q^{-1})$:

$$AOy = BOu + FOz + AO\xi \quad \ldots\ldots\ldots (7)$$

The above inclusion of r in the model, equation (3) is known as the internal model principle.

Since ξ is a random signal, the polynomial AO before it would imply near differentiation of this noise term. This is not acceptable and the model has to be processed further.

Both $A(q^{-1})$ and $O(q^{-1})$ may involve differentiation or near differentiation. To deal with the A polynomial we proceed as follows:

Introduce a filter 1/E where E is a stable, monic polynomial and $\partial_E = \partial_A$.

$$E(q^{-1}) = e_0 + e_1 q^{-1} + \ldots\ldots + e_n q^{-n}$$

Also, write $\eta = A\xi$ + modelling errors.

Rewrite (3) as

$$AOy' = BOu' + COz' + O\eta_f' \quad \ldots\ldots\ldots (8)$$

where

$$y' = (1/E)y$$

$$z' = (1/E)z$$

$$u' = (1/E)u$$

$$\eta_f' = (A/E)\xi$$

The $O(q^{-1})$ polynomial will have zeros on the stability boundary which implies near differentiation. $O(q^{-1})$ could become unstable due to near differentiation of the noise frequencies. In the same way that the E polynomial was introduced to overcome near differentiation due to A, we introduce a Q polynomial to prevent near differentiatior by O operating on η_f'.

Choose Q, a stable polynomial operator "close to" O. If the Q polynomial is not "close to" the O polynomial then one risks filtering out important plant frequencies. Appendix A gives an example of how Q is chosen.

Operating by 1/Q gives:

$$Ay_f = Bu_f + Cz_f + \eta_f \quad \ldots\ldots\ldots (9)$$

where

$$y_f = \frac{O}{EQ}\, y$$

$$u_f = \frac{O}{EQ}\, u$$

$$z_f = \frac{O}{EQ}\, z$$

$$\eta_f = \frac{AO}{EQ}\ \xi \qquad (\simeq 0)\ .$$

The operator $\frac{O}{EQ}$ is a digital pre-filter that ensures the estimator will only try to model the frequency band of interest. The noise term η_f is now acceptable for estimation and should not affect the estimator.

Let the model predict $y' = \frac{O}{Q}\ y$, which means that the estimated value appears after a high pass filter. The model can predict either y, y_f or y'. From a mathematical point of view an elegant choice seems to be y', but the exact implications of this choice over y_f and y are not clear.

Let $y' = Ey_f = \frac{O}{Q}\ y$.

Also, add Ey_f to either side of equation (9)

$$Ey_f = Ey_f - Ay_f + Bu_f + Cz_f + \eta_f \quad \ldots\ldots\ldots\ (10)$$

This can now be arranged into standard regression form as required for parameter estimation:

$$y' = (E-A)y_f + Bu_f + Cz_f + \eta_f \quad \ldots\ldots\ldots\ (11)$$

or in regression form

$$y'(t) = \phi(t-1)^T\theta + \eta_f$$

where

$$\Phi(t-1)^T = [y_f(t),\ldots\ y_f(t-n),$$
$$u_f(t-d)\ldots\ u_f(t-d-m),$$
$$z_f(t-d)\ldots\ z_f(t-d-r)] \quad \ldots\ldots\ldots\ (12)$$

$$\hat{\theta}^T = [e_0-a_0,\ldots e_{n-1}-a_{n-1},b_0\ldots b_m,c_0\ldots c_r] \quad \ldots\ldots\ldots\ (13)$$

3.3.2 COPING WITH MODELLING ERROR WITHIN THE BANDWIDTH OF INTEREST

Estimator divergence can be due to the best possible model fit only being an approximation to the plant. It is caused by the estimation error that drives the estimator always being large. Present methods simply use a basic deadzone function where the model parameters are not updated if the error is less than a set constant. A more attractive approach according to Goodwin et al. (1986) is to vary the width of the deadzone dynamically according to the noise present and the magnitude of the signals being processed by the estimator.

The basic dead zone function is defined as follows:

$$f(g,e) = \begin{cases} e - g & \text{if } e > g \\ 0 & \text{if } |e| \leq g \\ e + g & \text{if } e \leq -g \end{cases}$$

When the error e(t) that drives the estimator falls below |g| then e(t) is zero. If |e(t)| > |g| then the error that drives the estimator is e(t). A relative deadzone is now described where the |g| is not constant but changes dynamically.

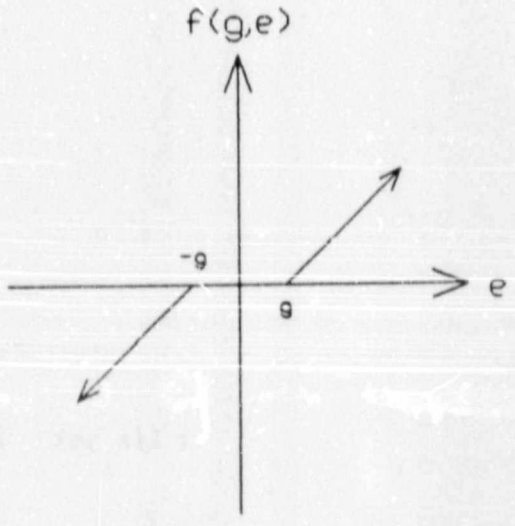


Figure 4. The Basic Deadzone Function

Take the standard RLS estimator and add a Boolean variable a(t), if a(t) is 1 then the parameter estimates are updated, otherwise they are not updated. This can be written as.

$$\theta(t) = \theta(t-1) + a(t).\frac{P(t-2)\phi(t-1)}{\phi(t-1)^T P(t-2)\phi(t-1) + 1}.e(t)$$

$$P(t-1) = P(t-2) - a(t).\frac{P(t-2)\phi(t-1)\phi(t-1)^T P(t-2)}{\phi(t-1)^T P(t-1)\phi(t-1) + 1}$$

Now choose

$$\sigma_0' \in (0,1)$$

$$\varepsilon_0 \geq 0$$

$$\varepsilon_1 \geq 0$$

$$\varepsilon_2 \geq 0$$

$$\varepsilon_3 \geq 0$$

$$m_0 \geq 0$$

so that

$$|\eta_f(t)| \leq m(t) \quad \text{for all } t$$

where $m(t)$ is the solution of

$$m(t) = \sigma_0 m(t-1) + \varepsilon_0 + \varepsilon_1|u(t-1)| + \varepsilon_2|y(t-1)| + \varepsilon_3|z(t-1)|$$

for any $\sigma_0 \in (\sigma_0', 1)$, and $m(0) = m_0$.

