LAKEHEAD UNIVERSITY

# AN ADAPTIVE OPTIMIZER BASED ON THE SEQUENTIAL SIMPLEX METHOD APPLIED TO THE CONTROL OF A THERMO-MECHANICAL PULPING SCREENING ROOM

by

M.V. VENHOLA

A THESIS

SUBMITTED TO THE FACULTY OF ENGINEERING

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE IN ENGINEERING

in

CONTROL ENGINEERING

FACULTY OF ENGINEERING

LAKEHEAD UNIVERSITY

THUNDER BAY, ONTARIO

AUGUST, 2004

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

*Your file Votre référence*
*ISBN: 0-494-10676-X*
*Our file Notre référence*
*ISBN: 0-494-10676-X*

## Abstract

An adaptive optimizer is designed for control of multi-variable processes where the operating characteristics of the plant are time varying. Based on the original sequential method for static applications, the modified direct search method is made adaptive by allowing only limited expansion and contraction and by re-measuring all points of a simplex when improvement of the simplex position is not possible by reflection of the vertices, and when expansion or contraction occurs. In one method, suitable for performance functions that have a known optimum value, i.e., sums of squares of errors from target values, expansion and contraction is determined by a threshold value of the performance function. In a second method, expansion and contraction is determined by ranking the newest measurement against the historical measurement values in the simplex, similar to the Nelder-Mead method.

The optimizer was tested on simulated two by two multi-input multi-output first order plus delay processes. It worked best in the absence of dynamics, but could give acceptable performance where the delay and time constants were less than five measurement sample times.

The adaptive optimizer is applied also to a simulated model of a thermo-mechanical pulping screening room to control the accepts fibre distribution, using the main line and reject screen volumetric rejects ratios. Two types of screens are modeled: screens with smooth holed apertures in the screening baskets, and screens with slotted apertures. The adaptive optimizer is capable of controlling the accepts fibre distribution over limited ranges of disturbances in mean fibre length. The control range of the slotted screen is narrower than that of the holed screen because of its flatter fractionation profile. Transport and capacity lags do not have a great impact because most of the fibre flow follows the main accepts line.

## Acknowledgements

I thank Dr. Allan Gilbert, Lakehead University, for his inevitable and invaluable feedback in all aspects of this project, and his patience in supporting the re-education of a former industrial colleague in things that an electrical engineer was probably never meant to appreciate or fully comprehend. I would also like to thank former colleagues Garth Pattyson and Henry Blekkenhorst, Bowater Mill, Thunder Bay, for their help with operational process information and control system infrastructure, respectively, towards the potential of moving to a mill trial. I am ever thankful to my wife, Kathleen, for her continual faith and support in this later-in-life hiatus, which has caused her significant challenges at home.

## Table of Contents

## List of Tables

## List of Figures and Illustrations

# Chapter One: Introduction

## 1.1 Overview

This work develops an optimization method to be applied on-line as the controller in a control system that will attempt to maintain a process at an optimal operating point. The optimal operating point is determined as being a function of the process output, or measured, variables, and possibly the process input, or manipulated, variables, which are adjusted by the optimizing controller. This function is specified as a calculated value to be minimized or maximized and is known as the objective function or the performance index. The optimizing controller has no process knowledge, other than the process output variables, and thus no process model is used in the controller. The optimization method is classed as a *direct search* method, and is based on the sequential simplex method developed by Spendley, Hext and Himsworth [1,19]. The optimizer is adaptive, in the sense that it must adapt to an optimal operating point that is expected to be time varying, and also because the plant may include process dynamics.

A simplified on-line multivariable process optimization schematic is shown in Figure 1.1. The process has multiple input and output variables. The process output variables may require conditioning by sophisticated measurement sensors for data simplification and reduction. These measured variables, and possibly the process input, or manipulated variables, are combined functionally into a performance measure or index, i.e., the objective function. The objective function calculation can also include targets, or setpoints, for any of the measured variables. For example, the performance index may be a function of errors, i.e. the difference between setpoint and measurement. The objective function produces one value, to be minimized or maximized. The optimizing controller moves the manipulated variables in attempting to produce an optimal objective function value, i.e., a minimum or a maximum. Since this is an on-line system to be applied to an industrial process, process dynamics, in the form of gain, bias, lags and delay, can be expected to affect the loop.

When the performance index function includes target, or setpoint variables, the adaptive optimizer can be seen to act as a conventional process controller. In this sense, it might be compared to Proportional – Integral – Derivative (PID) mode controllers, which are known to provide a minimum variance performance. The adaptive optimizer has not been directly compared to PID controllers in performance testing, since it is to be applied more generally to non-minimum variance performance functions. It will be noted in the adaptive optimizer testing in chapter two that, in general, the optimizer cannot provide the same performance as a conventional PID controller, when the performance index function is minimum variance.

A targeted industrial environment for the adaptive optimizer is shown in Figure 1.2. In a Thermo-Mechanical Pulping (TMP) screening room process, several pulp screens are manipulated by their *rejects ratio*, which is the volumetric ratio of one output flow, i.e., the rejects flow, to the screen input, or feed, flow. A second screen output flow is the accepts flow, not requiring further pulp refining, while the rejects stream requires further refining. The accepts stream pulp quality is measured by Pulp Quality Monitoring (PQM) sensors, which provide data reduction to a small number of measurements in the form of statistical values for various pulp fibre length categories. These measured properties have a target value for each fibre length category and the performance index is a function of the errors for each fibre length category, i.e., the sum of error squares. This objective function value, or performance index, is to be minimized, where a zero error is ideal.

In manipulating the screen reject ratios towards producing a lower performance index value, the optimizer moves a number of screen reject ratios at a time using a direct search method based on a short history of previous moves, which are stored in the form of a geometric simplex. The adaptive optimizer uses a regular simplex, where each side is of equal length. In particular, where two screen ratios are manipulated, the simplex is formed as an equilateral triangle, being a history of the last three best moves, ranked from worst to best. The adaptive optimizer attempts to improve the overall position of the simplex by replacing the higher valued, or worst, of the previous positions of the manipulated variables, as indicated by the history of performance function values for each move, which are stored in the simplex along with the manipulated variable positions. The general mechanism is by reflection of the non-best points towards the direction of the best point, as indicated by the corresponding performance function values. Thus the mechanism is direct search in nature.

Chapter one of this work introduces optimization techniques, including analytical optimization using derivatives and applied to deterministic models, and direct search methods, as applied to functions where the models are unknown. Chapter two introduces the design of the adaptive optimizer. The optimizer is best used where the performance index function has concavity or convexity, which result in global or local extrema. An example of such a performance index function is the unconstrained minimum of the sum of error squares from a series of performance index targets. The optimizer can be applied, as an alternative to other multiple-input and multiple-output process control schemes, to processes of certain dynamic classes. Chapter three discusses an optimization problem for Thermo-Mechanical Pulp (TMP) screening processes in the pulp and paper industry. The screening room of a typical TMP plant is simulated according to recently developed models. Several performance index functions, which are suitable for use with the adaptive optimizer, are measured over the complete range of process input variable manipulation.

## 1.2 The Optimization Problem

An optimization problem, wherein the application of a manipulated or decision variable evaluation leads invariably to a specific outcome, is termed deterministic in nature. Deterministic models generally have the following characteristics:

1. Decision or manipulated variables – controllable input variables to the system.
2. Objective function – ranks the desirability of the results of decision variable applications to the system.

3. Constraint functions – conditions imposed on the decision variable in restriction application range.
4. Feasible solutions – any solution to the model that satisfies the constraints but does not necessarily minimize or maximize the objective function.
5. Optimal solutions – a feasible solution that satisfies the constraints and also minimizes or maximizes the objective function.

Optimization techniques can be characterized according to whether the physical plant or process is modeled by an equivalent mathematical function, or whether the model is not used or is unknown. Models may be linear or non-linear in nature.

Some important categories of optimization techniques are as follows:

1. Unconstrained optimization – analytical method of calculus.
2. Constrained optimization – Lagrange multipliers.
3. Optimization of linear models – linear programming.
4. Dynamic programming - finding optimal paths in a multi-stage process.
5. Direct search methods.

The first category of optimization techniques, where a plant model is required, is reviewed in the next section for the purpose of comparing analytical techniques to the methods of direct search. Principles of convexity are important in optimization methods that use analytical techniques and these same principles have a bearing on the success of direct search methods. The other optimization categories are not directly relevant to this paper.

## 1.3 Unconstrained Analytical Optimization Using Deterministic Models

Optimization methods for multiple variable functions have often used derivative functions in their search for local or global extrema. The concept of a *convex set of points* is important for optimization since a local or global minimum can be found in such a region. A useful property arising from this concept is that a set of points satisfying the relation $x^T H(x)x \leq 1$ is convex, if the Hessian matrix H(x) is a real symmetric positive-semidefinite matrix. H(x) is a symbol for $\nabla^2 f(x)$, the matrix of second partial derivative of f(x) with respect to each $x_i$, $H(x) \equiv \nabla^2 f(x)$. For a multiple variable function, the nature of convexity can be evaluated by examining the eigenvalues of H(x). For example, given a quadratic function in two variables, the geometric interpretation of the function using the eigenvalue characterization may be circular, elliptical, hyperbolic, linear or parabolic. The geometries may form a convex (or concave) region or they may form 'degenerate' functions, for which no finite minimum or maximum or non-unique optima occurs.

For a positive-definite symmetric Hessian matrix, the eigenvectors form an orthonormal set (i.e. perpendicular to each other in two dimensions) and these eigenvectors correspond to the directions of the principal axes of the contours of f(x). An efficient minimum search routine selects a search direction that generally follows the axis of a valley. The valley lies in the direction of the eigenvector of the Hessian matrix (the smaller of the two eigenvalues in two dimensions).

Using a Taylor series expansion about a presumed extremum x* of a given function f(x), $f(x) = f(x^*) + \nabla^T f(x^*)\Delta x + \frac{1}{2}(\Delta x^T)\nabla^2 f(x^*)\Delta x + \ldots$, it can be seen (from the second term) that a necessary condition for a minimum or maximum of f(x) is that the gradient of f(x) equal 0 at x*, i.e. $\nabla^T f(x^*) = 0$, and therefore x* is a stationary point. The third term establishes the character of the stationary point as a minimum, maximum or saddle point. Sufficient conditions to guarantee a unique minimum or maximum is that H(x*) be either positive-definite or negative-definite, respectively. Necessary conditions are that f(x) is twice differentiable at x*, that a stationary point exists at x* and that H(x) must also be defined at x = x* [2].

### 1.3.1 Unconstrained Analytical Optimization - One Dimensional Search Techniques

Numerical optimization demands a good technique for functions of just one variable because techniques for unconstrained and constrained optimization problems usually make repeated use of one-dimensional searches.

An analytical method of finding x* at the minimum of f(x) is to set the gradient of f(x) to zero and solving the resulting equation for x*. The value of the second derivative function at x* can determine if x* is really a minimum (as opposed to a maximum).

The following are some prominent one-dimensional search techniques.

**Newton's method** [3] numerically solves $f'(x) = 0$ as follows: $x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}$. At each iteration, $f(x^{k+1}) < f(x^k)$ must be true for a minimum. Under some conditions, including a poor initial estimate, the method may converge slowly or not at all.

The **quasi-Newton (secant) method** [3] approximates $f'(x)$ as a straight line. It starts by using two points spanning the interval of x at which the first derivatives are of opposite sign. At each iteration, the two new points are the calculated approximation of x* and either of the previous points, whichever is required to maintain the oppositely signed bracket derivatives. Convergence is slightly slower than the original Newton's method.

Another class of one-dimensional minimization attempts to locate x* by extrapolation and interpolation using **polynomial approximations** [3] as models of f(x). Both quadratic and cubic approximations have been used using the approximation functions only or both the functions and derivatives. In functions where the first derivative is continuous, these methods are efficient.

### 1.3.2 Unconstrained Analytical Optimization - Multi-Dimensional Search Techniques

In minimizing a function of several variables, the general procedure is to calculate a search direction and then reduce the value of f(x) by taking one or more steps in that search direction. Each step can be seen as extending a vector in the direction of the search by a scalar. This constitutes a unidirectional or line search. The function minimum must be bracketed by this search and one of the above one-dimensional methods can be used to find the scalar that provides the minimum function value for this search direction.

*Conjugate* search directions have proven to be more effective than orthogonal or univariate directions. Two directions $s^i$ *and* $s^j$ are said to be *conjugate* with respect to a positive-definite matrix Q, if $(s^i)^T Q(s^j) = 0$. Orthogonality is a special case of conjugacy, where Q = I, the identity matrix, and $(s^i)^T (s^j) = 0$.

The following are some prominent multi-dimensional search techniques.

The **first derivative method** [3] is as follows. A good search direction reduces (for minimization) the objective function so that $f(x^{k+1}) < f(x^k)$. Such a search direction is called a descent direction and satisfies the following condition at any point: $\nabla^T f(x)s < 0$.

The **steepest descent method** [3] is as follows. The gradient is the vector at a point **x** that gives the local direction of the greatest rate of increase in f(x). It is orthogonal to the contour of f(x) at **x**. For minimization, the search direction can be the negative of the gradient or 'steepest descent', i.e. $s^k = -\nabla f(x^k)$. In steepest descent the new point is given by: $x^{k+1} = x^k + \alpha^k s^k = x^k - \alpha \nabla f(x^k)$, where $\alpha$ is a scalar of step length. The iterations are applied until the elements of the gradient vector approach zero. The step size at each iteration ($\alpha^k$) is determined by a one-dimensional line search. The gradients, and therefore the search directions, at points $x^k$ *and* $x^{k+1}$ are orthogonal. For functions that are not perfectly scaled, i.e., a quadratic function with interactive terms, large efficient steps are taken early in the search, but more smaller steps must be taken later.

The following are conjugate gradient methods.

The **Fletcher-Reeves method** [5] is as follows. The search directions are conjugate – a major improvement over steepest descent without a large increase in computational effort. It combines current information about the gradient vector with that of gradient vectors from previous iterations to obtain a new search direction (a linear combination of the current gradient with the previous search direction). For a non-quadratic function, more cycles are required involving a re-initialization of the search direction every *n* cycles. The method will take longer and may not converge with severely non-quadratic functions.

**Newton's method** [4] makes use of the second-order (quadratic) approximation of f(x) at $x^k$ and therefore employs information about the curvature of f(x) at $x^k$ to identify better search directions than can be obtained by the gradient method. By differentiating the Taylor series expansion of f(x) gives $\nabla f(x) = \nabla f(x^k) + H(x^k)\nabla x^k = 0$ or $x^{k=1} - x^k = \Delta x^k = -[H(x^k)]^{-1} \nabla f(x^k)$. Both the direction and step are specified. While Newton's method usually requires few iterations, it has the disadvantages of requiring the solution of a set of *n* symmetric linear equations and requiring both first and second partial derivatives.

The **Quasi-Newton (BFGS) method** [4] is as follows. This method replaces $H(x^k)$ with a positive-definite approximation $\widetilde{H}^k s^k = -\nabla f(x^k)$, where $\widetilde{H}^k$ is initialized as any positive-definite symmetric matrix (the identity or another diagonal matrix) and is updated after each line search using the changes in **x** and in $\nabla f(x)$ over the last two points, as measured by the vectors $d^k = x^{k+1} - x^k$ and $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$.

The **Broyden, Fletcher, Goldfarb and Shanno** [10,11,12,13] update formula for $\widetilde{H}^k$ is $\widetilde{H}^{k+1} = \widetilde{H}^k + \frac{y^k (y^k)^T}{(d^k)^T y^k} - \frac{(\widetilde{H}^k d^k)^T (\widetilde{H}^k d^k)^T}{(d^k)^T \widetilde{H}^k d^k}$. If $\widetilde{H}^k$ is positive-definite and $(d^k)^T y^k > 0$, it can be shown that $\widetilde{H}^{k+1}$ is positive-definite. The condition $(d^k)^T y^k > 0$ is always satisfied if f(x) is strictly convex. $\widetilde{H}^k$ is not updated if this condition is not met. For non-quadratic functions, BFGS code usually requires more iterations than Newton's method and may not be as accurate, but each iteration is faster because second derivatives are not required and the linear equation solver is not needed.

## 1.4 Direct Search Methods

The term *direct search* was coined by Hooke and Jeeves [7,19] as "...to describe sequential examination of trial solutions involving comparison of each trial solution with the *best* obtained up to that time together with a strategy for determining (as a function of earlier results) what the next trial solution will be." They implied a preference to not employ traditional analysis techniques as outlined in sections 1.3, and as such, the *direct* *search* methods are generally *derivative-free*, i.e., being methods that neither compute nor approximate derivatives. Direct search methods can be characterized by the use of rank, or order, only when interpreting the performance or objective function, i.e., numerical values are not otherwise directly interpreted [7,19]. The performance index values are only *compared against each other* for the purpose of determining the next operation. The actual values of the performance index are not used in computing the value of the next operational move.

### 1.5 Simplex Search Methods

Simplex search methods use a simple geometric device that guides the search. Spendley, Hext and Himsworth [1,19] proposed the original sequential simplex search method based on the observation that, for a performance function *f(x)*, such that $f : R^n \to R$, it should take no more than *n+1* values of the performance function to identify a downhill or uphill direction. Since *n+1* points determine a simplex, and in the graph of *f(x)* they also determine a plane, a non-degenerate simplex in $R^n$ provides a simple and frugal way to sample the performance function space in a sequential manner. Additionally, if a vertex of the simplex is replaced by reflecting it through the centroid of the opposite face, as shown in Figure 1.3, the result is also a simplex.

A single move in the sequential search method is always based on reflection. The pre-condition is that the vertex points must be ordered as to their respective values of the performance function. After the reflection of the *worst* point, the new reflected point is either accepted as *better* than the previous *worst* point or, if the new point is still the *worst*, the *next worst* point of the original simplex is reflected in an attempt to determine if a move in that direction will achieve a *better* than the previous *worst* point. The overall objective of the search is to continuously improve the *best* point of the current simplex or to accept the convergence of the *best* point to within a certain tolerance limit of the optimal value.

The search may fail to replace any vertex of the simplex. The size of the simplex may prevent further convergence to the optimum value. In this case, subsequent reflections that produce a circling sequence of simplices, as shown in Figure 1.4, can indicate that the simplex is in the neighbourhood of a stationary optimum value for the performance function. Under these conditions, Spendley, Hext and Himsworth suggested that the lengths of the edges adjacent to the *best* vertex be reduced to obtain faster local convergence to an optimum value, or secondly, higher order techniques be used.

Nelder and Mead [8,19] proposed, in a method that is well known and often used today, to optimize the sequential search method by creating additional move types that would accelerate the search. This method has been coded in many custom software applications, and has been included in popular programming packages such as Matlab ™ [14], GNU Scientific Library [15] and Numerical Recipes in C [16]. The latter two packages provide source code in the *C* programming language.

The additional moves would supplement the original reflection move with options that deform the simplex to better adhere to the 'landscape' of the performance function. Nelder and Mead added expansion and contraction moves, as shown in Figure 1.5 through Figure 1.7, where a two dimensional simplex is shown. Normal reflection of the *worst* point (x1W,x2W) would provide a new point for the simplex at (x1R,x2R), providing that that point was better than the *next worst* point. However, the expansion move provides acceleration, by increasing the length of the step from the centroid to the reflected point, as shown in Figure 1.5, so that the new point is (x1E,x2E). Expansion occurs under the conditions where the normal reflected point is better than the *best* point. Contraction moves de-accelerate, by decreasing the length of the step from the centroid to the reflected point, in a step known as outside contraction, or by decreasing the length of the step so that the new point is inside the original simplex. These moves are shown graphically in Figures 1.6 and 1.7, where the new points are (x1CO,x2CO) and (x1CI,x2CI), respectively. Inside contraction is performed if the reflected point is worse than the *worst* point and outside contraction is performed if the reflected point is better than the worst point but not better than the *next worst* point. Nelder and Mead also proposed a shrink move, to be applied when all other moves fail, i.e., in anticipation of local convergence to an optimum value. This step is shown graphically in Figure 1.8, where it can be seen that the simplex shrinks to the new size, keeping the best point as a static vertex.

The Nelder-Mead algorithm enjoys wide use and can be very efficient, but its robustness has been questioned [9], with one potential drawback being that the deformation of the simplex may cause the search direction to become numerically orthogonal to the performance function gradient, resulting in very slow convergence or possible non-convergence.

### 1.6 Sequential Search Methods for Time Variant Systems

The sequential search may fail to replace any vertex of the simplex. However, should the system characteristics change over time and cause a performance function 'landscape' shift, the simplex point values may be out of date and this can cause the same effect of no possible movement for the simplex. Previous optimization methods that use sequential search methods also assume that process measurements, manipulated variables and performance index are at steady state. This may not be true for on-line optimization of industrial plants. More recent work has provided sequential search type methods for adaptation to time variant system functions. However, this later work has not been applied to systems where the process contained dynamics, i.e., where the performance index is not necessarily at steady state when it is evaluated.

Jutan and Xiong [17] have recently provided a method for continuous optimization of a time varying system using a *dynamic simplex method.* In this work, the Nelder-Mead simplex method has been modified and extended to allow tracking of a moving optimum where noise contaminated the measurement data. The algorithm was applied to linear and non-linear drifting optimal functions of variables in two and three dimensions. Results are not shown for step changes in optimal function and the optimal functions are assumed to be at steady state.