Then choose $\varepsilon_4 > 0$ and $\alpha \in (0,1)$ to implement the dead zone as follows:

$$\beta = \sqrt{\varepsilon_4 + 1/(1-\alpha)}$$

and $a(t) = \beta s(t)$

where

$$s(k) = \begin{cases} 0 \text{ if } |e(t)| \leq \beta m(t) \\ f(\beta m(t), e(t))/e(t) \quad \text{otherwise} \end{cases}$$

Some guidelines as to the selection of the deadzone constants and its behaviour are given in an M.Sc.(Eng) thesis by van der Merwe (1987).

The function a(t) thus implements the dead zone to account for modelling errors. The key idea is that a(t) = 0 when the prediction error e(t) falls below the bound $\beta m(t)$.

3.4 EXTRACTING THE STEADY-STATE MODEL FROM THE DYNAMIC MODEL

The optimizer needs a steady-state model which is extracted from the dynamic model. This is easily done by setting the backward shift operator q^{-1} equal to unity. It means that the variables are assumed not to change with time and the plant is in steady-state. Important questions arise as to whether this is a good way of obtaining a steady-state model:

- How well does the dynamic model predict, during plant transients, the future steady-state values?
- How does the dynamic model order affect the steady-state prediction?
- Does over-estimation or under-estimation of the plant dead-time affect the extracted steady-state model?
- Is a linear dynamic model good enough to give accurate steady-state information?
- Is the extracted steady-state model more, or less sensitive to bad modelling errors than the dynamic model?
- Is the extracted steady-state model sensitive to poor model parameter estimation?

It is beyond the scope of this research project to answer the above questions in general. The validity of obtaining the steady-state model from the dynamic model will be looked at only in the context of two specific case studies.

3.5 ON-LINE OPTIMIZATION

3.5.1.1 Introduction

The two step method described in the section entitled "Theoretical Foundations for the Adaptive Optimizing Regulator" on page 11 has been criticized (Nachane, 1978) on the grounds that the point parametric model used to calculate the plant trajectory is based on non-optimal conditions. An integrated approach to identification and optimization is required. Various integrated approaches have been proposed. Nachane (1978) gives four alternative methods, Ellis and Roberts (1984) give the ISOPE algorithm (Integrated System Optimization and Parameter Estimation) and Brdys' and Roberts (1984) give an integrated approach based on Lagrangian optimality conditions.

All these methods assume on-line identification of a static plant model so that the estimator must wait for the plant to settle between estimator samples. This limits the plant input move update time. The estimator sampling rate must be the same as the optimizer sampling rate and hence the plant input move update rate. Only then is it possible to combine the minimization of the model error (parameter estimation) and the minimization of the objective function into one optimization problem. For

two-step, static identification and then optimization, nothing is lost in combining the optimization problems and an integrated approach is justified. Whereas for the two step, dynamic identification and extraction of a static model and then optimization, the dynamic identification can be done far more often than the static optimization and an integrated approach is not applicable.

Also, dynamic model identification is much faster, and assuming a good steady state model can be extracted, it allows a higher plant input move bandwidth. This can be used to increase the number of input moves to reach the optimum, so the model will change less markedly between input moves, and the problem of the plant trajectory being based on non-optimal conditions is not an issue. This motivates the two-step procedure provided that the static model is extracted from the identified dynamic model.

3.5.1.2 Optimization Algorithm

From the previous discussion on robust model identification and steady state model extraction it can be assumed that a reliable and very accurate point parametric steady state model of the objective function is available. Although the model is simple it is accurate due to its parameters being continuously updated. The convergence to new parameters is robust to modelling errors and to signal noise.

The remainder of this section looks at how the plant inputs y_1^* and u_2 are calculated in order to drive the plant to its optimum, as determined by the objective function, and keep it there in the face of persistent measurable and unmeasurable disturbances.

The algorithm based on a gradient search is given below. Recall the static optimization problem:

$$\min_{m} \Psi(y_2, u_1, m, d_s) \quad \ldots\ldots\ldots (14)$$

such that

$$f(y_1^*, y_2, d_s) = 0 \quad \ldots\ldots\ldots (15)$$

$$g(y_1^*, y_2, u, d_s) \leq 0 \quad \ldots\ldots\ldots (16)$$

where the symbols have there usual meaning except for convenience of notation m is introduced

$$m \equiv [\, y_1^*, u_2 \,]^T \equiv \text{optimizer outputs}$$

and from now on Ψ is assumed to be a function of the variables given above.

Unconstrained Optimization: Dealing with the unconstrained case first, the gradient search algorithm at the operating point $m(l)$ is:

$$m(l) = m(l-1) - \mu S_l \nabla_m \Psi|_l$$

where

$S_l \equiv$ positive definite matrix

$\mu \equiv$ input move stepsize

$\nabla_m \Psi|_l \equiv$ gradient of the objective function at $m(l)$

Remembering that $\mathbf{y}$ is a function of $\mathbf{m}$ in the general case so the gradient is given by:

$$\nabla_{\mathbf{m}} \Psi|_1 = \frac{\partial \Psi}{\partial \mathbf{m}}\Big|_1 + \frac{\partial \Psi}{\partial \mathbf{y}}\Big|_1 \left(\frac{d\,\mathbf{y}}{d\,\mathbf{m}}\right)_1$$

where

$$\left(\frac{d\,\mathbf{y}}{d\,\mathbf{m}}\right)_1 = (A^{-1}\,B)_1 \equiv \text{Jacobian matrix}$$

where A and B have there usual meaning as q^{-1} polynomial operators.