Jutan and Xiong constructed an algorithm that re-applies and measures the *best* point to avoid data invalidation through time. The method uses a fixed size simplex to avoid sluggish movement in orthogonal to gradient directions of the performance function. The nominal simplex size is related to the maximum noise level, i.e., the difference in measurements between vertices of the simplex must be larger than the mean noise value. Completing an iteration of the algorithm consists of making a set of successive reflections, as in the original sequential simplex sequence of reflections (Figure 1.4). However, the simplex for the next iteration, or set of reflections, is chosen as the simplex (from the previous successive reflection set) whose points produce the best average performance function value. From an on-line process optimization point of view, where feedback response time, i.e., the manipulated variable application and subsequent process measurement time, is critical, the measurements involving this set of reflections may be costly in terms of time. While the algorithm has been shown to follow a slower moving and drifting optimal value, it has not been shown to respond to a step change in the optimal value, or any system change that might include a step change and dynamics in the form of lag or delay.

Hennings [18], applying optimization to control of a single pulp screen process, also employs re-measurement of the *best* point in two distinct sequential search algorithms. One method is similar to Nelder-Mead, with limits on simplex sizes through expansion or contraction. This prevents the simplex from being deformed in such a way as to have its search direction to be orthogonal to the direction of steepest descent. On contraction, the *best* point is replaced, for efficiency in re-application of the performance measure. The second method uses a variable size simplex, without simplex deformation, and where the *best* point is again rejected on contraction. Hennings applies his algorithm to step changing optimal functions only and does not include any drifting or ramping optimal function variations. Hennings suggests that noise remains a problem for his algorithms, which are only applied to steady state conditions.

In general terms, the work in this thesis is directed at the same on-line environment targeted by Jutan/Xiong and Hennings. However, the environment includes testing of the adaptive optimizer, presented here in Chapter Two, with time varying optimal functions of both ramping and step changes. Measurement noise has been added, and system dynamics, in the form of first order lags, are included, where the process may have steady state gain and bias. The adaptive optimizer presented here is based largely on the original sequential simplex method proposed by Spendley, Hext and Himsworth. The simplex is not deformed, as per Nelder-Mead, and thus is a regular simplex, having equal length sides.

The simplex grows and shrinks, within limits, upon detection of 'closeness' to the optimal function value, through one of two methods of determining this 'closeness' to the optimal value. One contraction and expansion method is based on a comparison of the performance function value to a performance function value limit, or threshold. The second method of contraction and expansion is based on ranking the newest performance function value relative to the performance function values held in the simplex, i.e., a method similar to the Nelder-Mead contraction method.

The optimal value is assumed to be capable of moving immediately in a drifting or a step-changing manner, and therefore a characteristic of the adaptive optimizer is the continuous re-measurement of the simplex under steady state conditions, and when the simplex grows or shrinks. Tests in systems of low dimension have indicated that the adaptive optimizer is effective in tracking moving optimal values in a variety of conditions, as outlined above.

Figure 1.1 On-Line Multivariable Process Optimization

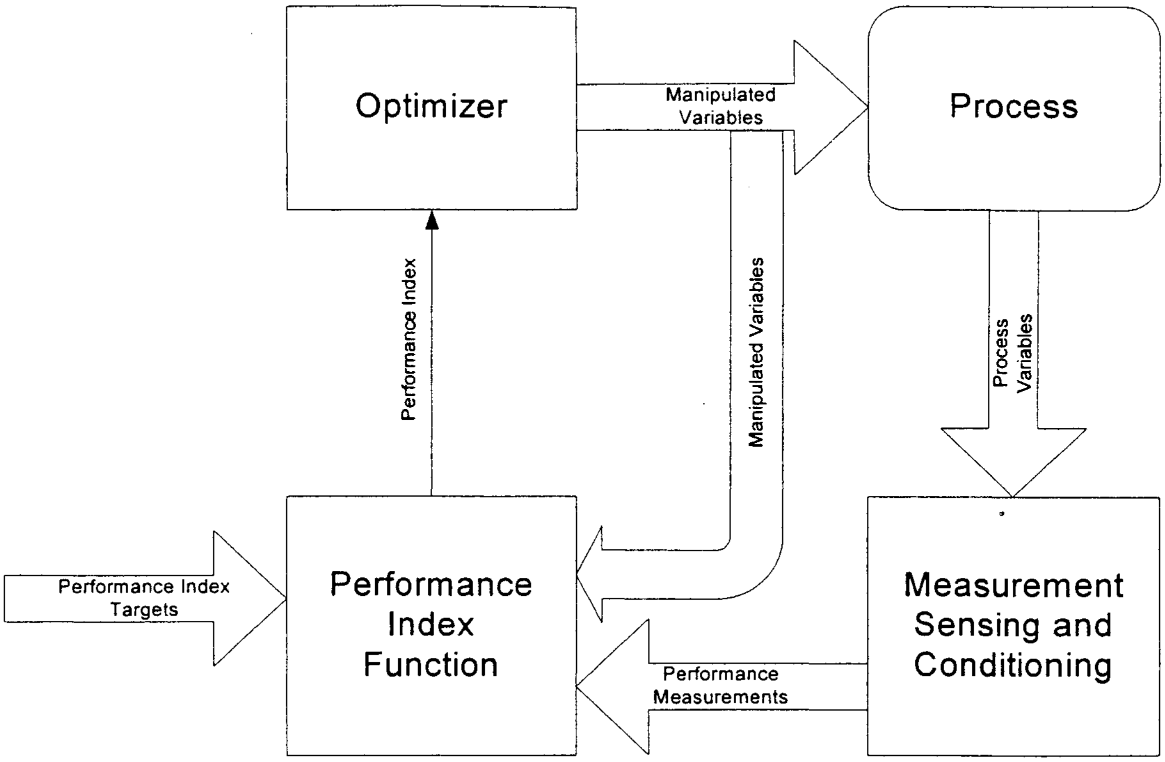


Figure 1.2 Adaptive Optimizer Targeted Environment

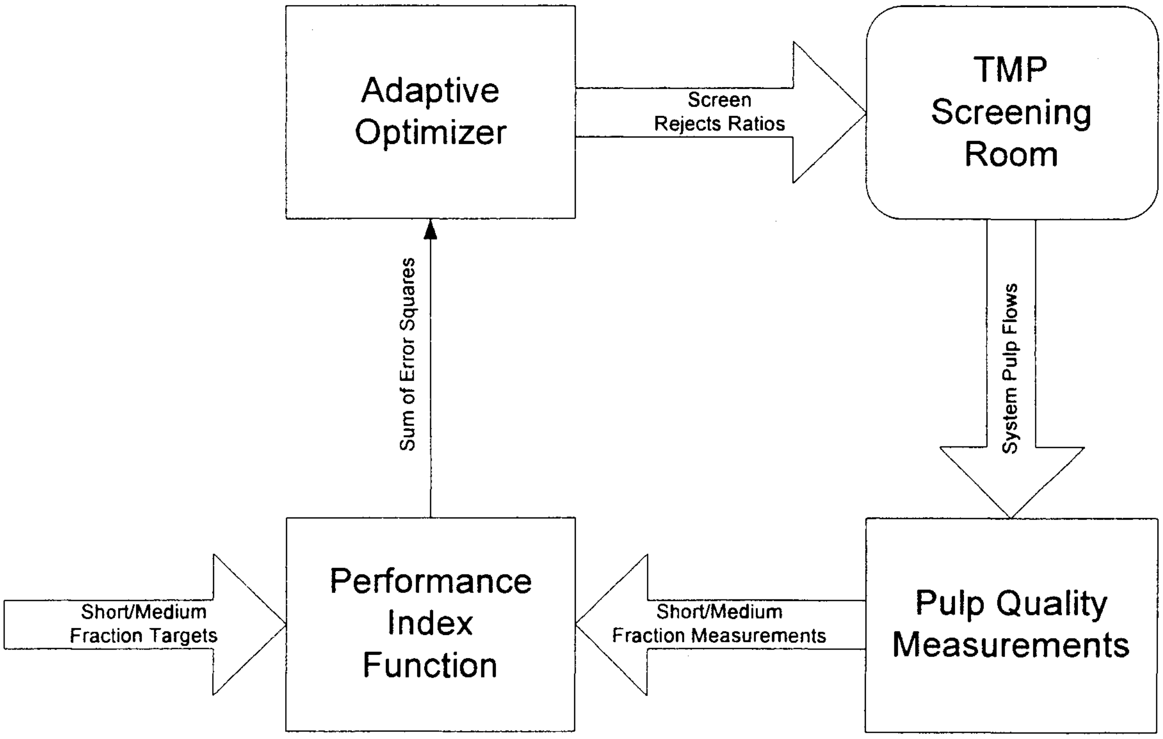

Adaptive Optimizer
Screen Rejects Ratios
TMP Screening Room
Sum of Error Squares
System Pulp Flows
Short/Medium Fraction Targets
Performance Index Function
Short/Medium Fraction Measurements
Pulp Quality Measurements

**Figure 1.3 Reflection of One Vertex of a Simplex**

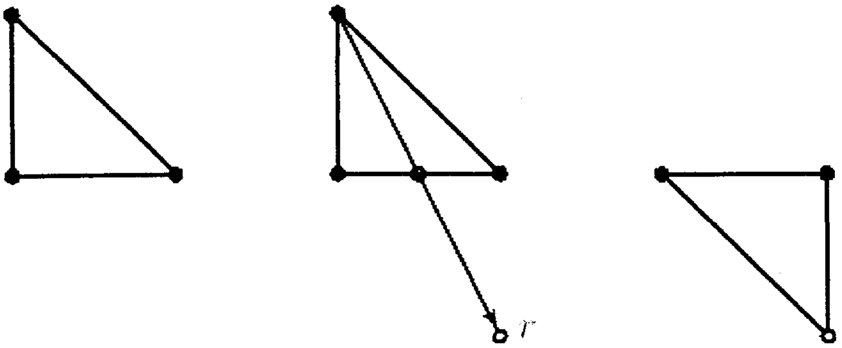


**Figure 1.4 A Sequence of Reflections Failing to Replace the Best Vertex**

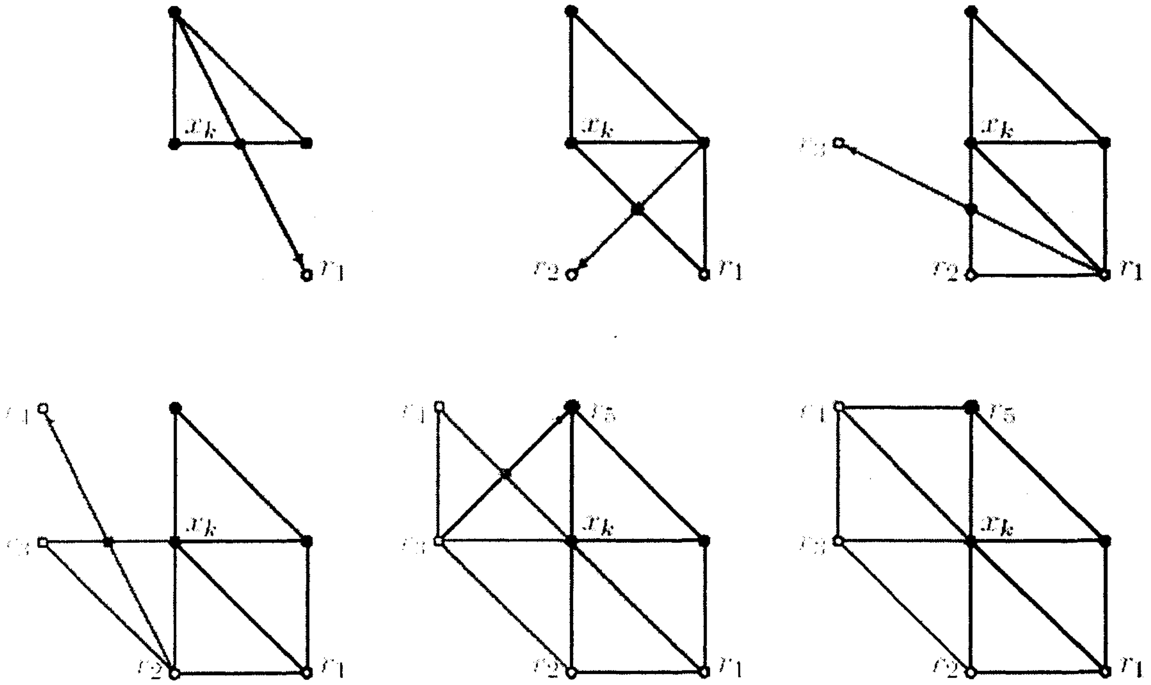


**Figure 1.5 Nelder-Mead Expansion Step**

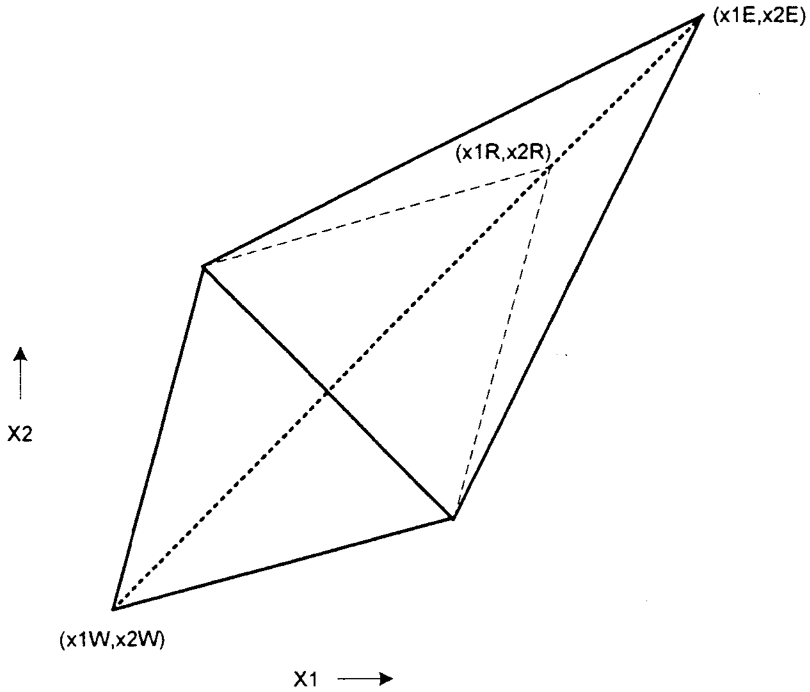
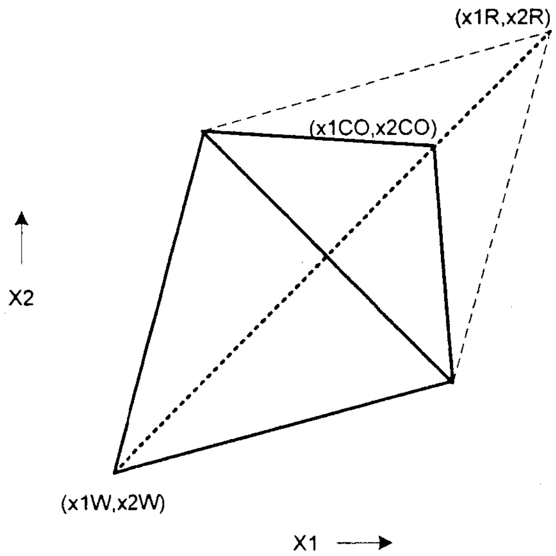
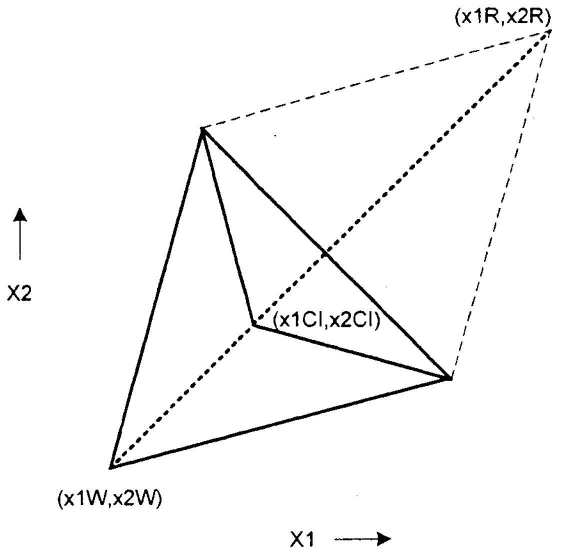


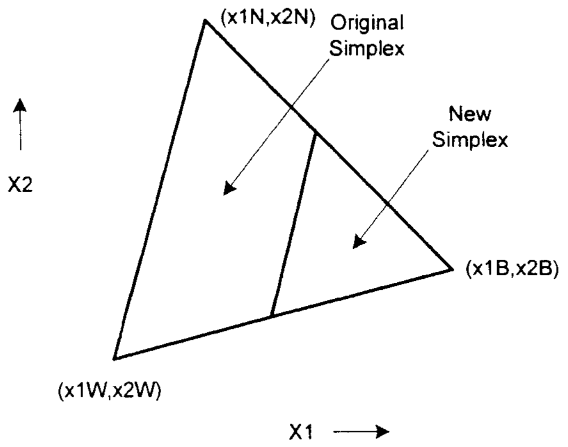
**Figure 1.6 Nelder-Mead Contraction Outside Step**



## Figure 1.7 Nelder-Mead Contraction Inside Step



## Figure 1.8 Nelder-Mead Shrink Step



# Chapter Two: Adaptive Optimizer

## 2.1 Introduction

A test environment for the adaptive optimizer is shown in Figure 2.1. The optimizer is configured here as a feedback controller in the sense that the performance, or objective, function is the sum of error squares of the process output variables from the target values. The optimizer attempts to minimize this total error square value to zero by manipulating the process input variables that determine the value of the objective function, subsequent to the application of process dynamics and the measurement sensor functions. The objective function is:

$$ObjectiveFunction = (Target_1 - Measurement_1)^2 + (Target_2 - Measurement_2)^2 . \qquad (2.1)$$

The adaptive optimizer is intended primarily for use in a steady state environment, where measurement sample times are long compared to process lags and delays. However, in order to evaluate the adaptive optimizer's potential use in an industrial plant setting, process dynamics have been added. The plant, or process, simulation provides dynamics in the form of a First Order Process with Dead Time (FOPDT) for each manipulated variable and measurement pair, where both first order lag time and dead time are adjustable parameters. Noise, in the form of a random Gaussian normal distribution, within limits, is added in certain test conditions. Additionally, process gain and absolute offset values are adjusted in other test conditions. Process input and output variables are scaled to the range of 0 to 100 per cent. The block diagram of each individual FOPDT process is shown in Figure 2.2.

The complex frequency domain transfer function of the FOPDT is standard, as follows:

$$\frac{Y(s)}{X(s)} = \frac{Ke^{-\theta s}}{\tau s + 1}, \tag{2.2}$$

where $K$ is the process gain, $\theta$ is the delay time and $\tau$ is the process lag time.

Process bias is added as an absolute value, 0 to 100 per cent. Noise is added to the output *y(t)* in the form of Gaussian random numbers with a variance of 1.0 and a mean of 0.0. The noise value is then multiplied by a variable noise gain factor, typically in the range of 1 to 10.

The manipulated variable is delayed by processing through a First In First Out (FIFO) memory buffer with variable delay $\theta$:

$$x(t) = x(t - \theta). \tag{2.4}$$

For the purpose of further testing the adaptive optimizer towards application in a typical industrial environment, the process simulation includes a pair of cross-coupled FOPDT block functions to provide process interaction. The complete process block diagram is shown in Figure 2.3.

The optimizer is intended primarily for use in situations where the optimal target function may change over time and steady state conditions are assumed for all measurements. Different from static function optimization, the adaptive optimizer provides continuous re-measurement of the simplex performance function values and it provides a contraction and expansion technique. It has been designed to provide efficient response to both step changes of considerable magnitude and to drifting or ramping changes of smaller magnitude.

Initial testing of the optimizer provides a target function that is a function of time, specifically a square wave function or a triangular wave function, and a process without dynamics. Subsequent testing uses the same varying target functions, but having one or both of the processes include FOPDT dynamics. Later testing includes interactive cross-coupled FOPDT process dynamics, and a final test relates the adaptive optimizer to standard stability criteria.

### 2.2 Adaptive Optimizer Method

The adaptive optimizer uses a direct search method, based on sequential simplex moves, in attempting to minimize the performance function to zero. The adaptive optimizer uses an equilateral triangle $(R^2 \to R)$simplex and reflects the *worst* and *next worst* points in attempting to improve the performance value, as per the original sequential simplex method. If unable to improve the *worst* value or the *next worst* value by *worst* and *next worst* reflections, respectively, it re-measures the *best* point. An unchanged *best* point evaluation causes continuation of the reflection cycles, while a change in the *best* point value forces a re-measurement of the *next best* point and the *worst* point, which is then followed by a return to the reflection cycles. The re-measurement cycle is normally a re-measure of the *best* point, followed by re-measurements of the *next best* and the *worst* points. Re-measurement of at least the *best* point is critical to the adaptive optimization method, since the performance function value at that point may change in time. Some options were tested to change the sequence of the re-measurement operations and to add additional reflection moves. These options and their use will be discussed later.

The size of the simplex is variable, expanding or contracting in response to performance function values or their ranking. Two methods for triggering expansion and contraction are used by the optimizer. One option uses a comparison of the current value of the performance function to a threshold value, while the other option uses the current and previous moves function values ranking, similar to the Nelder-Mead technique.