S_1 can be chosen to give a Newtonian, Quasi-Newtonian or if $S_1 = I_{n\times n}$ then it is a steepest descent search. The stepsize μ is chosen depending on the choice of S_1 to ensure convergence(Polak, 1971).

Appendix B gives a simple example of an implementation of the above gradient search algorithm.

Constrained Optimization: A detailed account is given in Garcia and Morari (1984). Here it is sufficient to give an understanding of what is involved and how a constrained optimization algorithm can be implemented into the controller.

The optimizer must produce feasible plant moves that lie inside a region of operation, bounded by the constraints $\mathbf{g}$. If the constraints $\mathbf{g}$ are non-linear it is not possible to always ensure a feasible direction. The only way to get around this is to introduce regulators so the constraints are $\mathbf{g}^*$, the setpoints to the lower level regulators. In this way a non-linear operating region is transformed to a linear operating region. The following algorithm guarantees feasibility at each iteration (Zoutendijk, 1960; Mangsarian, 1969):

Problem: $\min_{m} \Psi(m, y_2, u_1, d_s)$

subject to $g(m) \leq 0$

where $g_i(m)$ i=1..dim(m) are assumed to be linear.

It is assumed the number of constraints is less than or equal to the number of inputs, dim(m). Otherwise the set of active constraints is found and the lower level regulators have their inputs controlling the active constraints, see Garcia and Morari (1984).

A further assumption is that the constraint functions g_i are known. If not they can be identified in the same way that the objective function is identified (see section "On-line Objective Function Identification" on page 16).

At an operating point m(1), a direction q_1 which points towards the interior of the feasible region and decreases the objective ψ, is given by the solution of the following linear program:

$$\min \varepsilon$$

$$\varepsilon \; q$$

such that

$$q^T \nabla_m P \; (m(1)) < \varepsilon$$

$$q^T \nabla_m g_J(m(1)) < \varepsilon \; e$$

$$|q_i| < 1 \quad i=1..,dim(m(1))$$

$$e = (1 \; 1 \; \ldots \; 1)^T$$

where

$\nabla_m P \equiv$ gradient of objective function

$\nabla_m g \equiv$ Jacobian matrix of constraints

$q \equiv$ constrained to avoid unbounded solutions

$J \equiv \{j \mid -\sigma_1 < g_j \ (m(1) \leq 0, \ \sigma_1 > 0\}$ = set of σ-active constraints at $m(1)$

The new operating point $m(l+1)$ is found by selecting a stepsize μ_1 along q_1, which is usually chosen as the maximum positive value μ such that

$$m(l+1) = m(l) + \mu \, q_1. \quad \ldots\ldots\ldots \quad (17)$$

lies inside the allowed operating region. The existence of q_1 and μ_1 is guaranteed if g_J satisfies certain qualification, which we assume here to be filled (Mangasarian, 1969).

Provided the underlying regulators are chosen well the constrained and unconstrained optimization algorithms are fairly straightforward numerical techniques. Their performance depends on the quality of the extracted steady state information, namely the objective function gradients and the constraint gradients.

3.6 SUMMARY OF DESIGN PROCEDURE AND OUTLINE OF ALGORITHM

The important design considerations are given. They are not hard and fast rules but rather pointers when doing a simulation study for a specific plant. The optimizing regulator algorithm should not be highly sensitive to the choice of design parameters but there is always room for experimentation and refinement.

$q \equiv$ constrained to avoid unbounded solutions

$J \equiv \{j \mid -\sigma_1 < g_j \ (m(1) \leq 0, \ \sigma_1 > 0\}$ = set of σ-active constraints at $m(1)$

The new operating point $m(l+1)$ is found by selecting a stepsize μ_1 along q_1, which is usually chosen as the maximum positive value μ such that

$$m(l+1) = m(l) + \mu \, q_1. \quad \ldots\ldots\ldots \quad (17)$$

lies inside the allowed operating region. The existence of q_1 and μ_1 is guaranteed if g_J satisfies certain qualification, which we assume here to be filled (Mangasarian, 1969).

Provided the underlying regulators are chosen well the constrained and unconstrained optimization algorithms are fairly straightforward numerical techniques. Their performance depends on the quality of the extracted steady state information, namely the objective function gradients and the constraint gradients.

3.6 SUMMARY OF DESIGN PROCEDURE AND OUTLINE OF ALGORITHM

The important design considerations are given. They are not hard and fast rules but rather pointers when doing a simulation study for a specific plant. The optimizing regulator algorithm should not be highly sensitive to the choice of design parameters but there is always room for experimentation and refinement.

1. Plant sampling interval T_s. This is usually chosen as fast as is necessary considering the dominant plant time constants.

2. Define a safe plant operating region and choose suitable constraints to describe the region.

3. Decide on an objective function or set of objective functions. The choice is limited by the following two requirements:

   a. It must have a single extremum in the operating region.

   b. It must be calculated on-line from plant measurements to give an instantaneous value.

4. Low level regulators: Choose between all existing approaches to regulatory process control. This includes analog or digital control, SISO and MIMO, classical and modern control theory as well as adaptive control. Where the purpose is to regulate the plant at set points in the face of fast disturbances as well as making the plant operating region linear and preventing constraint interaction.

5. Form of objective function model: Section "Choice of Model" on page 16 discusses the choice between a linear or non-linear model, the model degree and the inclusion of measurable disturbances in the model.