The threshold comparison method requires a 'closeness to optimum' threshold value, in terms of the performance function, for contraction and expansion decisions. If the current performance function value is less than the threshold value, the optimizer contracts the simplex in anticipation of convergence. Otherwise, if the current performance function value is greater than the threshold parameter, the optimizer is free to expand. The threshold value may be difficult to specify for many performance functions. In the case of the sum of error squares from targets, where the optimum is zero, the threshold value may be more easily determined. While the adaptive optimizer only provides one threshold parameter level, which is likely suitable for most industrial control ranges, several threshold parameters and several contraction/expansion levels, giving a number simplex sizes, may be suitable for larger ranges of control. The threshold method, while requiring a parameter that qualifies the performance function value, has been shown to provide as good, and often better, control than the ranking method under all test conditions. Comparable test results, where total and average errors are calculated, are shown in later sections, for both methods of contraction and expansion.

The ranking method of contraction and expansion compares the performance function value from the newest reflection to two of the performance function values associated with the current simplex. If the newest performance function value is better (i.e., less than) the performance function value for the *best* point of the current simplex, the simplex is allowed to expand. If the newest performance function value is worse (i.e., greater than) the performance function value for the *worst* point of the current simplex, the simplex is free to contract. The ranking method has the benefit of not requiring a threshold parameter, or any sense of the 'quality' of the performance function value, and as such, it is more suitable for situations where the performance function has a non-zero or unknown optimum value. This type of performance function may be a cost type of function that is dependent on the different operating conditions of the plant.

For either of the above methods, the expansion and contraction factors are always inverses of each other, and the number of expansion or contraction steps is limited. Simplex sizes are therefore a prescribed set of values. This mechanism provides limits on the number of contraction and expansion cycles, which take time to effect, due to the re-measurement cycles. In a typical industrial process control setting, convergence to a very small error may be counter-productive if the simplex must quickly adapt to a higher-valued step change in the performance function. After an expansion or contraction, a simplex re-measure cycle is always forced. Figure 2.4 is the adaptive optimizer flowchart for the first option, i.e., with expansion and contraction based on a current performance function threshold value. Figure 2.5 is the adaptive optimizer flowchart that uses the second option, i.e., the performance values ranking method.

## 2.3 Sequential Simplex Method for the Two Dimensional Case

While the sequential simplex method is general enough for multi-variable control in higher dimensions, the adaptive optimizer is restricted to two dimensions as a conservative first step in potential application to an industrial process. Working in two dimensions has the advantage of intuitive graphs and, as will be seen, allows for less manipulation and re-measurement of the process for each translation of the simplex. Simplex geometry is shown in Figure 2.6, where *m* is the length of the side of the equilateral triangle. The distance from any vertex of the simplex to the opposite side is of length *d*. The simplex is initially oriented as shown in Figure 2.6, i.e., two sides are equally centered by angle between the X1 and X2 axes. The distances *a*, and *b*, in terms of the length of the simplex side *m*, are calculated as:

$$a = \frac{\sqrt{3}+1}{2\sqrt{2}} m \text{ and} \tag{2.5}$$

$$b = \frac{\sqrt{3}-1}{2\sqrt{2}}m. \qquad (2.6)$$

These two values are used to calculate the position of the other two vertices, given one point of the simplex. For example, if the *worst* point of the simplex is known as *(x1W,x2W)*, the *next best* point is *(x1N,x2N)* and the best point is *(x1B,x2B)*, then the *best* and *next best* point positions can be calculated from the *worst* point position as follows:

$$(x1N = x1W + Ua,\ x2N = x2W + Ub) \qquad (2.7)$$

$$(x1B = x1W + Ub,\ x2B = x2W + Ua) \qquad (2.8)$$

where U, a directional scalar factor, is either 1.0 or –1.0.

A simplex move is made by reflecting one vertex through the centroid of the opposite side, as is shown in Figure 2.6, where the *worst* point reflection is indicated. The length of the move has a vector length of two times *d*, where *d* is

$$d = \frac{\sqrt{3}m}{2}. \qquad (2.9)$$

The *worst* point reflected position is then calculated as follows:

$$(x1R_w = x1B + x1N + x1W - 2.0 \times x1W,\ x2R_w = x2B + x2N + x2W - 2.0 \times x2W), \quad (2.10)$$

where $(x1R_w, x2R_w)$ is the reflection of the *worst* point.

The *next best* point reflected position is calculated as follows:

$$(x1R_n = x1B + x1N + x1W - 2.0 \times x1N,\ x2R_n = x2B + x2N + x2W - 2.0 \times x2N). \quad (2.11)$$

Contraction of the simplex reduces the length of the side of the triangle *m* by a given contraction factor, which is specified as a parameter. The *best* point position is maintained, while the *worst* point and the *next best* point positions are recalculated from the *best* point position, using the same method as equations 2.7 and 2.8, such that the newly contracted simplex has the same orientation as the original. The simplex is thus contracted along the sides of the equilateral triangle adjacent to the *best* point position as shown in Figure 2.7. The next moves after contraction or expansion are always a simplex re-measurement cycle.

Expansion is the reverse process as contraction, which includes the use of the inverse of the contraction factor as the expansion factor. Thus, if one contraction/expansion level is provided, the simplex will only have two sizes, one larger and one smaller. Again the expansion occurs along the sides adjacent to the *best* point position.

The simplex re-measurement cycle is normally a re-measurement of the performance function values for all points in the simplex, i.e., *best, next best* and *worst*. The re-measurement cycle occurs if the simplex position cannot be improved upon by any reflections, or immediately after a contraction or expansion step. Optionally, the optimizer can be configured to only re-measure the *best* point, or to just re-measure the *best* and *next best* points. While it may appear that re-measurement of fewer points is more efficient and should be sufficient for good control, particularly since reflection cycles will follow, it will be seen that re-measurement of all simplex points provides the lowest mean errors from targets for all of the test conditions.

In attempting to improve the current simplex position through reduction of the performance function values, the adaptive optimizer follows the sequential simplex method as shown in Figure 2.8. If the current simplex is as indicated by the solid line triangle, with *best, next best* and *worst* points being (x1B,x2B), (x1N,x2N) and (x1W,x2W) respectively, the first attempted improvement is by reflection of the *worst* point to (x1RW,x2RW). If this fails to improve upon the *worst* point's performance function value, the next step is a reflection of the *next worst* point to (x1RN,x2RN). If this fails to improve the performance function value over the *next worst* points value, a simplex re-measurement cycle is forced, followed by a return to the reflection steps.

An option has been added to the adaptive optimizer to allow further reflections around the *best* point space through the extension of *worst* and *next worst* points through the sides of the simplex adjacent to these points. For example, the *worst* point can be extended to the point (x1XW,x2XW). Should the performance function fail to improve upon either of the *worst* or *next worst* points, and extension of the *next worst* point is allowed to the new point (x1XN,x2XN). A failure again to improve the performance function results in a simplex re-measurement cycle, followed by a return to the reflection cycles. This option was provided to help overcome directional inefficiencies for slow-moving or ramping performance functions or for processes with considerable lags. While this option does improve the control performance under some such limited conditions, it also degrades control performance under other testing conditions. This option is similar to a component of Jutan's method [17], where the simplex for the next iteration, or set of reflections, is chosen as the simplex (from the previous successive reflection set) whose points produce the best average performance function value.

### 2.4 Adaptive Optimizer Performance Tests

The adaptive optimizer is tested through changing the targets performance function equation over time with either a square wave function or a triangular wave function. Each process variable target is changed by this perturbation function. The optimizer attempts to minimize the performance function, and in doing so, the process variables tend to converge to the changing target conditions. Test results are evaluated by reviewing time plots of the process variables and their targets, and by noting the Average Sum of Error Squares (ASES), i.e., the performance function values, for all sampled iterations of the test. This average performance function value is calculated as:

$$ASES = \frac{\sum_{1}^{N}[(\overline{X}_1(t) - X_1(t))^2 + (\overline{X}_2(t) - X_2(t))^2]}{N}, \qquad (2.12)$$

where $N$ is the number of iterations in the sampling time domain, $\overline{X}_1$ is the process variable one target, $\overline{X}_2$ is the process variable two target, and $X_1, X_2$ are the process variables.

Performance tests are categorized into the following general cases:

1. Basic tests with no process dynamics, process interaction or noise; and with no simplex contraction or expansion.

2. Tests that compare the simplex contraction and expansion methods using a process having no dynamics, process interaction or noise.
3. Basic tests with process dynamics in the form of lags, but no process interaction or noise, and with no simplex contraction or expansion.
4. Tests that compare the simplex contraction and expansion methods using a process having dynamics in the form of lags, but with no process interaction or noise.
5. Basic tests with process noise, but no process interaction or dynamics, and with no simplex contraction or expansion.
6. Tests that compare the simplex contraction and expansion methods using a process with noise, but no process interaction or dynamics.
7. Basic tests with process interaction, but no process noise or dynamics, and with no simplex contraction or expansion.
8. Tests that compare the simplex contraction and expansion methods using a process with process interaction, but no process noise or dynamics.
9. Tests that compare the re-measurement cycle options for the adaptive optimizer.
10. Tests that compare the simplex extension options.
11. Basic tests with process delay, but no process noise or dynamics, and with no simplex contraction or expansion.
12. Basic tests indicating adaptive optimizer stability issues.
13. Basic tests indicating adaptive optimizer tuning issues.

## 2.5 Case Test Conditions and Results

In all cases, the process variables and targets have been scaled to the range of 0.0 to 100 percent. Each point on the plots represents one sample time, i.e., an iteration of the simplex controller, where one move of the controller outputs is affected and one measurement of the process variables is achieved. For example, in the application to the screen room control in chapter three, this sample time would be ten minutes, which is the sampling time of the pulp quality measurement system. The initial simplex size is specified in percentage and represents the length of the side of the equilateral triangle. The simplex size may be contracted by a factor of 0.5 and later expanded, after contraction, to the original initial size. As a measure of controller performance, the Average Sum of Error Squares (ASES) is calculated and noted. In all cases, initial conditions for the process variables are 50.0 percent. Evenly numbered cases use a square wave target function perturbation, while odd numbered cases use a triangular wave target function perturbation. Except for special test category 9, the simplex re-measurement cycle includes a re-measurement of all three simplex points. Except for special test category 10, the simplex extension option is not included.

### 2.5.1 Category 1 Tests

Category 1 tests are basic tests with no process dynamics, process interaction or noise; and with no simplex contraction or expansion.

- Figure 2.9, Case 1.0 – No process dynamics, process interaction or noise; No simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 295

- Figure 2.10, Case 1.1 – No process dynamics, process interaction or noise; No simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; ASES = 139
- Figure 2.11, Case 1.2 – No process dynamics, process interaction or noise; No simplex contraction or expansion; Initial simplex size = 5.0; Step target changes; ASES = 264
- Figure 2.12, Case 1.3 – No process dynamics, process interaction or noise; No simplex contraction or expansion; Initial simplex size = 5.0; Ramp target changes; ASES = 116

Simplex size is important for best performance. A smaller simplex size allows closer convergence to the optimum. However, under step changes, a smaller simplex size requires a longer time to converge to the new optimum, thus introducing a further lag in the system. A compromise must be achieved between an optimal simplex size for step changing and ramp changing performance functions.

### 2.5.2 Category 2 Tests

Category 2 tests compare the simplex contraction and expansion methods using a process having no dynamics, process interaction or noise.

- Figure 2.13, Case 2.0 – No process dynamics, process interaction or noise; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 230
- Figure 2.14, Case 2.1 – No process dynamics, process interaction or noise; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; ASES = 94
- Figure 2.15, Case 2.2 – No process dynamics, process interaction or noise; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 283
- Figure 2.16, Case 2.3 – No process dynamics, process interaction or noise; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; ASES = 118

The control performance of the adaptive optimizer for these tests is compared to the control performance of Category 1 tests, as measured by the ASES values. A dynamic simplex size can improve control performance over a static simplex size and can improve control under both step changing and ramp changing performance functions. The threshold method of contraction and expansion can be 'tuned', by setting an appropriate threshold value, to have better control performance than the ranking method. In the above cases, the threshold is set at a value of 50.0, through trial and error, for best performance using the threshold method of contraction and expansion. However, in the experimentation, it was noted that a threshold value that was too small, i.e., in approaching the size of the initial simplex, prevented the simplex from contracting. The ranking method of contraction and expansion is a more general device, for use with any type of performance function, but in the testing of this method the performance results were usually worse than the threshold method's results.

### 2.5.3 Category 3 Tests

Category 3 tests are basic tests with process dynamics in the form of lags, but no process interaction or noise, and with no simplex contraction or expansion.

- Figure 2.17, Case 3.0 – Process Lags = 3.0; No process interaction or noise; No simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 362
- Figure 2.18, Case 3.1 – Process Lags = 3.0; No process interaction or noise; No simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; ASES = 166
- Figure 2.19, Case 3.2 – Process Lags = 6.0, No process interaction or noise; No simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 737

Control of processes with smaller amounts of lag time can be handled by the adaptive optimizer. Once the lag time increases to be beyond the immediate history horizon of the simplex method, i.e., beyond three sample times, control performance deteriorates significantly. The sequential simplex method, and the re-measurement cycles designed into the adaptive optimizer, provide an effective delay in the system of as many as five to eight sample times, as indicated by the process variable response to step changes in the performance function.

### 2.5.4 Category 4 Tests

Category 4 tests compare the simplex contraction and expansion methods using a process having dynamics in the form of lags, but with no process interaction or noise. These tests allow for one level of contraction, using a contraction factor of 0.5 for the length of the side of the simplex. The following cases can be compared to Cases 3.0 and 3.2, respectively.

- Figure 2.20, Case 4.0 – Process Lags = 3.0; No process interaction or noise; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 260
- Figure 2.21, Case 4.2 – Process Lags = 3.0, No process interaction or noise; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 607

The threshold method of contraction and expansion is able to improve control performance in comparison to the use of a static simplex size, as in Case 3.0, for a step changing performance function, and with processes of smaller lags. The ranking method of contraction and expansion is not able to improve the control performance under the same conditions as in Case 3.2, and in fact, the control performance suffers somewhat more.

### 2.5.5 Category 5 Tests

Category 5 tests are basic tests with process noise, but no process interaction or dynamics, and with no simplex contraction or expansion. The noise levels have a standard deviation of 2.0 per cent, while the length of the simplex side is 5.0 per cent. At steady state, the process variables have a deviation of approximately 5.0 per cent for one variable and 10.0 per cent for the other variable, due to the reflection and re-measurement cycles of the adaptive optimizer. On average, the noise to signal ratio is then approximately $\frac{2.0}{0.5(5.0+10.0)} = 0.2666$, which is a fairly severe test.

- Figure 2.22, Case 5.0 – Noise = 2.0, No process interaction or process dynamics; No simplex contraction or expansion; Initial simplex size = 5.0; Step target changes; ASES = 407
- Figure 2.23, Case 5.1 – Noise = 2.0, No process interaction or process dynamics; No simplex contraction or expansion; Initial simplex size = 5.0; Ramp target changes; ASES = 123
- Figure 2.24, Case 5.2 – Noise = 4.0, No process interaction or process dynamics; No simplex contraction or expansion; Initial simplex size = 5.0; Step target changes; ASES = 524

Within a limit, system noise can be handled by the adaptive optimizer. Once the noise level reaches a size where it is able to mask the effects of simplex movements, control performance degrades significantly. With a process gain of one, the adaptive optimizer is able to work with noise levels that, on statistical average, are in the range of one-quarter to one third of the simplex size, as measured by the length of the regular simplex side.

### 2.5.6 Category 6 Tests

Category 6 tests compare against Case 5 tests by implementing the simplex contraction and expansion methods using a process with noise, but no process interaction or dynamics. The final steady state simplex size is 5.0 per cent through contraction.

- Figure 2.25, Case 6.0 – Noise = 2.0, No process interaction or process dynamics; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 280
- Figure 2.26, Case 6.2 – Noise = 2.0, No process interaction or process dynamics; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 399

Smaller amounts of system noise cause the control performance to degrade when contraction and expansion methods are used by the adaptive optimizer, as shown by a comparison of Cases 2.0 and 2.2 with Cases 6.0 and 6.2. Higher levels of noise create the same problem as noted in Category 5 tests – control degrades significantly if noise levels approach the measurement levels produced by simplex moves.

### 2.5.7 Category 7 Tests

Category 7 tests are basic tests with process interaction, but no process noise or dynamics, and with no simplex contraction or expansion.

For the following tests, a quantitative measure of control loop interaction, known as Bristol's relative gain array [20], is used to specify the level of process interaction in the manner of the cross-coupled FOPDT block diagram of Figure 2.3. The relative gain array is a matrix composed of elements defined as ratios of open-loop to closed-loop gains according to the following equation, which relates the *ith* process input variable to the *jth* process output variable:

$$\lambda_{ij} = \frac{\left(\dfrac{\partial X_i}{\partial Y_j}\right)_{Y_k=const,k\neq j}}{\left(\dfrac{\partial X_i}{\partial Y_j}\right)_{X_k=const,k\neq i}} = \frac{\left(\dfrac{\partial X_i}{\partial Y_j}\right)_{other\,loops\,open}}{\left(\dfrac{\partial X_i}{\partial Y_j}\right)_{other\,loops\,closed}}, \tag{2.13}$$

where $Y$ is the process output variable and $X$ is the process input, or manipulated, variable. The relative gain array elements can be calculated from the open loop gain matrix of the individual FOPDT blocks in the interactive process as follows:

$$\lambda_{ij} = K_{ij}K^{-1}{}_{ji}, \tag{2.14}$$

where $K$ is the open loop gain matrix of the FOPDT blocks, and $K^{-1}$ is the inverse of this open loop gain matrix. For the two by two process example in this application, the first element in the relative gain array is:

$$\lambda_{11} = \frac{1}{1.0 - \dfrac{K_{12}K_{21}}{K_{11}K_{22}}}. \tag{2.15}$$

Since the rows and columns of the relative gain array sum to 1.0, the relative gain array is

$$\begin{vmatrix} \lambda_{11} & 1-\lambda_{11} \\ 1-\lambda_{11} & \lambda_{11} \end{vmatrix}. \tag{2.16}$$

For the following tests, a difficult case interaction is proposed as $\lambda_{11} = 0.5$. Letting the open loop gains $K_{11} = K_{22} = K_{21} = 1.0$, then $K_{12} = -1.0$. Also, for the following test, process dynamics are the same for all FODPT blocks, i.e., the process lag is 3.0 sample times.

The category 7 tests are as follows.

- Figure 2.27, Case 7.0 – Process interaction; No noise or process dynamics; No simplex contraction or expansion; Initial simplex size = 5.0; Step target changes; ASES = 238
- Figure 2.28, Case 7.1 – Process interaction; No noise or process dynamics; No simplex contraction or expansion; Initial simplex size = 5.0; Ramp target changes; ASES = 124

The control performance of the adaptive optimizer, under severely interactive conditions for the process variables, is comparable to non-interactive conditions, as per Cases 1.2 and 1.3. These tests also show that the adaptive optimizer is able to cope with levels of bias in the process variables. The adaptive optimizer has an inherent integral mode of operation.

### 2.5.8 Category 8 Tests

Category 8 tests compare the simplex contraction and expansion methods using a process with process interaction, but no process noise or dynamics.

The following tests use the same conditions as the Category 7 tests, except that in Case 8.0, where the threshold method of contraction and expansion is used, the threshold parameter value has been raised to a value of 100.

- Figure 2.29, Case 8.0 – Process interaction; No noise or process dynamics; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 211

- Case 8.2 – Process interaction; No noise or process dynamics; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 277

Both methods of contraction and expansion provided effective control comparable to Cases 2.0 and 2.2. However, it was necessary to increase the threshold parameter in the first case, since the simplex did not contract, due to the fact that the effects of the interactive process variables increased the average performance function values. This is a case where the ranking method of contraction and expansion has the advantage of simplicity and generality.

### 2.5.9 Category 9 Tests

Category 9 tests compare the re-measurement cycle options for the adaptive optimizer. For a number of the previous tests, as shown in first column of Table 2.1, the test was repeated under two different re-measurement options for the adaptive optimizer. The normal re-measurement cycle option is to measure all three points of the simplex when reflections fail to improve the simplex position, or when a contraction or expansion takes place. A second option is to re-measure only the *best* and *next best* points, and leave the re-measurement of the third point to a reflection move. The third option is to measure only the *best* point. For all tests, the Average Sum of Error Squares is shown for each re-measurement cycle option, for comparison purposes, in Table 2.1.

In every case, it is shown that re-measurement of all three points is the best option, i.e., the one leading to the best control performance, as measured by the ASES. Likewise, it is seen from virtually every case that re-measuring only the *best* point is the worst option for best control performance. This is a finding that distinguishes the adaptive optimizer from previous work, in that Jutan [17] and Hennings [18] both required re-measurement of the *best* point only.

### 2.5.10 Category 10 Tests

Category 10 tests compare the simplex extension options.

The adaptive optimizer provides an extension option, where, if after the *worst* point and *next best* point reflections fail to improve the simplex position, the *worst* and *next best* point are extended to provide a more complete sampling of the space around the *best* point. All of the previous tests were completed with the extension option enabled and the results were compared to the original test without the extension option enabled.

Two test results are notable.