6. Estimator: It is only under special circumstances that anything other than the RLS estimator (section "Estimation of Model Parameters" on page 20) should be considered. Select a modification to the RLS estimator so it can track plant changes.

7. Bandwidth of objective function model: This will determine the digital filter design (section "Coping with Deterministic Disturbances and Plant Noise" on page 24) as well as the estimator sampling rate. This can be greater than or equal to T_s.

8. Deterministic disturbances: If there are deterministic disturbances in the estimator bandwidth then they must be included in the model (see section "Coping with Deterministic Disturbances and Plant Noise" on page 24).

9. Relative deadzone design: Apply a deadzone to the estimator in case of bad model plant match and resulting estimator divergence (section "Coping with Modelling Error within the Bandwidth of Interest" on page 29).

10. Optimization update time T_{opt}: Time period between setpoint changes and/or direct plant input changes. The factors considered are:

    - Rate of convergence of the estimator
    - Frequency of the significant persistent disturbances d_s

11. The stepsize of the plant moves (section "On-line Optimization" on page 33) is a convergence parameter and should be, as a starting point, chosen conservatively.

The difficult part is defining the problem the optimizer has to solve and this includes the objective function formulation and choice of underlying regulators.

A breakdown of the algorithm given in Figure 5 on page 41 shows the functional structure of the adaptive optimizing regulator.

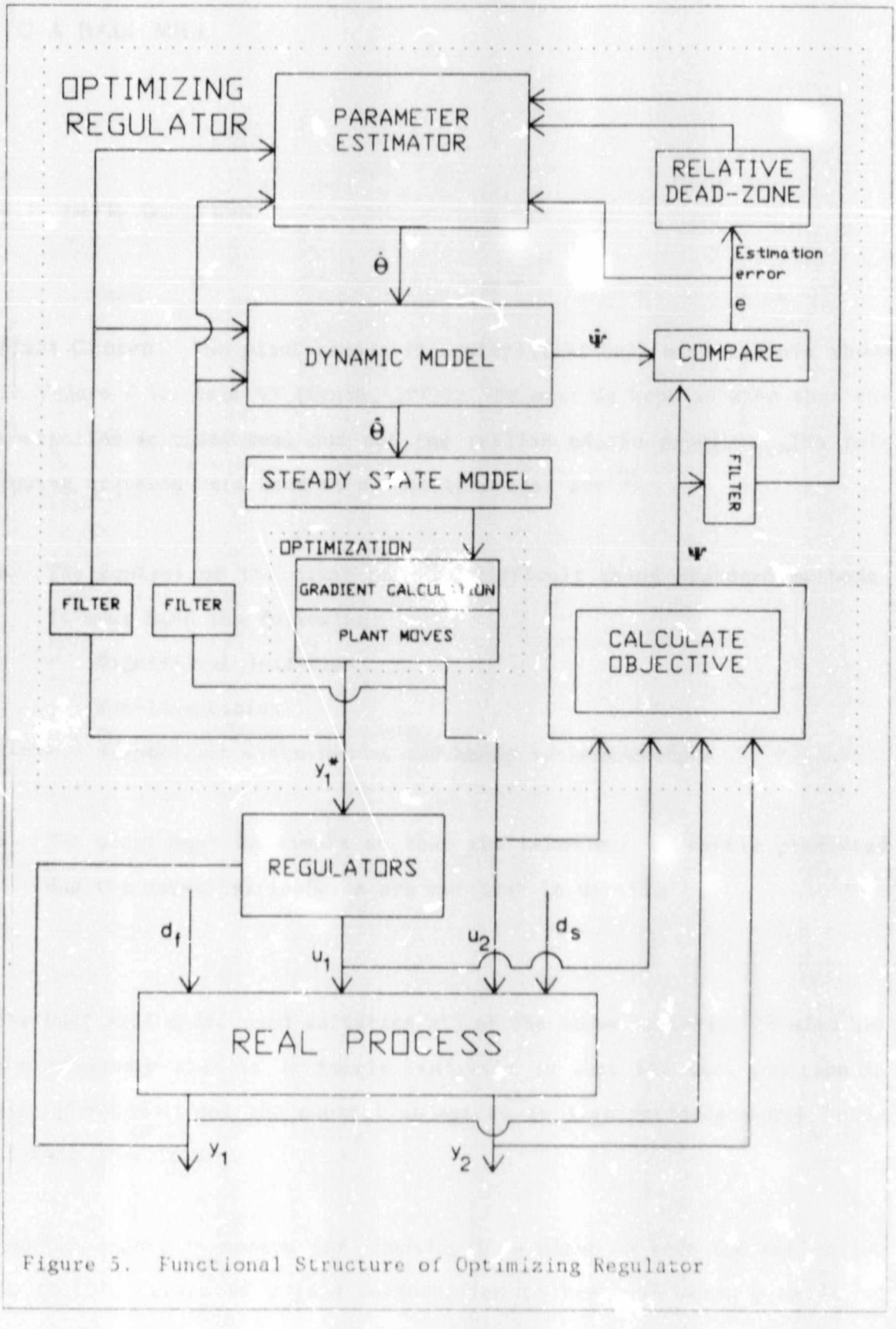


Figure 5. Functional Structure of Optimizing Regulator

4.0 CASE STUDY: APPLICATION OF ADAPTIVE OPTIMIZER THEORY TO A BALL MILL

4.1 INTRODUCTION

Plant Chosen: The plant chosen is a simplified ball mill circuit shown in Figure 6 on page 43 (Lynch, 1977). It must be kept in mind that the controller is under test and not the realism of the problem. The following criteria were used to select this case study:

- The control of the plant must be difficult using standard methods. It must have the following:
  - Significant dead-time
  - Non-linearities
  - Significant disturbances and hence optimum changes

- The plant must be simple so that the behaviour is easily predicted and the essential features are not lost in detail.