Figure 2.31, Case 10.1 – Process Lags = 3.0; No process interaction or noise; Ranking method simplex contraction or expansion; No extension option; Initial simplex size = 10.0; Ramp target changes; ASES = 276

Figure 2.32, Case 10.3 – Process Lags = 3.0; No process interaction or noise; Ranking method simplex contraction or expansion; Extension option; Initial simplex size = 10.0; Ramp target changes; ASES = 204

While in most cases, the ASES value while using the extension option was comparable to the same test not using this option, there was one test set where the extension option provided a noticeably better control performance, as measured by the ASES. For a ramping performance function and the ranking method of contraction and expansion, the use of the extension option, as in Case 10.3, provided a measured difference in control performance, as noted in Cases 10.1 and 10.3, where a process exhibiting lag is implemented. While the extension option may provide some benefit to the ranking method in helping to find direction with a slow moving target and some process lag, there is limited value in this option to the adaptive optimizer.

### 2.5.11 Category 11 Tests

Category 11 tests are basic tests with process delay, but no process noise or dynamics, and with no simplex contraction or expansion.

Many different tests were completed using process delays. Any delay more than one sample time caused problems for the adaptive optimizer. An example of case using delay is shown as Case 11.0, Figure 2.33. It should be noted that the extension option was enabled in this case and it provided a measure of control performance improvement.

- Figure 2.33, Case 11.0 – Process Lags = 3.0; No process interaction or noise, No simplex contraction or expansion; Extension option; Initial simplex size = 10.0; Step target changes; ASES = 946

The adaptive optimizer depends on a short history of measurements for correct orientation and movement. Should these measurements be continually out of date by process delay, the simplex decisions are made ineffective and the simplex wanders. The cycling period of the adaptive optimizer at steady state is five sample times, i.e., two reflections and three re-measurements. Additional delays due to the process cause the optimizer to continually work with bad measurements and the simplex sense of direction is dependent on good measurements.

### 2.5.12 Category 12 Tests

Category 12 tests are basic tests indicating adaptive optimizer stability issues. With the addition of process lag in previous tests, it has been seen that the control loop oscillation period changed substantially, which led to an investigation of stability boundaries. For these tests the natural period of the closed loop is observed from the time plots of the process variable response to a step change in the performance function targets.

Since the change in manipulated variable is fixed by the size of the simplex, which, however, may contract and expand in specific step sizes, it is difficult to compare the controller gain to a proportional type controller. However, on average, over a number of samples, the change in manipulated variable is calculated and the average change in measured variable is also noted. The average gain of the adaptive optimizer controller is calculated as follows:

$$Kc_{avg} = \frac{\sum_{1}^{N}[\Delta y_1(t) + \Delta y_2(t)]}{\sum_{1}^{N}[\Delta x_1(t) + \Delta x_2(t)]}, \tag{2.17}$$

where

$$\Delta y_1(t) = y_1(t) - y_2(t-1) \text{ and } \Delta x_1(t) = x_1(t) - x_1(t-1) \tag{2.18}$$

are the sampled time differences for the process output and process input variables, respectively, for the $N$ samples of the test.

By the Bode stability criteria, a closed loop linear system is stable when its amplitude ratio is less than 1.0 at its critical frequency, which is the frequency at which the feedback signal lags the input signal by 180 degrees, and is unstable if its amplitude ratio is greater than 1.0 at its critical frequency. The total lagging phase angle in degrees is calculated from

$$\phi = \tan^{-1}(-\omega\tau) - \theta\omega\left(\frac{360}{2\pi}\right), \qquad (2.19)$$

where $\tau$ is the process lag time, $\theta$ is the system delay time and $\omega$ is the closed loop frequency in radians per second. The amplitude ratio is the open loop ratio of the output signal over the input signal,

$$\frac{K_c K_p}{\sqrt{\tau^2\omega^2 + 1}}, \qquad (2.20)$$

where $K_p$ is the process gain and $K_c$ is the controller gain.

From the above equations, the expected controller gain $K_c$ can be calculated when the system exhibits oscillation, where the process gain $K_p = 1$. The expected controller gain $K_c$ can be compared to the numerically calculated gain of the adaptive optimizer for the purpose of determining if the adaptive optimizer is violating the Bode stability criteria.

The following are the Category 12 tests.

- Figure 2.34, Case 12.0 – Process Lags = 3.0; No process interaction or noise, No simplex contraction or expansion; Initial simplex size = 15.0; Step target changes; ASES = 307
- Figure 2.35, Case 12.2 – Process Lags = 5.0; No process interaction or noise, No simplex contraction or expansion; Initial simplex size = 15.0; Step target changes; ASES = 756

For the following calculations, it has been assumed that the adaptive optimizer presents an inherent average delay of six sample times, as observed in most response tests, to the control loop. This delay is due to reflections, re-measurements and contractions or expansions. Using this delay, and the process lags from the above tests, and the measured oscillation frequency from the test cases 12.0 and 12.2, the lagging phase angle is calculated from equation 2.19. If this lagging phase angle is close to 180 degrees, then the system may be at its critical frequency and the controller gain value may support the observed oscillations as instability. The expected controller gain is calculated from equation 2.20.

Results are shown in the following table.

| Test Case | $\omega$, *frequency* | $\phi$, *phase angle* | $K_c$, *gain* |
|---|---|---|---|
| 12.0 | 0.3925 | -184.5 | 1.54 |
| 12.2 | 0.19625 | -112.5 | (Not Calculated) |

In Case 12.0 the expected controller gain did not compare against the measured controller gain produced by the simulation software using Equation 2.17, which returned a value of

0.30. Case 12.2 calculations did not indicate instability, even though the control loop exhibits some oscillation under those conditions. The conclusion is that the observed oscillations are a result of the properties of the simplex and not the process dynamics, and that there is not a direct method of applying the Bode stability theory with the adaptive optimizer.

### 2.5.13 Category 13 Tests

Category 13 tests are basic tests indicating adaptive optimizer tuning issues.

Tuning the adaptive optimizer consists of specifying an initial simplex size, selecting the contraction and expansion method and choosing a contraction factor. As previously stated, the selection of the contraction and expansion method will likely depend on the type of performance function being used and whether a threshold parameter is easily specified. The initial simplex size and the contraction factor choices are determined somewhat by the expected nature of the performance function perturbations and the level of noise in the system. The contracted simplex size must be able to provide performance function value changes that are greater than the mean noise level. The initial simplex size must be of sufficient size to allow fast convergence under a step change in performance function, but not too large as to create significant oscillation deviations at steady state.

The following tests all have a contraction factor of 0.25, which provides a halving of the contracted simplex size compared to the contraction factor of 0.5 used in all of the previous tests.

- Figure 2.36, Case 13.0 – No process dynamics, process interaction or noise; Threshold simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; Contraction factor = 0.25; ASES = 190 (Compared to ASES = 230, Case 2.0)
- Figure 2.37, Case 13.1 – No process dynamics, process interaction or noise; Threshold simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; Contraction factor = 0.25; ASES = 96 (Compared to ASES = 94, Case 2.1)
- Figure 2.38, Case 13.2 – No process dynamics, process interaction or noise; Ranking simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; Contraction factor = 0.25; ASES = 314 (Compared to ASES = 283, Case 2.2)
- Figure 2.39, Case 13.3 – No process dynamics, process interaction or noise; Ranking simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; Contraction factor = 0.25; ASES = 125 (Compared to ASES = 118, Case 2.3)
- Figure 2.40, Case 13.4 – Process Lags = 3.0, No Process interaction or Noise, Threshold simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; Contraction factor = 0.25; ASES = 260 (Compared to ASES = 260, Case 4.0)

The ASES for both types of performance function perturbations improved slightly when using the threshold method of contraction and expansion. The ASES for both types of performance function perturbations diminished slightly when using the ranking method of contraction and expansion. In general, tuning of the simplex size and contraction factor with the threshold method of contraction and expansion can improve the control performance by allowing closer convergence at steady state and faster convergence on a step change. The last test plot, Case 13.4, compared to Case 4.0, indicates that by decreasing the contraction factor, the control loop may become more oscillatory. Because of the simplex re-measurement cycle, any contraction or expansion step will add some delay to the system.

The following tests allow sequential contraction levels. The adaptive optimizer is able to contract more than one level. After the first contraction, a second level of contraction, using the same contraction factor, can be enabled if the contraction criteria, i.e., using either contraction method, is met. This allows better convergence at steady state but requires a penalty of additional delay in responding to further step changes and often provides poorer control performance under ramping changes with the ranking method of contraction and expansion.

- Figure 2.41, Case 13.6 – No process dynamics, process interaction or noise; Threshold simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; Contraction factor = 0.5; Two contraction levels; ASES = 267
- Figure 2.42, Case 13.7 – No process dynamics, process interaction or noise; Threshold simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; Contraction factor = 0.5; Two contraction levels; ASES = 91
- Figure 2.43, Case 13.8 – No process dynamics, process interaction or noise; Ranking simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; Contraction factor = 0.5; Two contraction levels; ASES = 353
- Figure 2.44, Case 13.9 – No process dynamics, process interaction or noise; Ranking simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; Contraction factor = 0.5; Two contraction levels; ASES = 117

## 2.6 General Conclusions

The following are general conclusions on adaptive optimizer performance based on the system testing:

- Without the use of simplex expansion and contraction, the optimizer is capable of tracking step changing or ramping target functions.
- The size of the simplex, as indicated by the length of one side of the simplex, is important in determining the average error offset from steady state conditions, and also in determining the average tracking error when ramping target functions are used.
- A larger simplex size means a larger average error under the condition where no simplex contraction is allowed.
- A simplex size that is too small causes additional system lag due to the limited size of the reflected moves, when no expansion is allowed.
- Contraction of the simplex, as provided by both expansion and contraction methods, will reduce the average steady state offset and ramping tracking error.
- The threshold and ranking methods of expansion and contraction may allow further reduction of average offset error from steady state conditions with a smaller contraction factor.
- With prudent choices of initial simplex size and performance function threshold value, the adaptive optimizer can track ramping target functions as effectively as the tracking of step changing target functions.
- Process noise, at lower levels, can be accommodated with the adaptive optimizer, given both types of system perturbations, i.e., step changes or ramping functions.

System control performance is not adversely affected, under lower levels of added process noise, when optimizer contraction levels are allowed.

- Process bias or gain levels are not a problem for the adaptive optimizer, in the sense that longer term offset error is eliminated by the inherent integral nature of the direct search method.
- Re-measurement of all simplex points, if reflection cycles fail to improve the simplex position and on a re-sizing of the simplex, is important for the best control performance. Partial re-measurement of the simplex degraded the control performance in all of the test conditions.
- The extended reflections surrounding the *best* point of the simplex are generally not useful in improving control performance.
- Process delay cannot be counteracted with the adaptive optimizer. Any delay of more than one time unit for either process variable causes continuous directional problems for the feedback control.
- There are costs in using the adaptive optimizer in the absence of a process model. Besides the problem of not being able to compensate for system dead-time, the adaptive optimizer must continually probe the process, and thus the process outputs will always be in active flux.

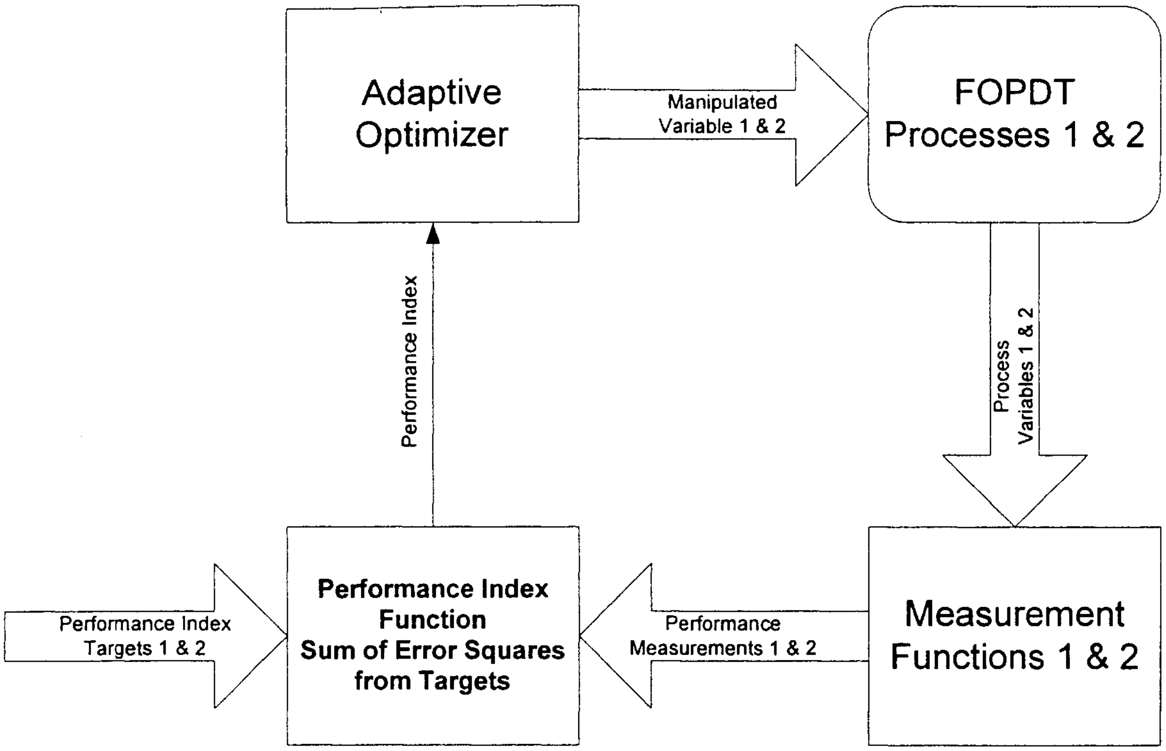

Adaptive Optimizer
Manipulated Variable 1 & 2
FOPDT Processes 1 & 2
Performance Index
Process Variables 1 & 2
Performance Index Targets 1 & 2
Performance Index Function Sum of Error Squares from Targets
Performance Measurements 1 & 2
Measurement Functions 1 & 2

**Figure 2.1 Adaptive Optimizer Tests Environment**

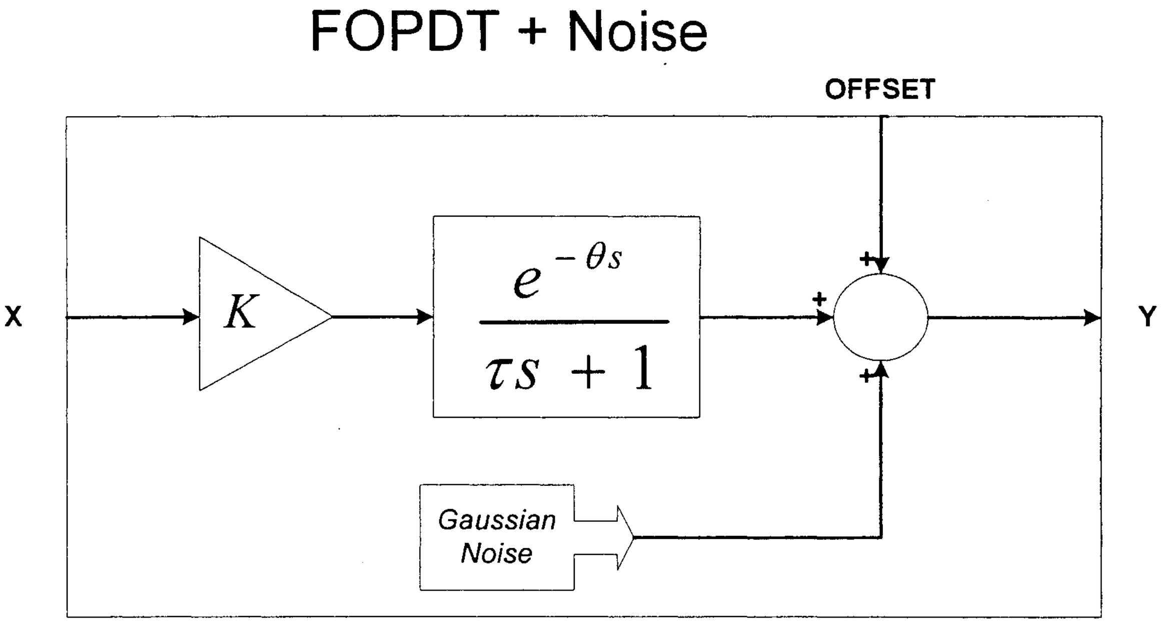


**Figure 2.2 First Order Lag Process Block with Bias and Noise Added**

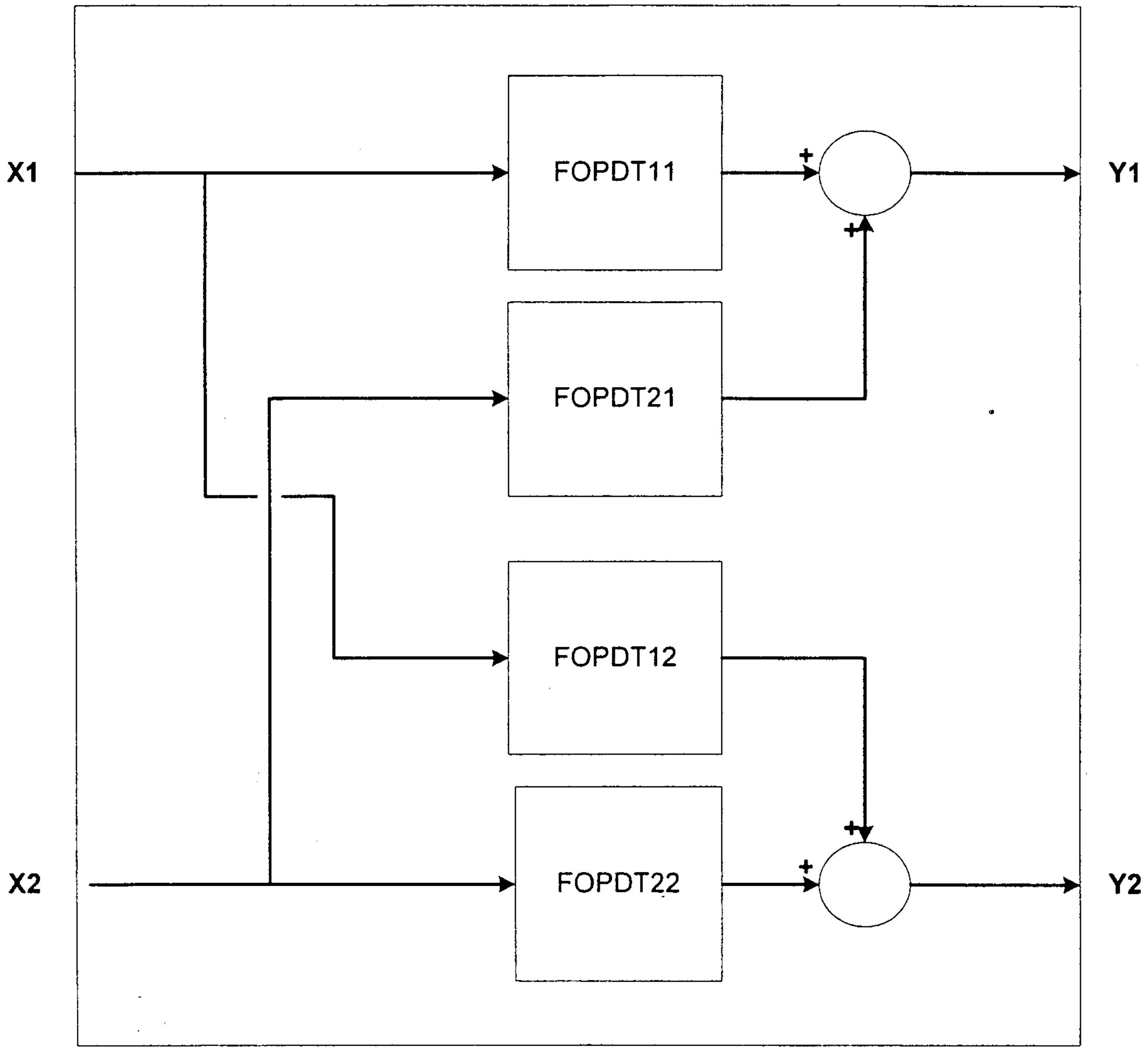


**Figure 2.3 Complete Process Block – Adaptive Optimizer Test Environment**

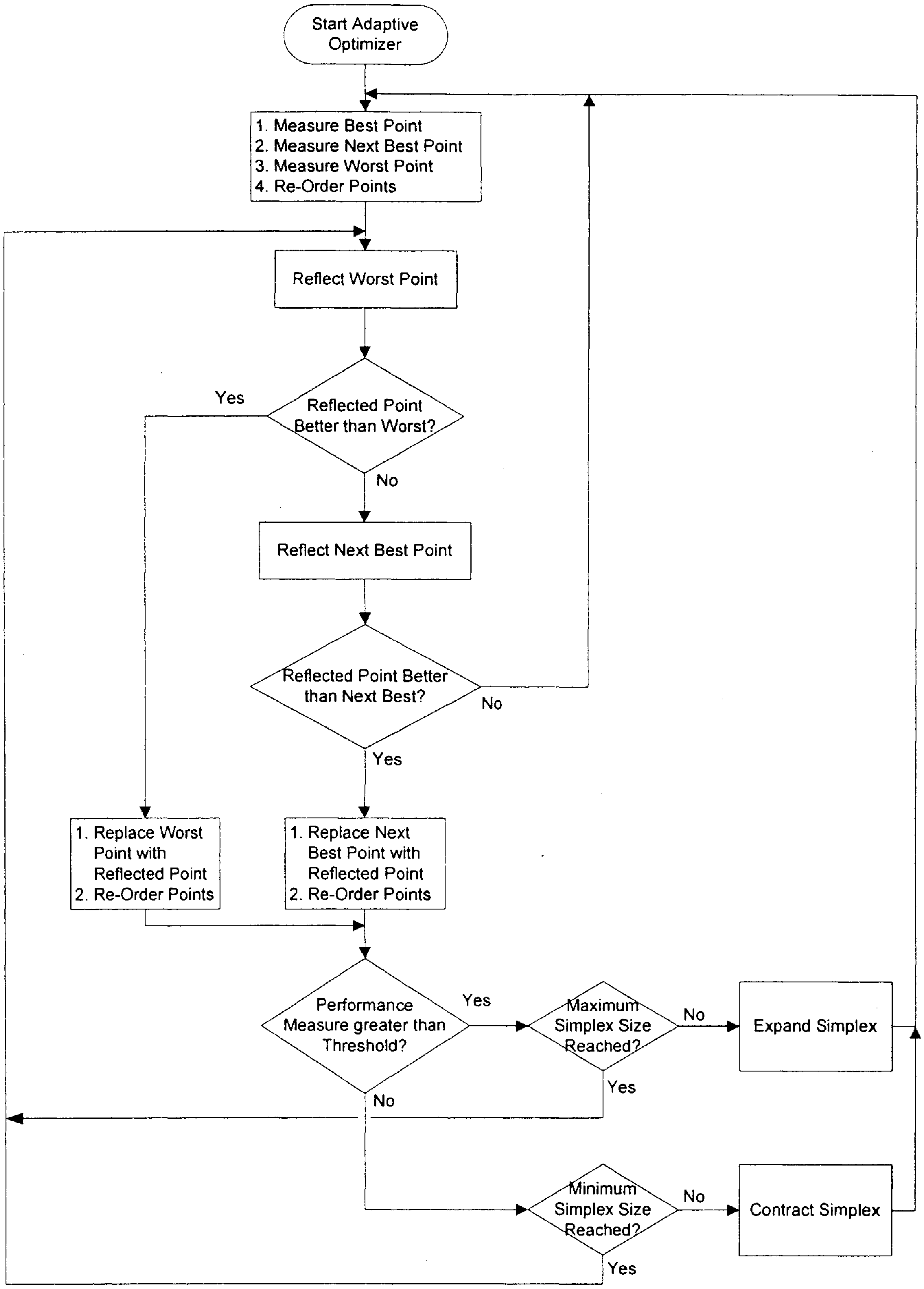

Start Adaptive Optimizer
1. Measure Best Point
2. Measure Next Best Point
3. Measure Worst Point
4. Re-Order Points
Reflect Worst Point
Reflected Point Better than Worst?
Yes
No
Reflect Next Best Point
Reflected Point Better than Next Best?
No
Yes
1. Replace Worst Point with Reflected Point
2. Re-Order Points
1. Replace Next Best Point with Reflected Point
2. Re-Order Points
Performance Measure greater than Threshold?
Yes
Maximum Simplex Size Reached?
No
Expand Simplex
Yes
No
Minimum Simplex Size Reached?
No
Contract Simplex
Yes

**Figure 2.4 Adaptive Optimizer Flowchart – Performance Function Threshold Expansion and Contraction**

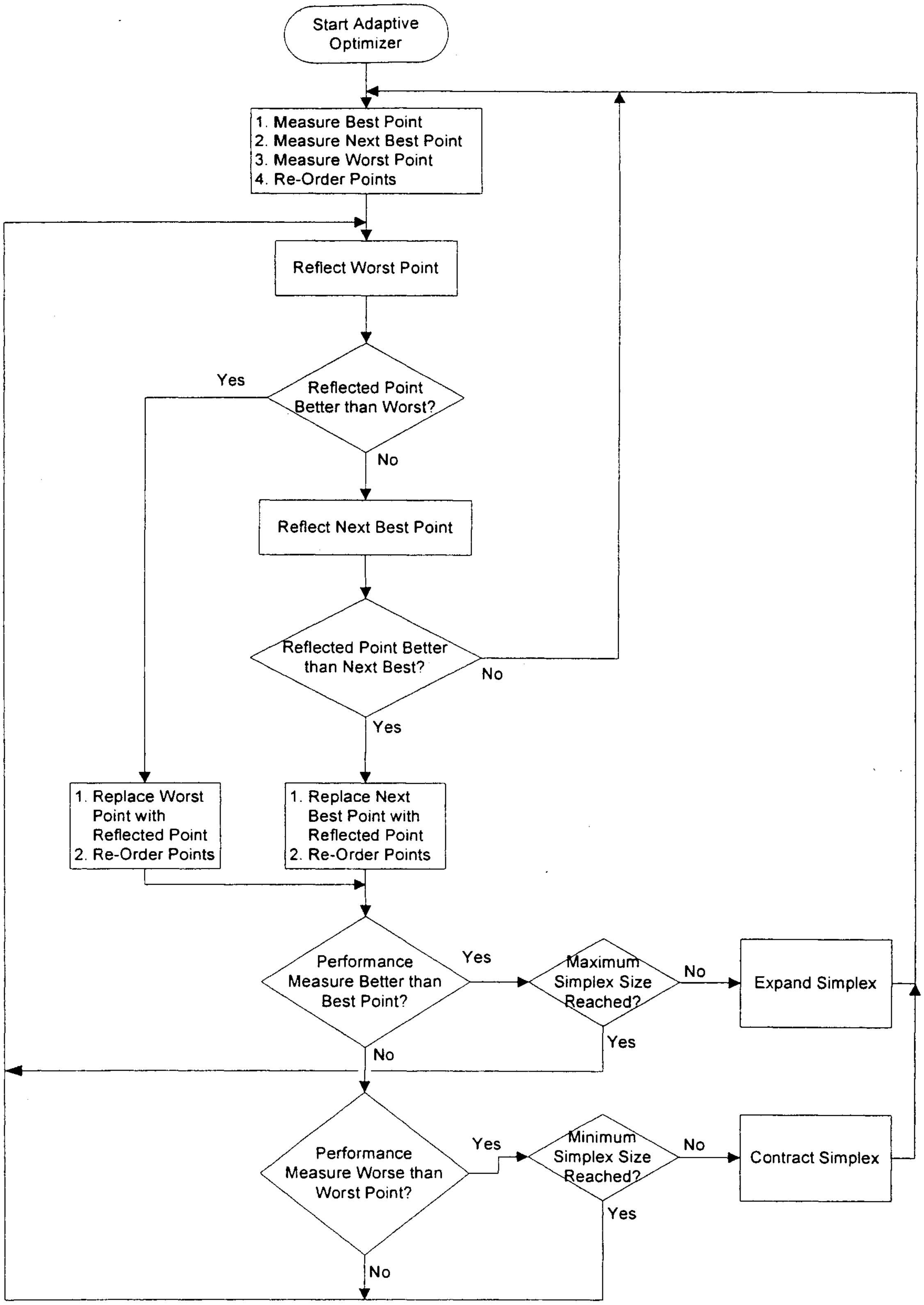

Start Adaptive Optimizer
1. Measure Best Point
2. Measure Next Best Point
3. Measure Worst Point
4. Re-Order Points
Reflect Worst Point
Reflected Point Better than Worst?
Yes
No
Reflect Next Best Point
Reflected Point Better than Next Best?
No
Yes
1. Replace Worst Point with Reflected Point
2. Re-Order Points
1. Replace Next Best Point with Reflected Point
2. Re-Order Points
Performance Measure Better than Best Point?
Yes
Maximum Simplex Size Reached?
No
Expand Simplex
Yes
No
Performance Measure Worse than Worst Point?
Yes
Minimum Simplex Size Reached?
No
Contract Simplex
Yes
No

**Figure 2.5 Adaptive Optimizer Flowchart – Performance Function Ranking Expansion and Contraction**

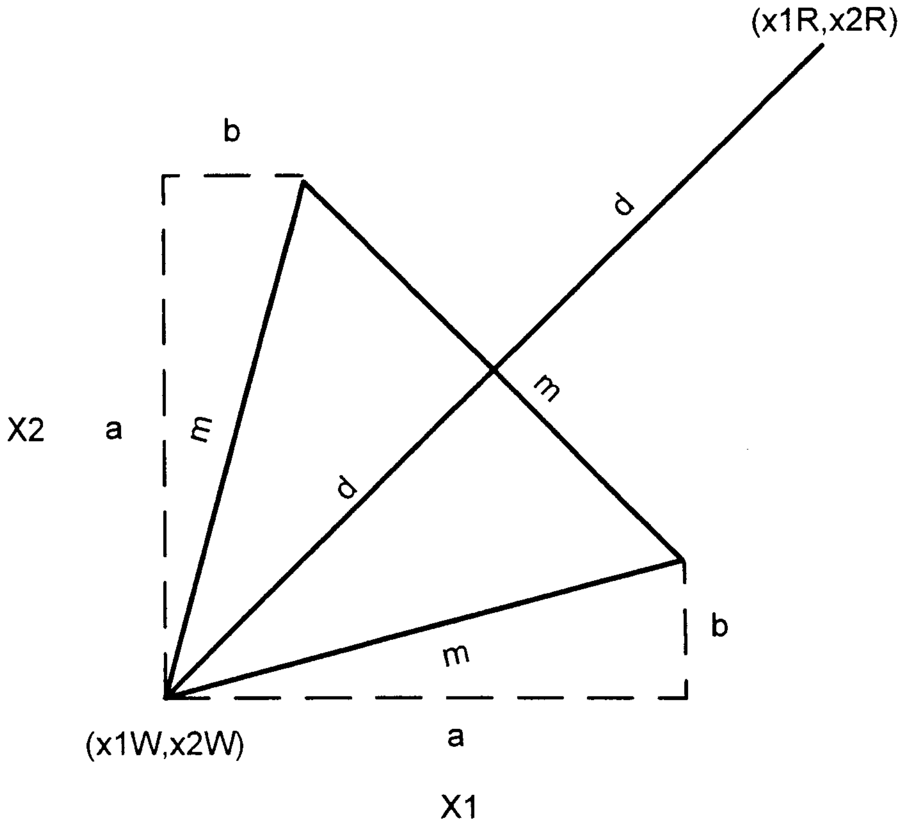


Figure 2.6 Equilateral Triangle - Simplex Geometry

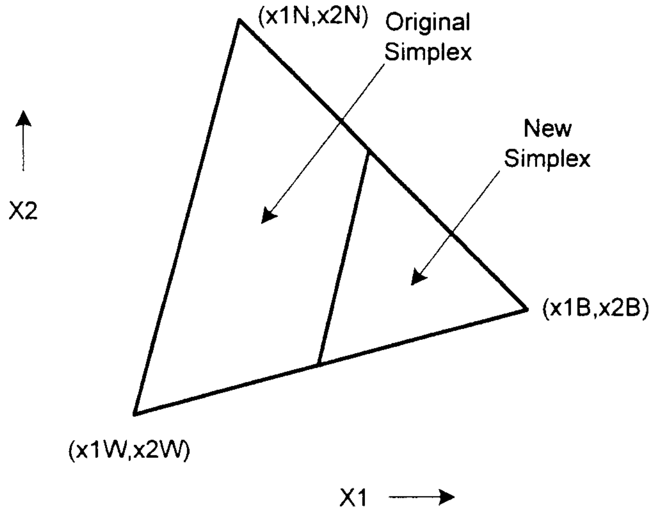


Figure 2.7 Simplex Contraction and Expansion

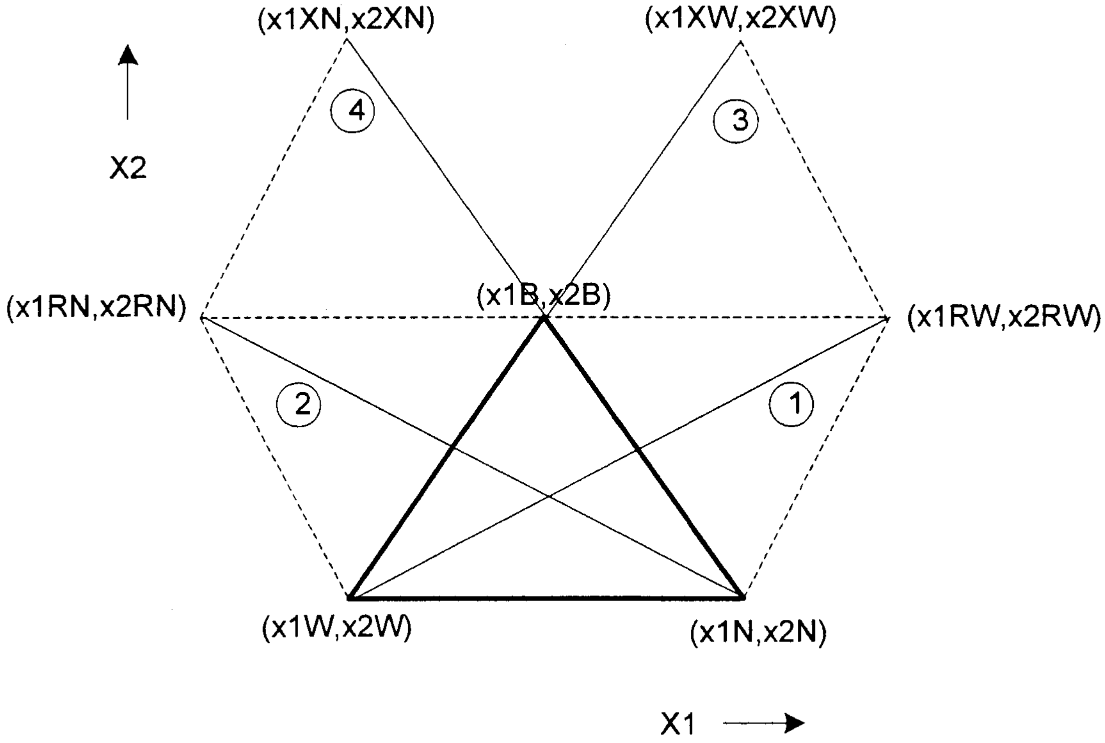


**Figure 2.8 Simplex Reflection Geometry and Extension Option**

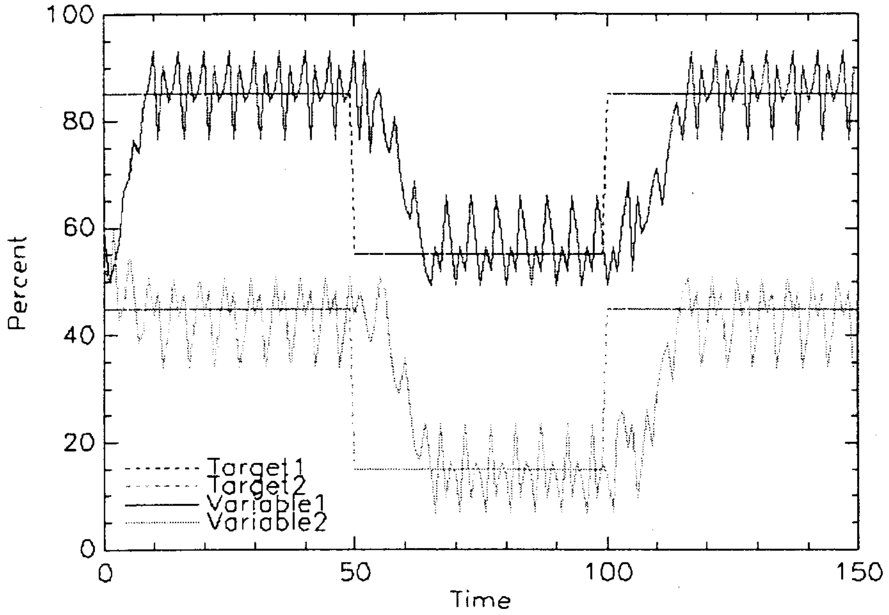


**Figure 2.9 Case 1.0**

No process dynamics, process interaction or noise; No simplex contraction or expansion; Initial simplex size = 10.0;

Step target changes; ASES = 295

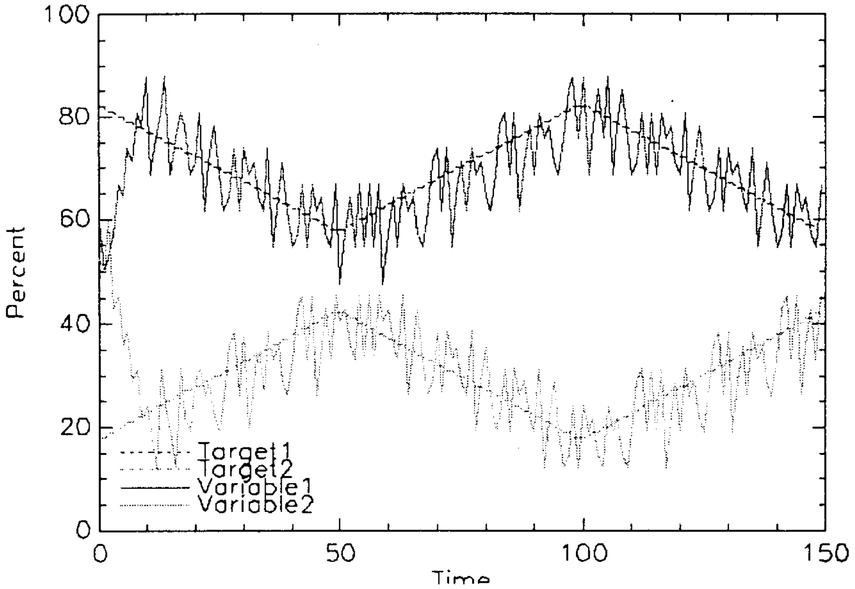


**Figure 2.10 Case 1.1**

No process dynamics, process interaction or noise; No simplex contraction or expansion; Initial simplex size = 10.0;

Ramp target changes; ASES = 139

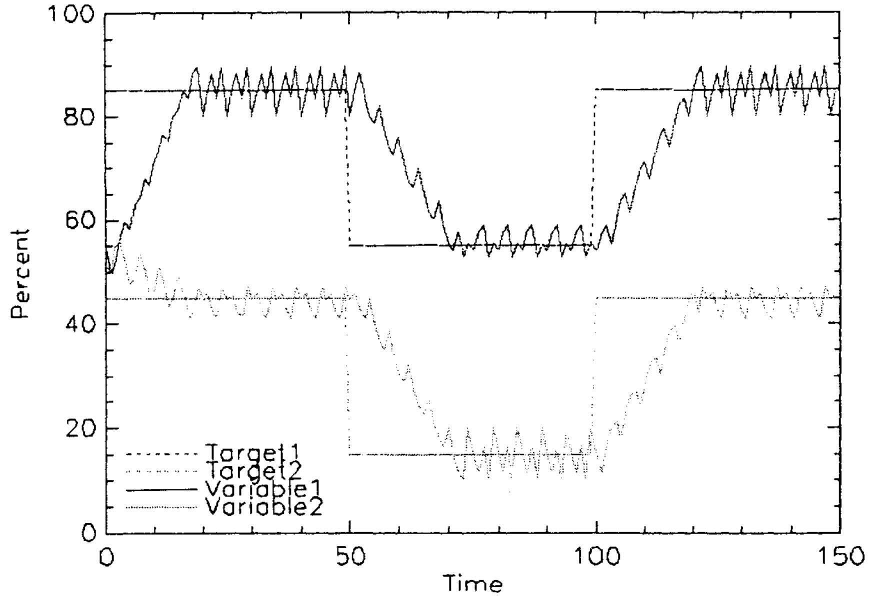


**Figure 2.11 Case 1.2**

No process dynamics, process interaction or noise; No simplex contraction or expansion; Initial simplex size = 5.0;

Step target changes; ASES = 264

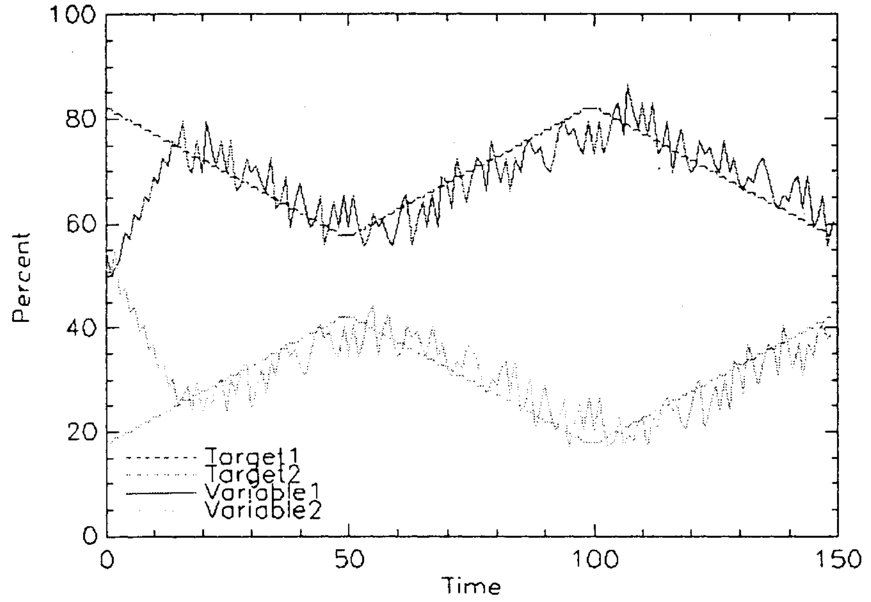


**Figure 2.12 Case 1.3**

No process dynamics, process interaction or noise; No simplex contraction or expansion; Initial simplex size = 5.0;