The ball mill model used satisfies all of the above criteria. It also has the advantage that it is fairly realistic in that the cost function is easily measured and the control objective is straightforward and intuitively justified.

Another important reason for choosing this plant to test the controller is that it serves as a good introduction to the real world problem of autogenous mill control, which is tackled later.

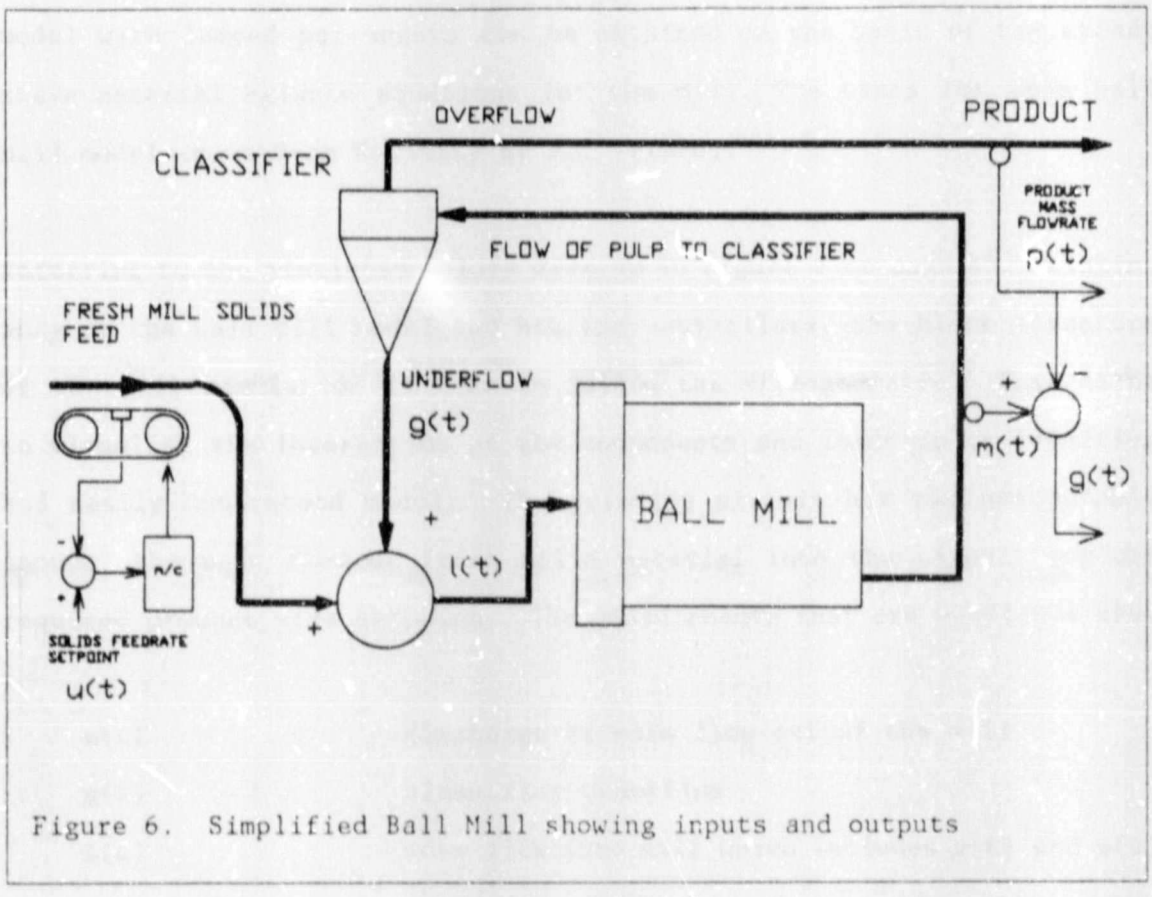


Figure 6. Simplified Ball Mill showing inputs and outputs

Control Objective: The control objective is to maximise the mill mass throughput given the constraint on size of product.

Simulation Environment: The simulations were done on a HP 9000 series 300 computer operating under UNIX. The language used was PASCAL which provided a well structured and hence versatile program. The program listing is given in appendix D.

4.2 BALL MILL MODEL

A ball mill and classifier can be modelled accurately using a mechanistic approach. This leads to a distributed-parameter system that has difficult

equations to handle and is cumbersome for control purposes. An approximate model with lumped parameters can be obtained on the basis of the steady state material balance equations for the mill. The basis for this ball mill model comes from Kevicżky et al. (1976).

Referring to the simulation block diagram in Figure 7 on page 45. Looking only at the ball mill model and not the controllers, the block structure of the mill simulation is seen to follow the mill geometry. This helps to visualise the interaction of the components and leads to an intuitive and easily understood model. The grinding circuit has two manipulable inputs, the mass flow of fresh solid material into the circuit and the required product size setpoint. The measurements that are available are:

m(t)	discharge or mass flow out of the mill
g(t)	classifier underflow
l(t)	mass flow into mill which includes g(t) and u(t)
p(t)	mass flow of final product or classifier overflow

The mill can be seen as an integrator, where l(t) is integrated to give the mill fractional filling f(t). The only other significant dynamics are the mill and the classifier pure delays. If the material classifier is a hydrocyclone then its delay is negligible compared with the mill pure delay.