Ramp target changes; ASES = 116

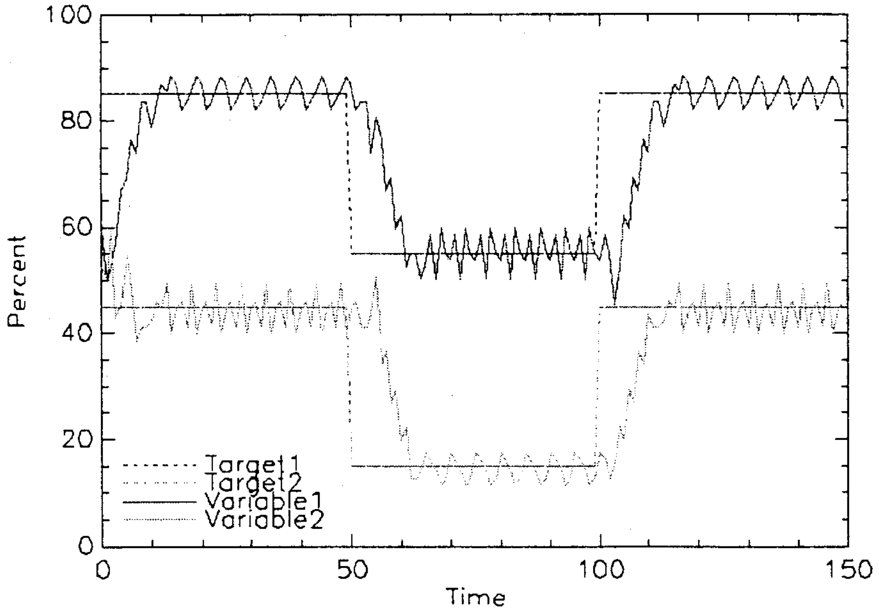


**Figure 2.13 Case 2.0**

No process dynamics, process interaction or noise; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 230

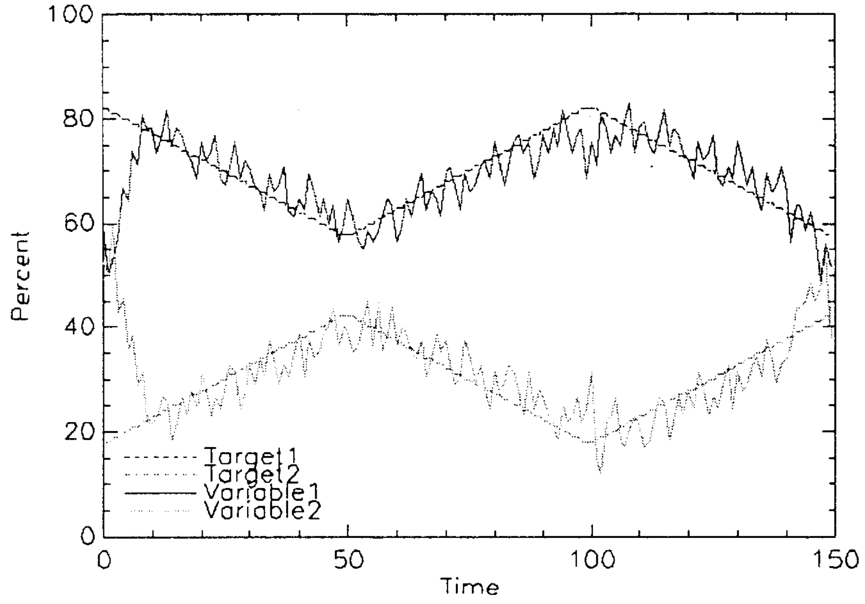


**Figure 2.14 Case 2.1**

No process dynamics, process interaction or noise; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; ASES = 94

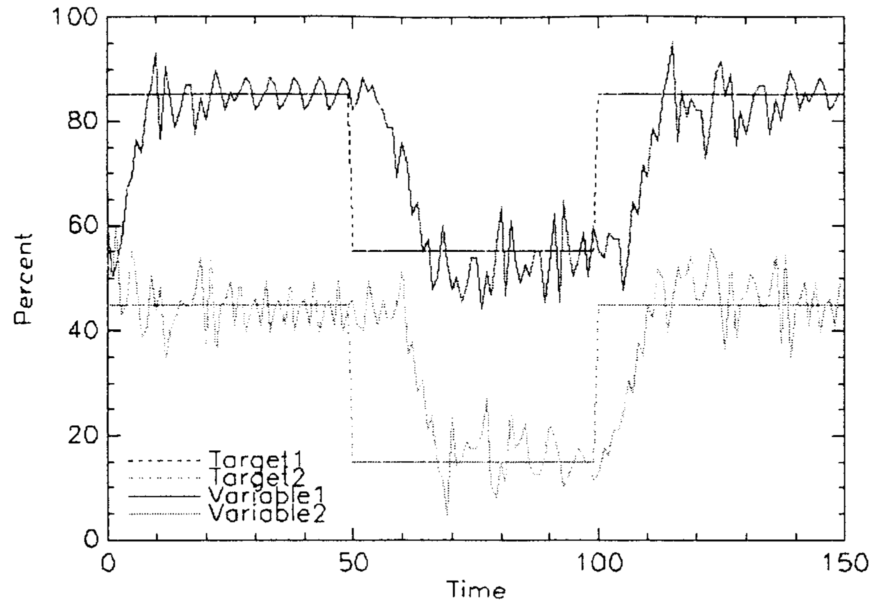


**Figure 2.15 Case 2.2**

No process dynamics, process interaction or noise; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 283

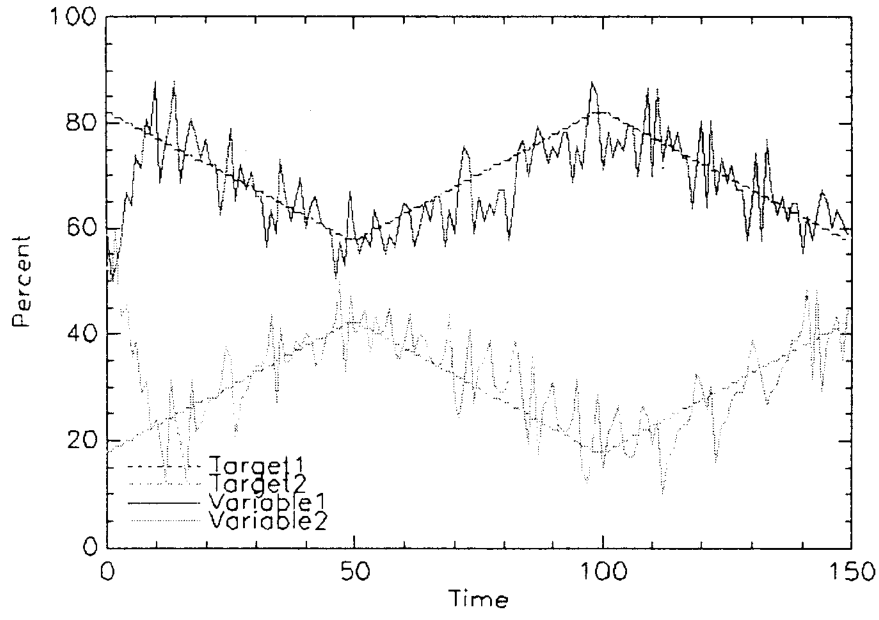


**Figure 2.16 Case 2.3**

No process dynamics, process interaction or noise; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; ASES = 118

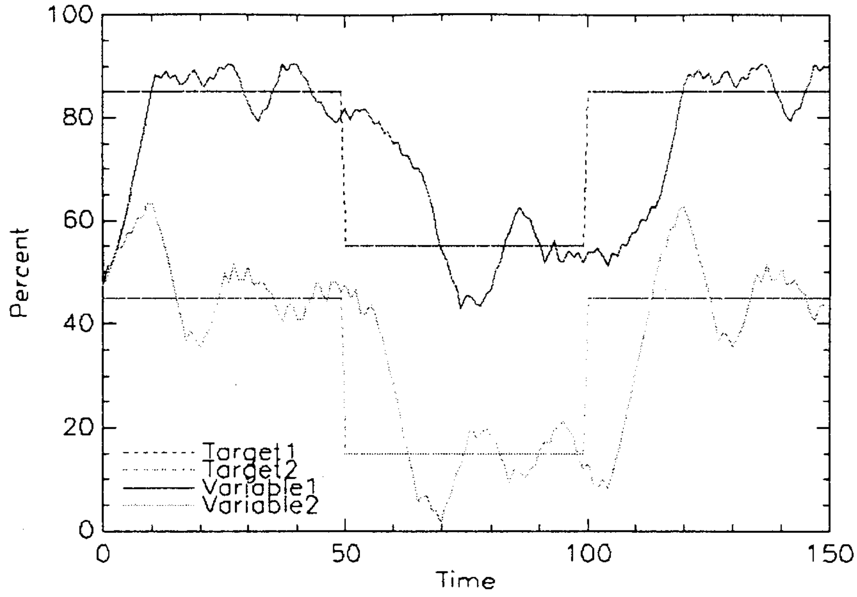


**Figure 2.17 Case 3.0**

Process Lags = 3.0; No process interaction or noise; No simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 362

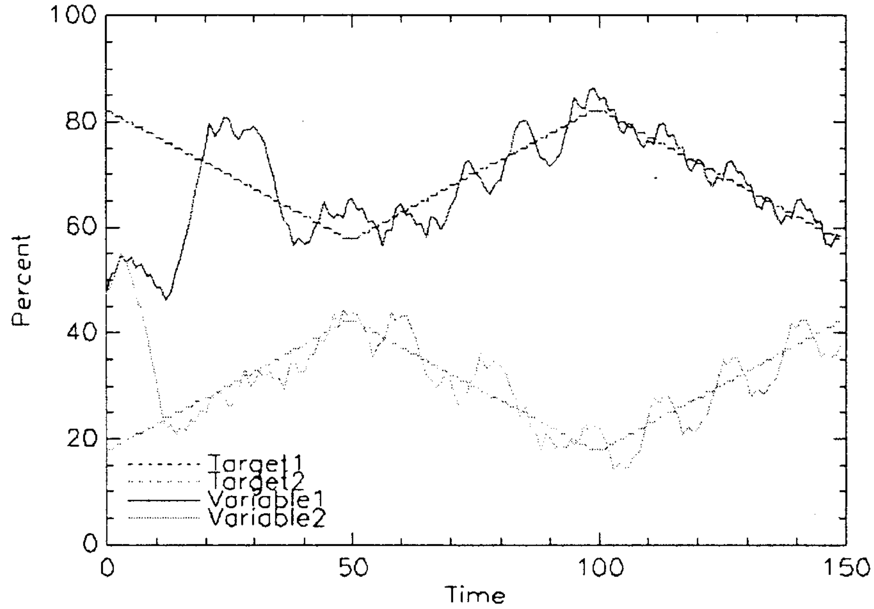


**Figure 2.18 Case 3.1**

Process Lags = 3.0; No process interaction or noise; No simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; ASES = 166

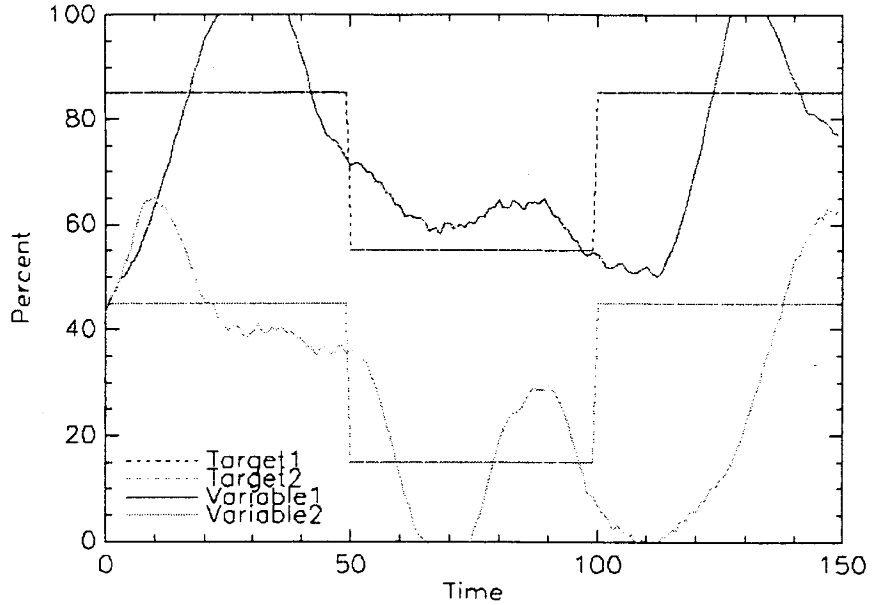


**Figure 2.19 Case 3.2**

Process Lags = 6.0, No process interaction or noise; No simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 737

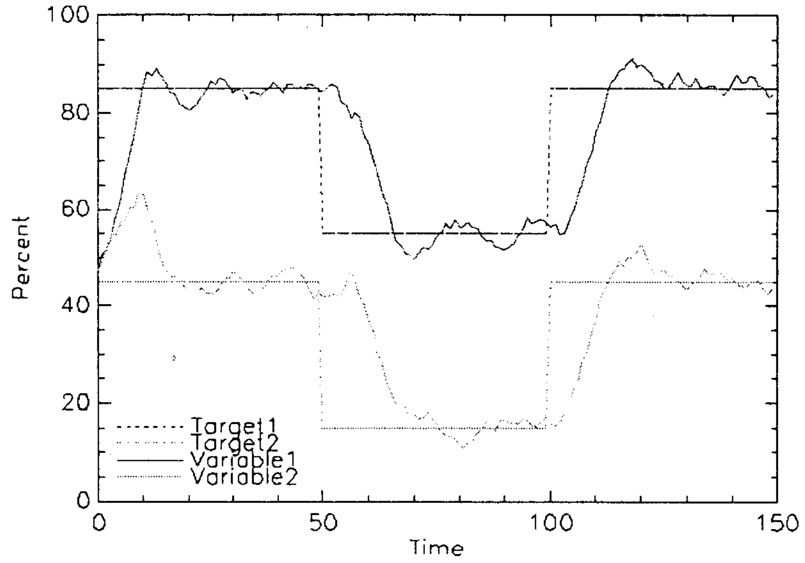


**Figure 2.20 Case 4.0**

Process Lags = 3.0; No process interaction or noise; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 260

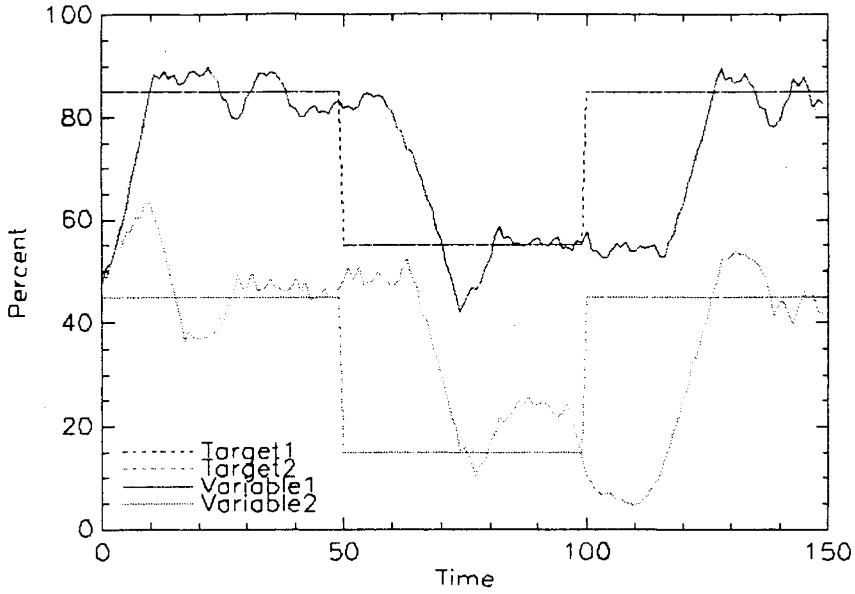


**Figure 2.21 Case 4.2**

Process Lags = 3.0, No process interaction or noise; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 607

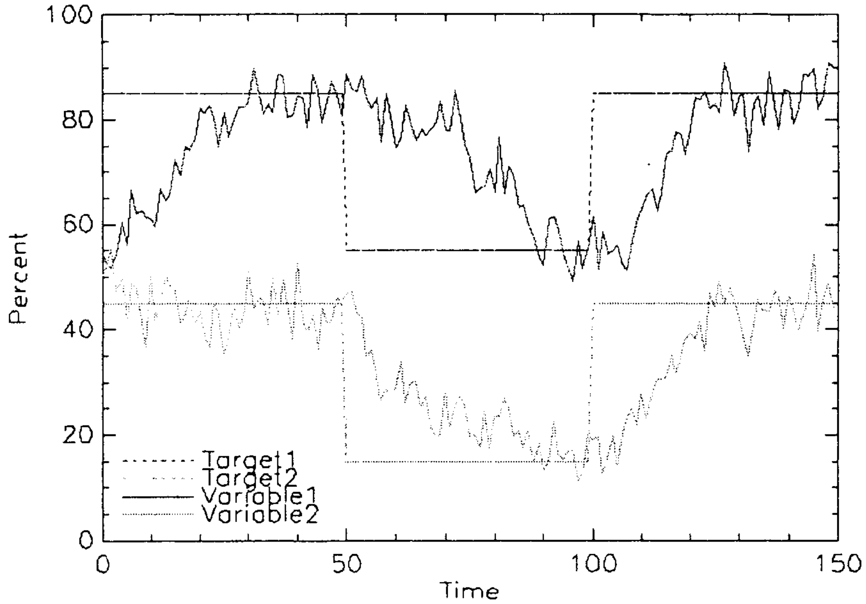


**Figure 2.22 Case 5.0**

Noise = 2.0, No process interaction or process dynamics; No simplex contraction or expansion; Initial simplex size = 5.0; Step target changes; ASES = 407

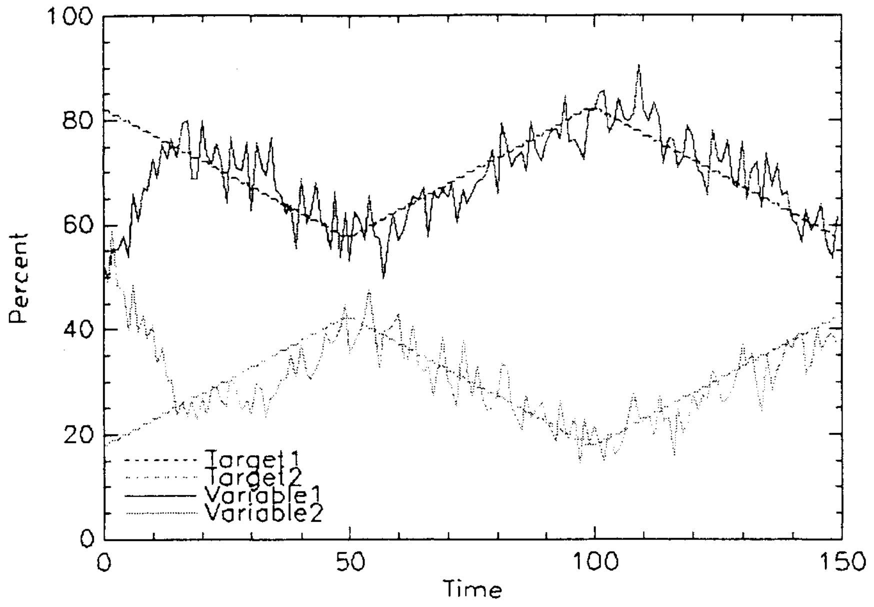


**Figure 2.23 Case 5.1**

Noise = 2.0, No process interaction or process dynamics; No simplex contraction or expansion; Initial simplex size = 5.0; Ramp target changes; ASES = 123

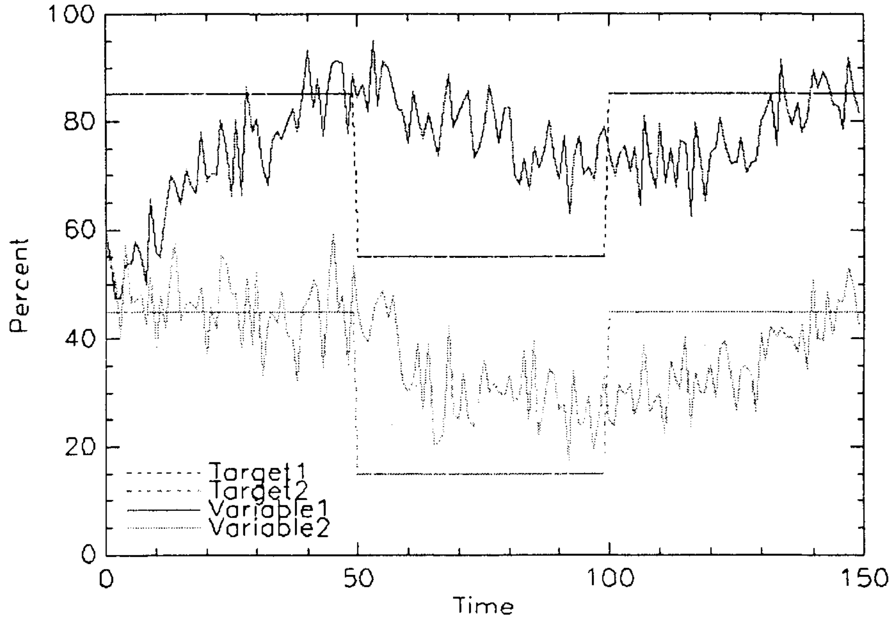


**Figure 2.24 Case 5.2**

Noise = 4.0, No process interaction or process dynamics; No simplex contraction or expansion; Initial simplex size = 5.0; Step target changes; ASES = 524