The discharge of material out of the mill m(t) is modelled only as a function of the fractional filling f(t). All other factors that affect discharge rate not as significantly, such as solids size distribution, pulp density, ball size etc. are lumped together and treated as a disturbance. The discharge function d(f(t)) is simply modelled as the flow of a head of liquid over a semi-circular weir, and this gives a quadratic

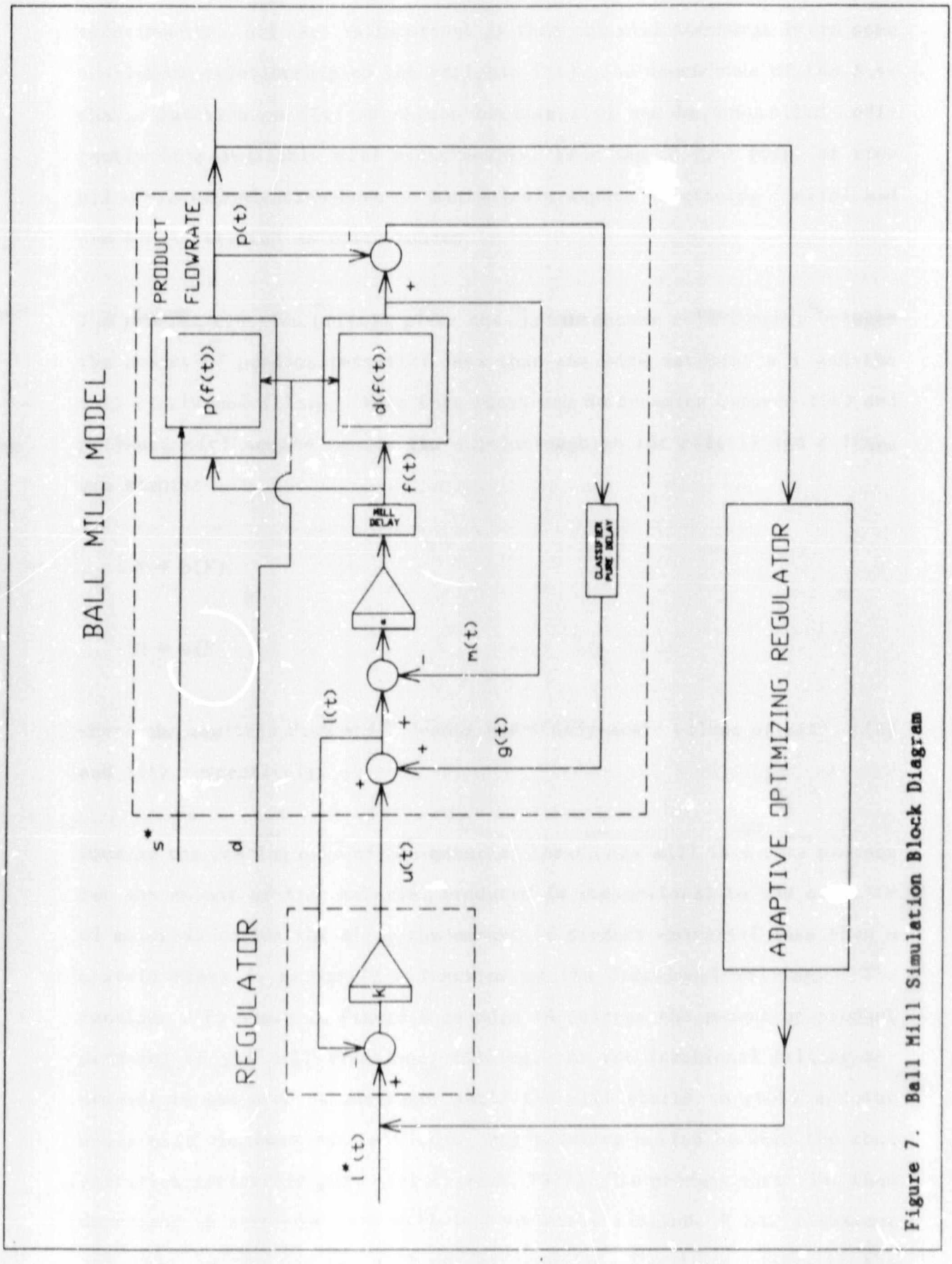


Figure 7. Ball Mill Simulation Block Diagram

relationship. All that is important is that the mill discharge bears some non-linear relationship to the variable f(t). The dependence of the discharge function on f(t) is chosen because f(t) can be controlled indirectly from available mill measurements. From the control point of view all other dependencies have no manipulable inputs to attempt control and are simply treated as disturbances.

The product function p(f(t)) gives the instantaneous relationship between the amount of product (material less than the size setpoint s^{*}) and the mill fractional filling. Note that there are no dynamics between f(t) and p(t) and m(t) so the steady state relationships for p(f(t)) and d(f(t)) are simply:

$$P = p(F)$$

$$M = d(F)$$

Where the capitals P, M and F denote the steady state values of p(t), m(t) and f(t) respectively.

Because the wearing down of the material inside the mill is a rate process ie. the amount of fine material produced is proportional to the quantity of material inside the mill, the amount of product (material less than a certain size) is primarily a function of the fractional filling. The function P(F) shown in Figure 8 on page 48 relates the amount of product produced to the mill fractional filling. As the fractional filling increases so the product increases until the mill starts to choke and the whole mill contents rotate without any relative motion between the rock particles (epicyclic gear effect Lynch, 1977). The product mass flow then decreases to zero when the mill is completely clogged. P has a maximum and this is the desired mill operating point. The disturbances to the product mass flow rate are large and varied, the important disturbances

are; size distribution of mill contents, physical size characteristics and ball and liner wear. These may change the position of the maximum of P(F) significantly and this is simulated by changing parameters in P(F). To include the whole family of relationships P(F) write P as P(F,d) where d represents the combined effect of all possible disturbances at that time. The product function P(F) is chosen as a fourth order polynomial. This is motivated on the grounds that a polynomial of this order models mill power draft as a function of F accurately (Hinde, 1977) and mill product mass flowrate is proportional to mill power draft for a certain product size and input feed size distribution.

The details of the non-linear functions d(F) and p(F) and exactly how the function parameters are changed to simulate the disturbances is given in appendix C.

It is assumed that the classifier is ideal and the material less than a certain size p(t) is separated and the remainder or classifier underflow, g(t) recycles into the mill. More realistic modelling involves giving each size fraction of the mill discharge or pulp a probability of reporting to the overflow p(t) rather than the underflow g(t). This level of detail is not required because a simple input output model is built and all the detailed factors that change the optimum are lumped together as a plant disturbance. The classifier cut size is controlled by, for example, changing the discharge density by water addition if a hydrocyclone classifier is used, or for a centrifugal cement mill classifier, by changing the speed of rotation. This simulation assumes the existence of a regulator to control the product separation to a setpoint.

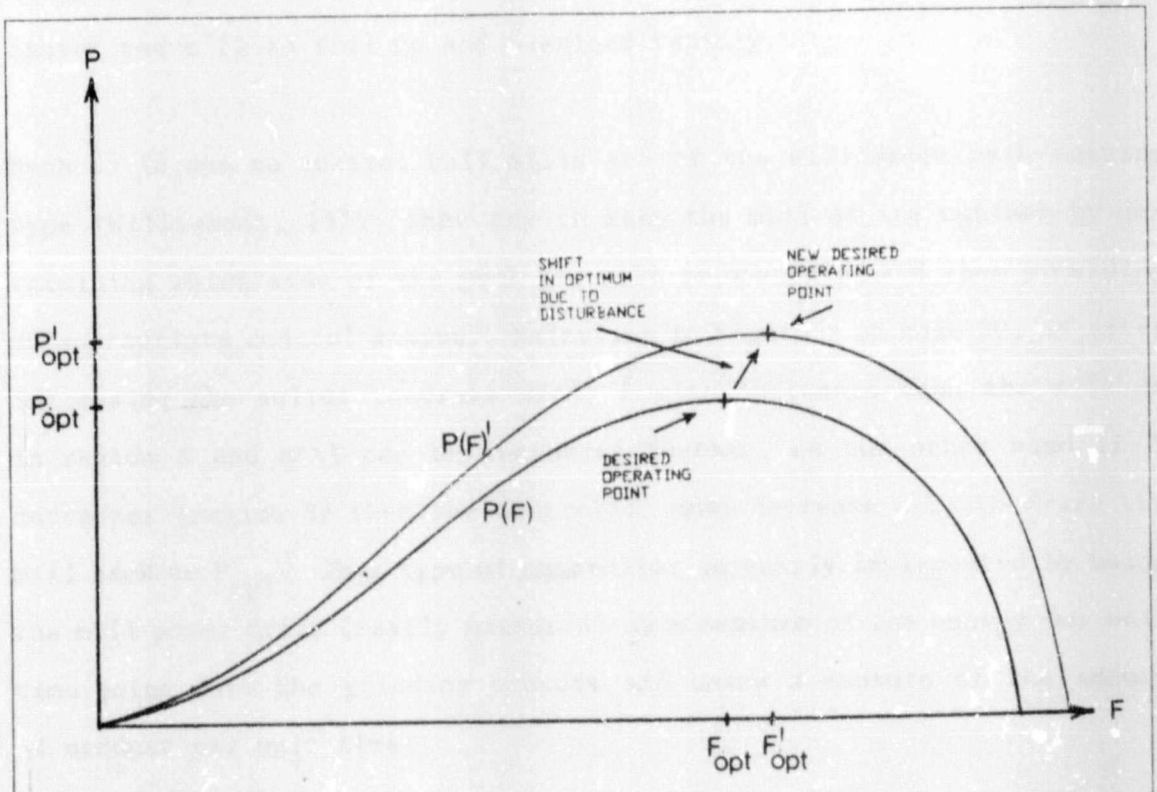


Figure 8. Relationship between P and F showing the desired operating point and a shift in this point due to a disturbance

4.3 GRINDING CIRCUIT CONTROL

It is known that the optimum operating point (Figure 8 on page 48) for maximum product has to be approached cautiously. A mill solids feedrate higher than that which gives the optimum will cause the mill to go into an unstable operating region and the mill will overload rapidly. An understanding as to why the region $F>F_{opt}$ is unstable is clear from Figure 7 on page 45. An increase in F above F_{opt} causes a decrease in P (relationship given in Figure 8) which will increase G more than M. For the increased u(t) (solids feedrate) to give $F>F_{opt}$ a positive feedback condition exists and there is no steady state. This can also be understood from a physical point of view. If the fresh solids feedrate is increased beyond that point that gives most efficient grinding then the increased

material input, in addition to less product being produced and extracted causes the mill to fill up and overload rapidly.

...ods in use to control ball mills are of the Williamson peak seeking ...illiamson, 1975) that try to keep the mill at its optimum by determining which side of the peak the mill is operating and then providing an appropriate control action. Referring to Figure 9 on page 50, if after increasing the solids feedrate u(t), P also increases then the mill is in region A and u(t) can be increased further, on the other hand if P decreases (region B) then the controller must decrease u(t) to drive the mill back to P_{opt}. This type of controller is easily implemented by using the mill power draft (easily measured) as a measure of the energy per unit time going into the grinding process and hence a measure of the amount of product per unit time.

These controllers are satisfactory provided the P,F relationship is not influenced significantly by disturbances and the curve remains approximately stationary. Figure 9 on page 50 shows a large disturbance changing the curve. The controller will detect a decrease in P of magnitude x and will subsequently decrease f(t), which drives the plant away from P_{opt}. It will confuse the old region B with the new region A'. For large frequent disturbances and a shifting optimum the controller action may be inappropriate. Also, due to the operating point being close to an unstable region the controller may overload the mill.

Now that the problem has been explained we proceed with a solution in the form of the adaptive optimizing regulator proposed in the previous two chapters. Knowing that the optimizer needs time and is designed for low bandwidth operation a low level regulator is introduced to cope with high frequency disturbances.