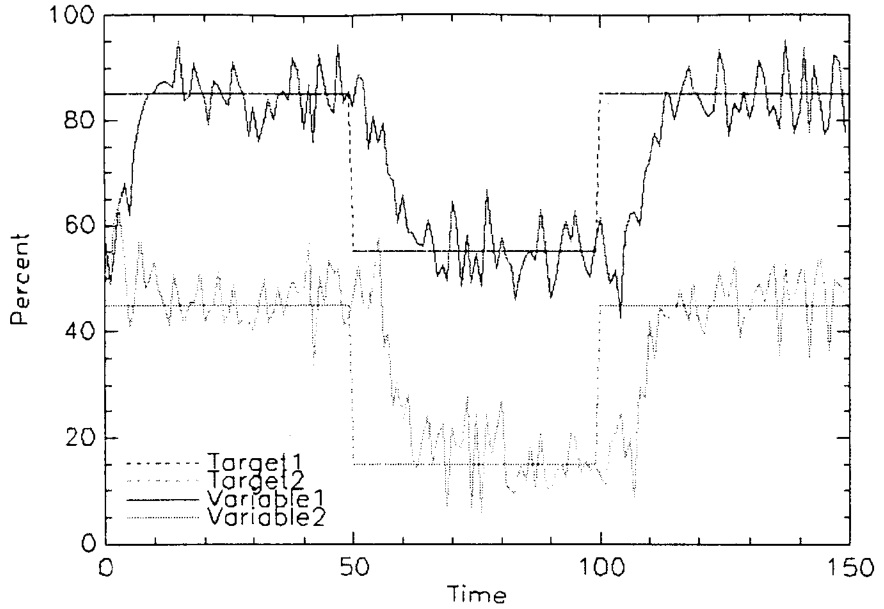


**Figure 2.25 Case 6.0**

Noise = 2.0, No process interaction or process dynamics; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 280

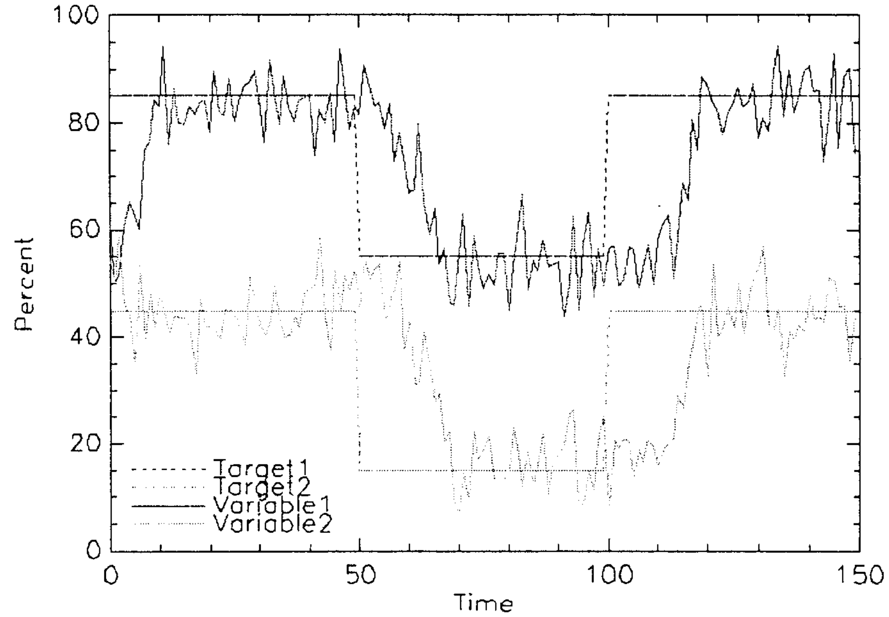


**Figure 2.26 Case 6.2**

Noise = 2.0, No process interaction or process dynamics; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 399

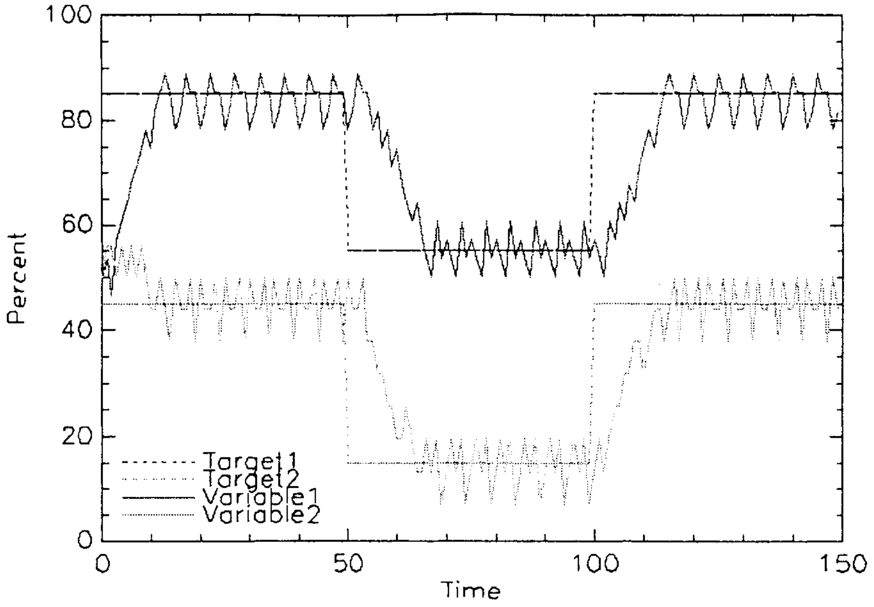


**Figure 2.27 Case 7.0**

Process interaction; No noise or process dynamics; No simplex contraction or expansion; Initial simplex size = 5.0; Step target changes; ASES = 238

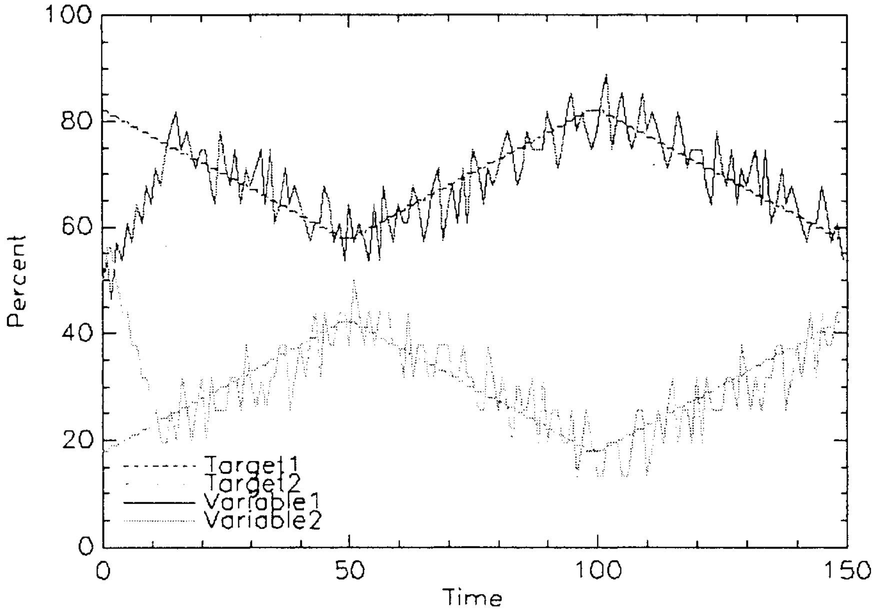


**Figure 2.28 Case 7.1**

Process interaction; No noise or process dynamics; No simplex contraction or expansion; Initial simplex size = 5.0; Ramp target changes; ASES = 124

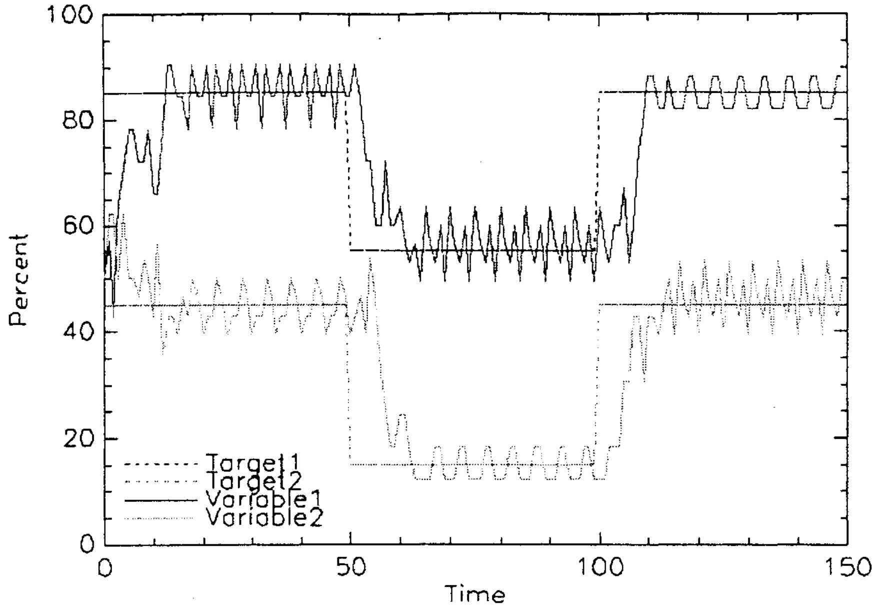


**Figure 2.29 Case 8.0**

Process interaction; No noise or process dynamics; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 211

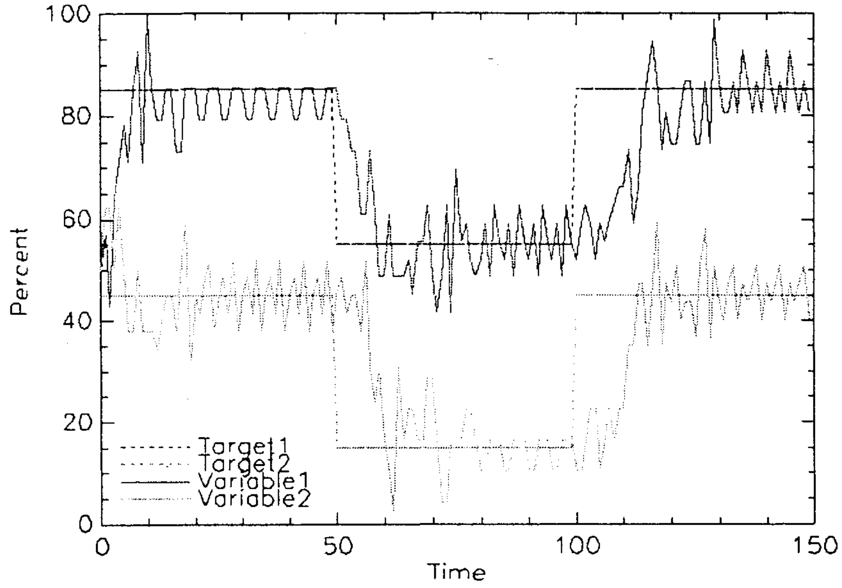


**Figure 2.30 Case 8.2**

Process interaction; No noise or process dynamics; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 278

| Re-Measure All Points<br>Original Test | Re-Measure *Best* and *Next Best* Points Only | Re-Measure *Best* Point Only |
|---|---|---|
| Figure 2.11, Case 1.2 – No process dynamics, process interaction or noise; No simplex contraction or expansion; Initial simplex size = 5.0; Step target changes; ASES = 264 | ASES = 353 | ASES = 412 |
| Figure 2.12, Case 1.3 – No process dynamics, process interaction or noise; No simplex contraction or expansion; Initial simplex size = 5.0; Ramp target changes; ASES = 116 | ASES = 249 | ASES = 265 |
| Figure 2.13, Case 2.0 – No process dynamics, process interaction or noise; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 230 | ASES = 239 | ASES = 322 |
| Figure 2.14, Case 2.1 – No process dynamics, process interaction or noise; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; ASES = 94 | ASES = 157 | ASES = 207 |
| Figure 2.15, Case 2.2 – No process dynamics, process interaction or noise; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 283 | ASES = 388 | ASES = 393 |
| Figure 2.16, Case 2.3 – No process dynamics, process interaction or noise; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; ASES = 118 | ASES = 230 | ASES = 332 |
| Figure 2.17, Case 3.0 – Process Lags = 3.0; No process interaction or noise; No simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 362 | ASES = 412 | ASES = 572 |
| Figure 2.18, Case 3.1 – Process Lags = 3.0; No process interaction or noise; No simplex contraction or expansion; Initial simplex size = | ASES = 214 | ASES = 264 |

|  |  |  |
|---|---|---|
| 10.0; Ramp target changes; ASES = 166 |  |  |
| Figure 2.20, Case 4.0 – Process Lags = 3.0; No process interaction or noise; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 260 | ASES = 503 | ASES = 499 |
| Figure 2.21, Case 4.2 – Process Lags = 3.0, No process interaction or noise; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 607 | ASES = 677 | ASES = 622 |
| Figure 2.25, Case 6.0 – Noise = 2.0, No process interaction or process dynamics; Threshold method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 280 | ASES = 281 | ASES = 349 |
| Figure 2.27, Case 7.0 – Process interaction; No noise or process dynamics; No simplex contraction or expansion; Initial simplex size = 5.0; Step target changes; ASES = 1136 | ASES = 1350 | ASES = 1505 |
| Case 8.2 – Process interaction; No noise or process dynamics; Ranking method simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; ASES = 772 | ASES = 1523 | ASES = 1422 |

**Table 2.1 Comparison of Adaptive Optimizer Re-Measurement Cycle Options**

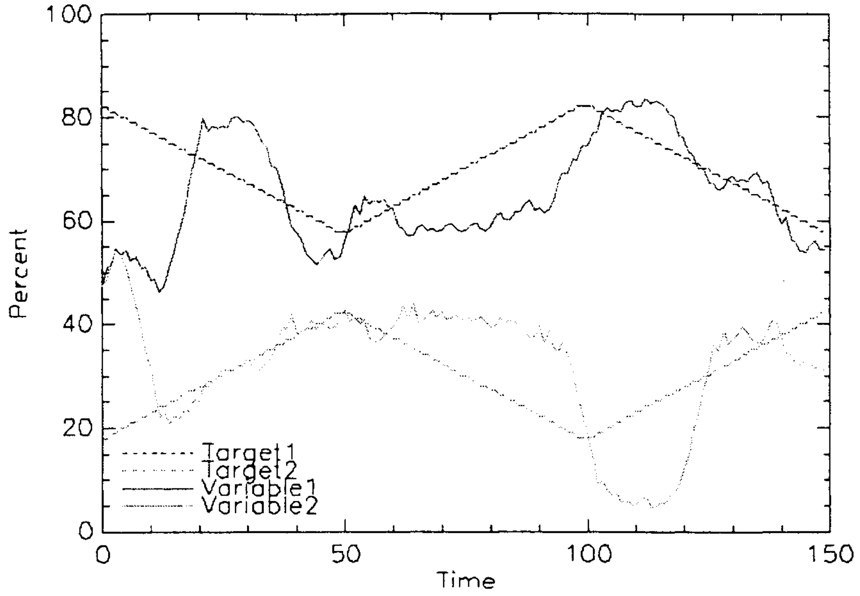


**Figure 2.31 Case 10.1**

Process Lags = 3.0; No process interaction or noise; Ranking method simplex contraction or expansion; No extension option; Initial simplex size = 10.0; Ramp target changes; ASES = 276

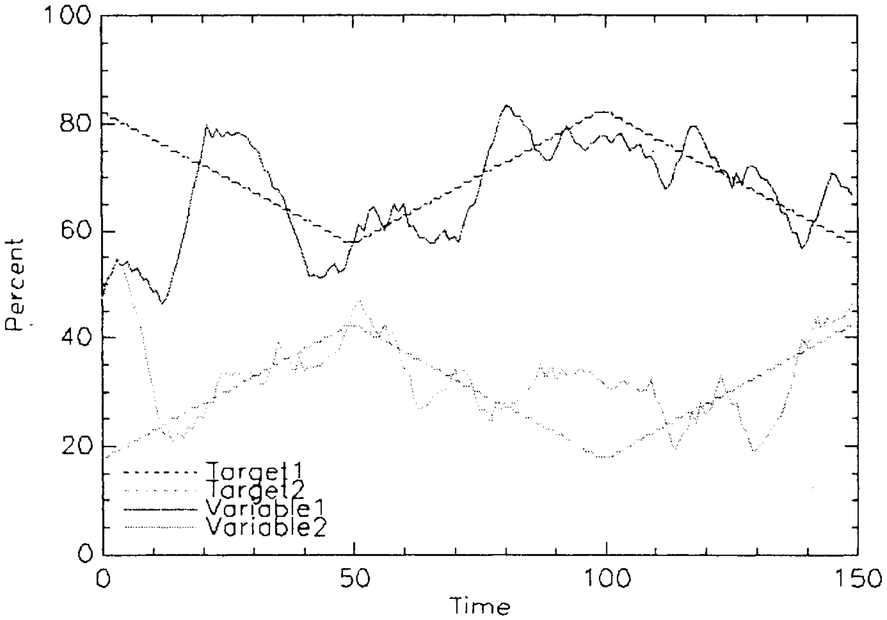


**Figure 2.32 Case 10.3**

Process Lags = 3.0; No process interaction or noise; Ranking method simplex contraction or expansion; Extension option; Initial simplex size = 10.0; Ramp target changes; ASES = 204

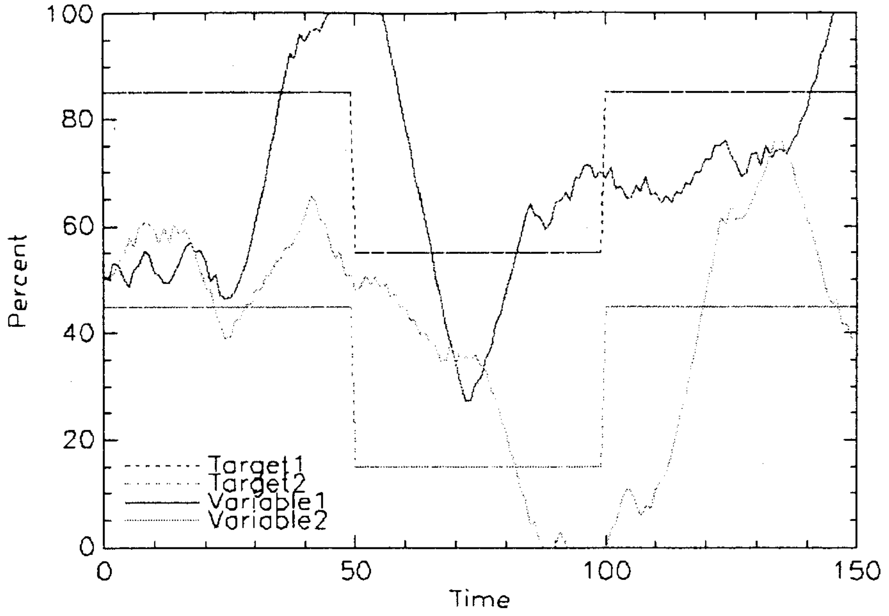


**Figure 2.33 Case 11.0**

Process Lags = 3.0; Delay = 3.0; No process interaction or noise, No simplex contraction or expansion; Extension option; Initial simplex size = 10.0; Step target changes; ASES = 946

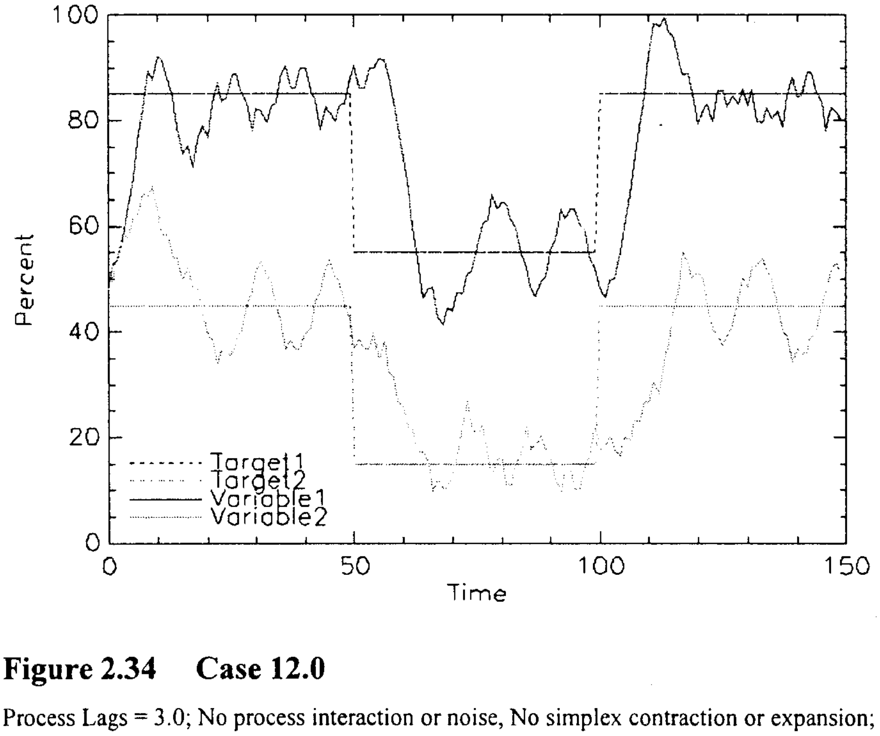


**Figure 2.34 Case 12.0**

Process Lags = 3.0; No process interaction or noise, No simplex contraction or expansion; Initial simplex size = 15.0; Step target changes; ASES = 307