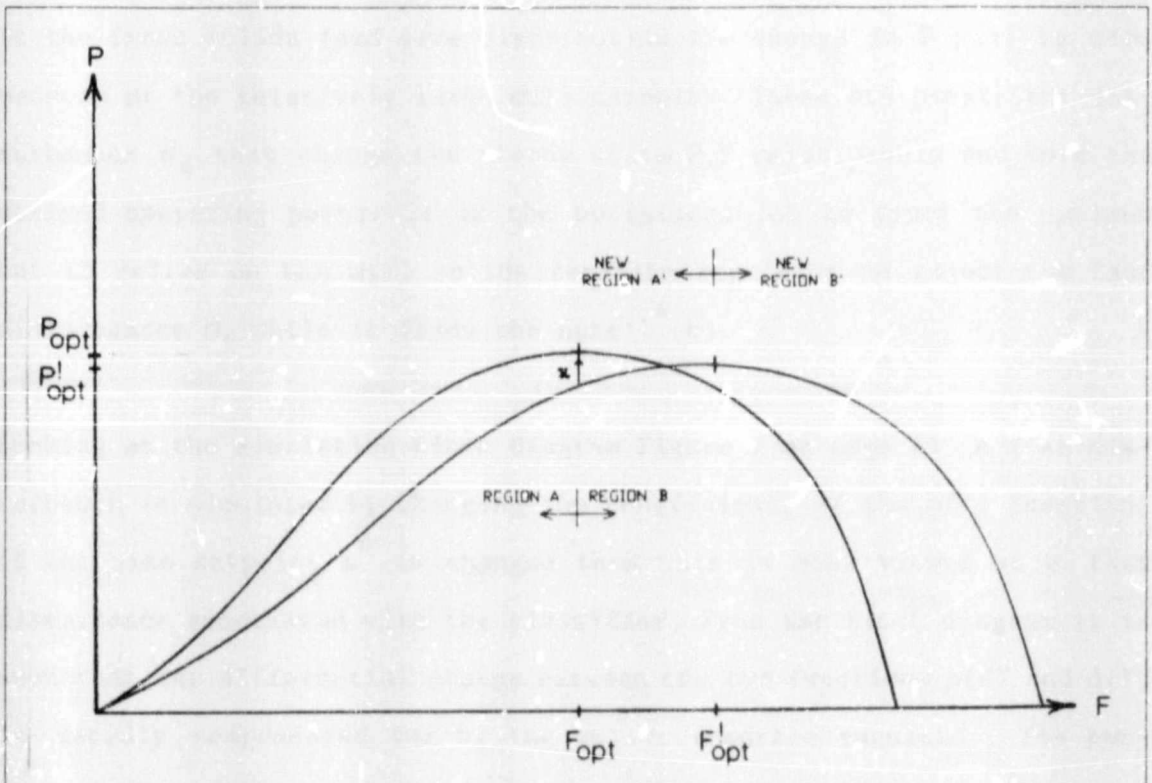


Figure 9. Steady State relationship between P and F showing regions of operation

4.4 MILL SOLIDS FEEDRATE REGULATOR

Keviczky et al. (1976) propose and motivate a mill solids feedrate regulator. They identify l(t) (material flowing into the mill ie.u(t)+g(t)) rather than u(t) (fresh solids feedrate) as the important measured variable to be controlled to a setpoint. Figure 7 on page 45 shows a simple integral action regulator that adjusts fresh solids feedrate u(t) to keep l(t) at $l^{*}(t)$. Their motivation is based on l(t) being a more direct control of conditions in the mill and thus of p(t). Also the regulator makes l(t) immune to the disturbances associated with the classification process. These are fast disturbances d_f when compared with the disturbances due to changing fresh solids feed characteristics (for example particle size distribution and physical properties of the rock) and hence

conditions inside the mill. Even if, for example, there is a step change in the fresh solids feed size distribution the change in P will be slow because of the relatively large mill capacity. These are persistent disturbances d_s that change the steady state P,F relationship and move the desired operating point. It is the optimizers job to track the optimum but it relies on the mill solids feedrate regulator to reject the fast disturbances d_f while it finds the next $l^*(t)$.

Looking at the simulation block diagram Figure 7 on page 45. A fast disturbance is simulated by changing the coefficients of the p(F) function. If the size setpoint s^* is changed then this is also viewed as a fast disturbance associated with the classifier. From the block diagram it is seen that any differential change between the two functions p(F) and d(F) is rapidly compensated for by the solids feedrate regulator. The persistent disturbances are simulated as a change in both the parameters of the functions p(F) and d(F). This makes sense, because changing conditions inside the mill (due to d_s) will change both the product function p(F) and the discharge function d(F). The discharge rate and the amount of product produced are intrinsically linked. The next section deals with the problem the adaptive optimizer has to solve and the necessary algorithms.

4.5 BALL MILL ADAPTIVE OPTIMIZER

4.5.1 PROBLEM FORMULATION

The problem to be solved by the adaptive optimizing regulator is:

$$\max_{l^*(t)} \; p(F(l^*),\, d_s)$$

where the product size regulator prevents violation of the size constraint and d_s represents the persistent disturbances.

The fresh solids feed rate regulator prevents the mill going into an unstable region due to fast disturbances d_f

To solve this problem the function $p(F(l^*))$ needs to be known. It must be identified on-line because it is changing all the time due to d_s .

4.5.2 OPTIMIZER

The simplest possible working solution is presented without complicating the algorithm with refinements.

Objective Function Model: Second order linear DARMA model with deadtime d and no measurable disturbance included.

$$(1 + a_1q^{-1} + a_2q^{-2})p(t) = q^{-d}(b_0 + b_1q^{-1})l^*(t) + c \quad \ldots\ldots\ldots (1)$$

Author Bleloch Mark John
Name of thesis An Adaptive Optimizing Regulator For An Autogenous Mill. 1987

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.