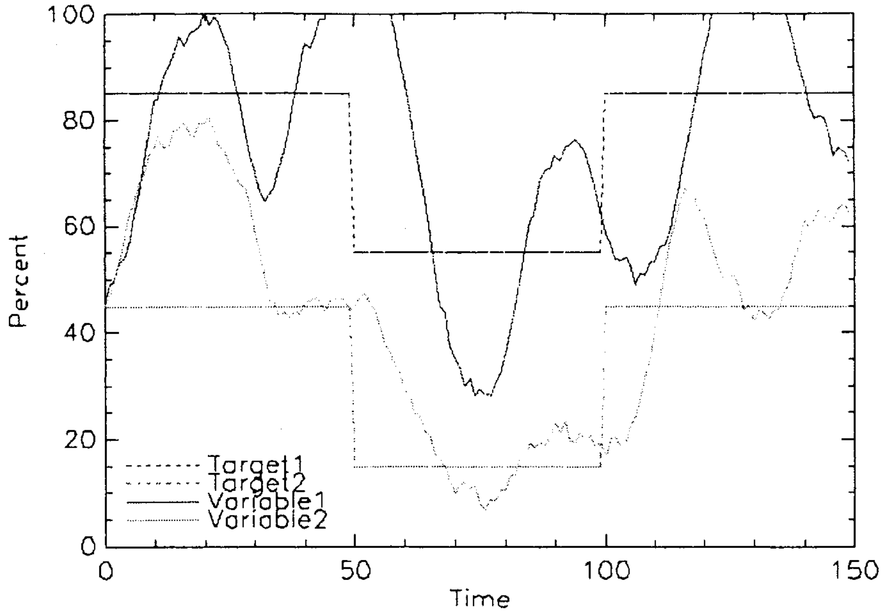


**Figure 2.35 Case 12.2**

Process Lags = 5.0; No process interaction or noise, No simplex contraction or expansion; Initial simplex size = 15.0; Step target changes; ASES = 756

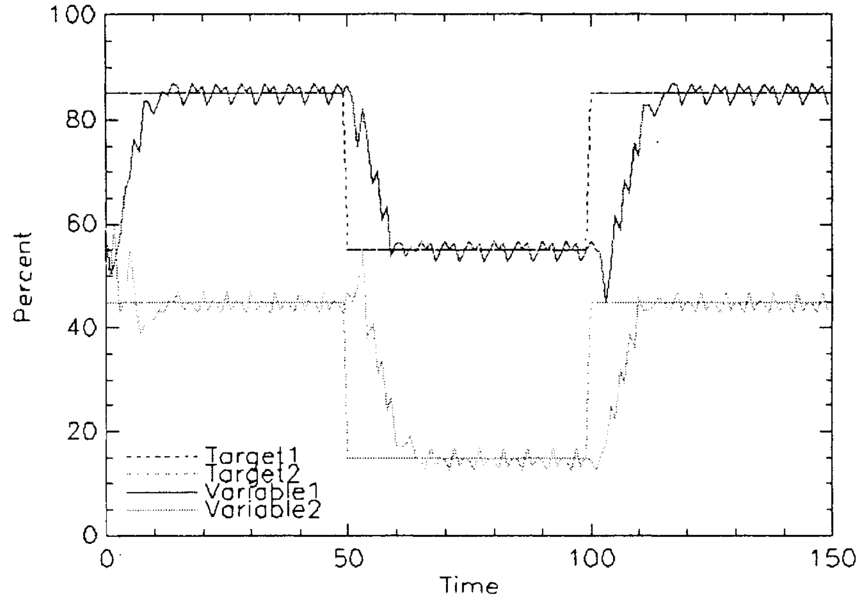


**Figure 2.36 Case 13.0**

No process dynamics, process interaction or noise; Threshold simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; Contraction factor = 0.25; ASES = 190 (Compared to ASES = 230, Case 2.0)

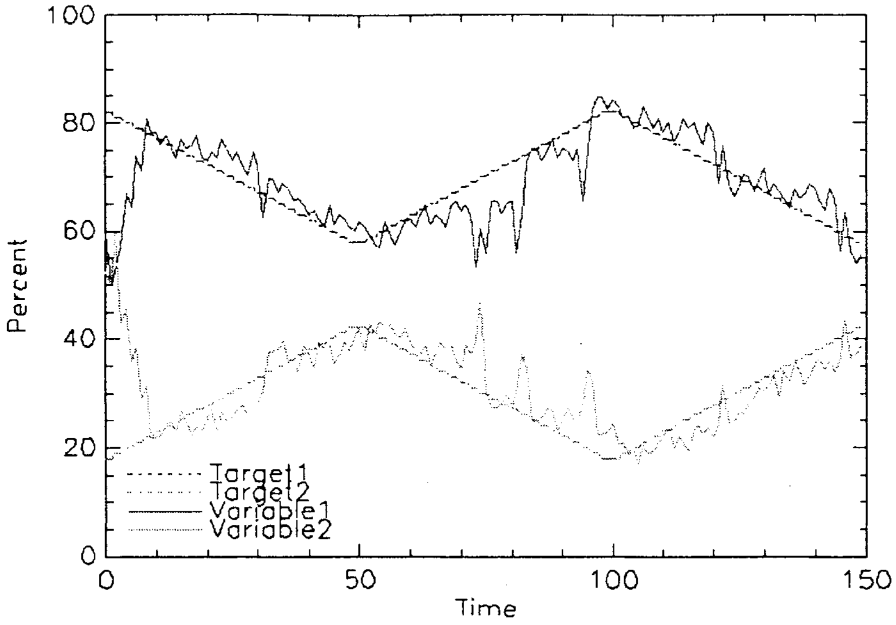


**Figure 2.37 Case 13.1**

No process dynamics, process interaction or noise; Threshold simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; Contraction factor = 0.25; ASES = 96 (Compared to ASES = 121, Case 2.1)

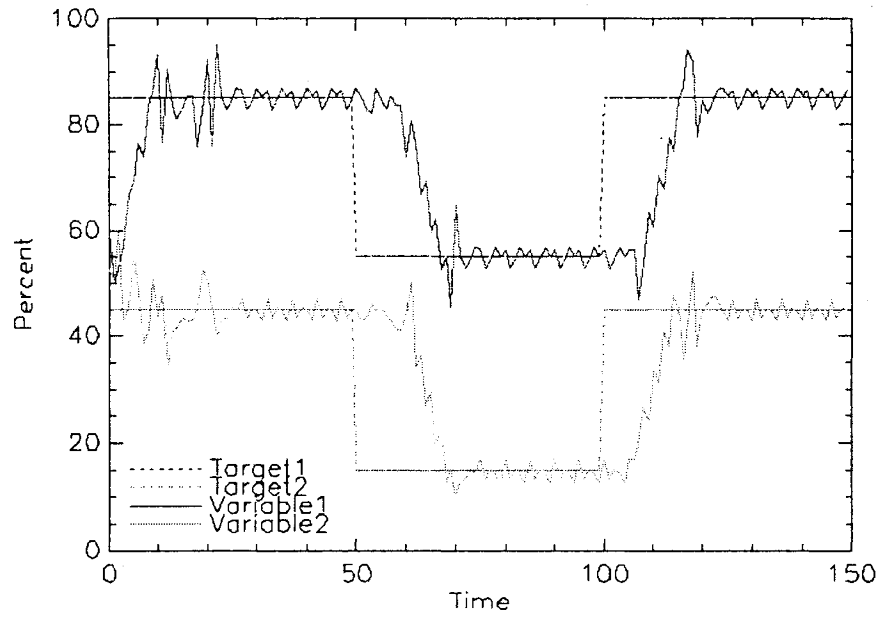


**Figure 2.38 Case 13.2**

No process dynamics, process interaction or noise; Ranking simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; Contraction factor = 0.25; ASES = 314 (Compared to ASES = 283, Case 2.1)

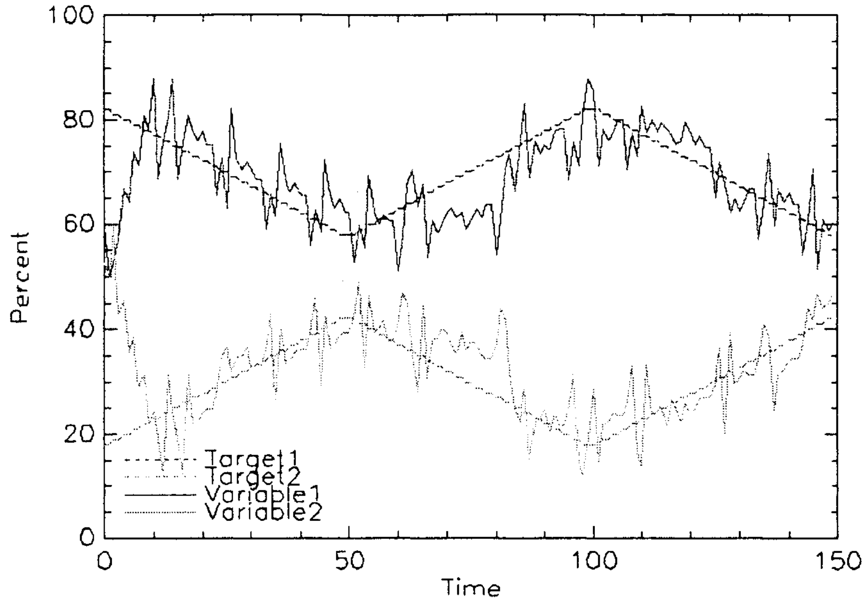


**Figure 2.39 Case 13.3**

No process dynamics, process interaction or noise; Ranking simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; Contraction factor = 0.25; ASES = 125 (Compared to ASES = 118, Case 2.2)

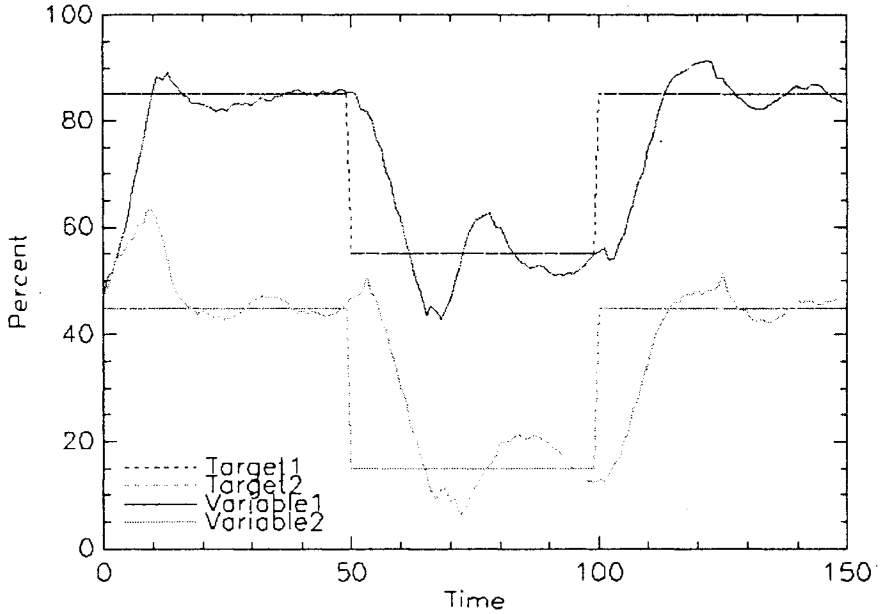


**Figure 2.40 Case 13.4**

Process Lags = 3.0, No Process interaction or Noise, Threshold simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; Contraction factor = 0.25; ASES = 260 (Compared to ASES = 260, Case 4.0)

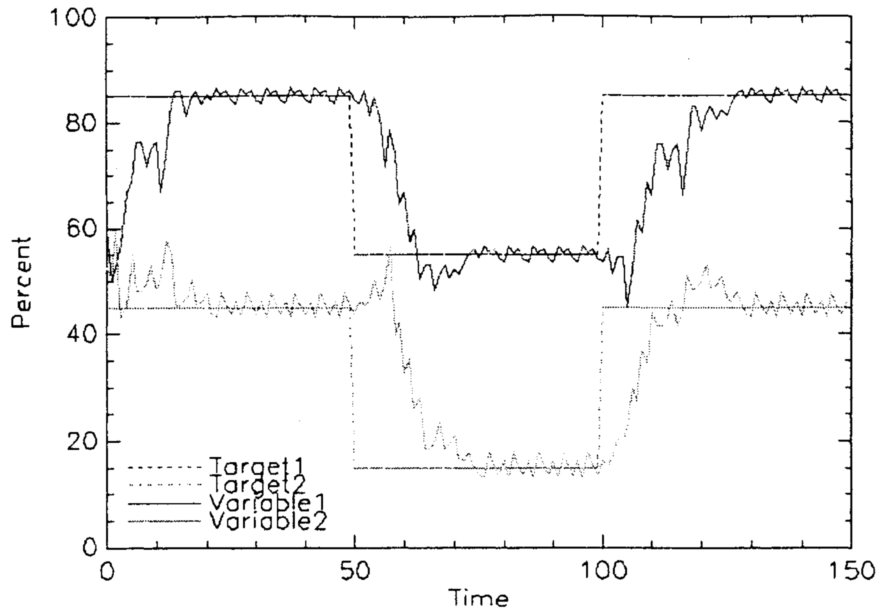


**Figure 2.41, Case 13.6**

No process dynamics, process interaction or noise; Threshold simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; Contraction factor = 0.5; Two contraction levels; ASES = 267

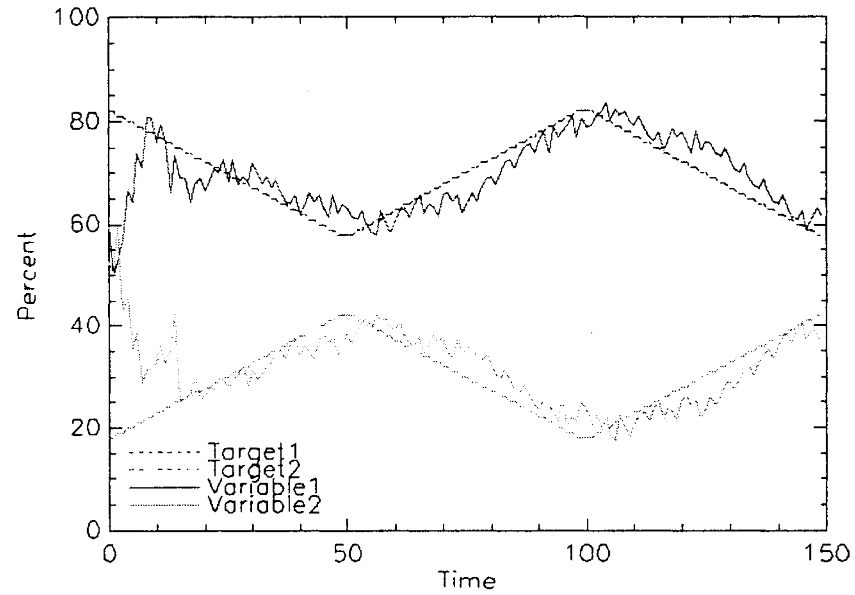


**Figure 2.42, Case 13.7**

No process dynamics, process interaction or noise; Threshold simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; Contraction factor = 0.5; Two contraction levels; ASES = 91

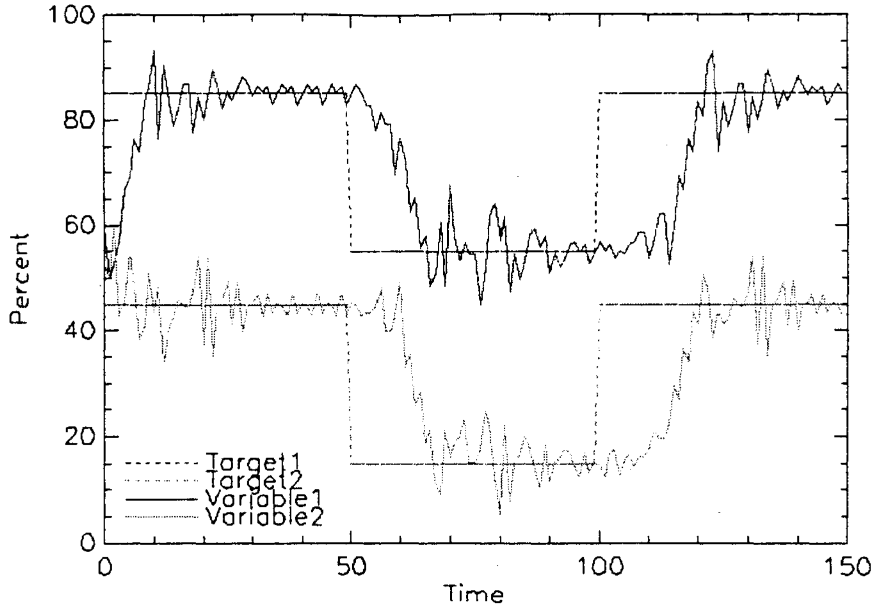


**Figure 2.43, Case 13.8**

No process dynamics, process interaction or noise; Ranking simplex contraction or expansion; Initial simplex size = 10.0; Step target changes; Contraction factor = 0.5; Two contraction levels; ASES = 353

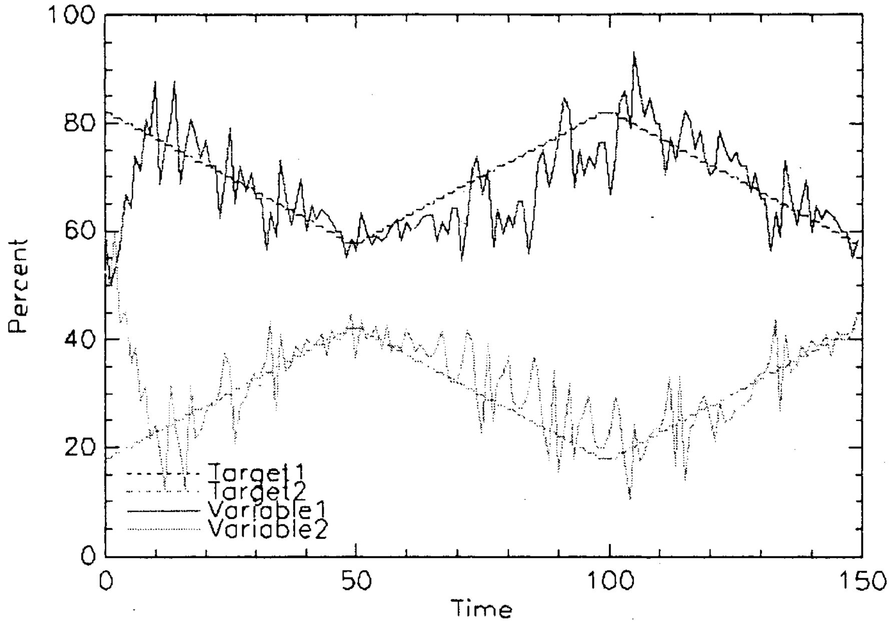


**Figure 2.44, Case 13.9**

No process dynamics, process interaction or noise; Ranking simplex contraction or expansion; Initial simplex size = 10.0; Ramp target changes; Contraction factor = 0.5; Two contraction levels; ASES = 117

# Chapter Three: Modeling The Plant – A Thermo-Mechanical Pulping (TMP) Screening Room

## 3.1 Introduction

The targeted process for the adaptive optimizer, as described in chapter two, is the pulp screening system found in thermo-mechanical pulping mills. In particular, the screening room system of the Bowater TMP plant, Thunder Bay, Ontario, Canada, is modeled for the purpose of further investigation of the adaptive optimizer as a controller. The pulp screens are the manipulated components of the system while pulp qualities, such as mean fibre length distributions, comprise the measured variables. Modeling of the pulp screens per fibre length distributions and other pulp properties is based on early work by Gooding and Kerekes [21] and on later work by Olson, Allison *et. al* [22,23,24,25].

## 3.2 TMP Screening Room

In a traditional system where the screen baskets have had smooth holed apertures, the screening process may be configured as in Figure 3.1, where flow feedback between the primary and secondary screening levels allows for re-screening and re-refining. A simplified schematic of the Bowater screening room is shown in Figure 3.2. With the use of slotted apertures in the main line screen baskets, flow feedback from the secondary screens to the main line screens is not used. The main line screens operate in parallel while their reject streams are thickened, re-refined and re-screened in the rejects system. Currently, the four main line screens use slotted aperture wedge wire baskets while the reject screens have smooth holed apertures. The aperture type determines the parameters for the screen models. The rejects stream flows from the rejects system screens are not recycled back to the main screens feed after reject refining, but are delivered to the cleaners.

### 3.3 Screen Operation

Figure 3.3, from Hautala *et. al* [28,26], shows a view of a modern pressure screen. Pulp enters the screen via the feed port and travels towards the reject port through the center of the basket. There is a foil-style rotor inside the basket that imparts a rotary motion to the pulp suspension inside the basket. The dominant velocity component is tangential to the basket surface, while at the apertures there is a radial velocity in proportion to the flow being taken off as accepts. The fibres that are accepted are those that are small enough or flexible enough to manage the momentum change from tangential to radial before they get swept past the aperture. Long stiff fibres and shives, which are incompletely separated fibre bundles, do not manage that transition, and so tend to remain inside the basket and exit with the rejects. Dilution water or shower water, as it has been traditionally called, is typically applied to the screen basket surface to wash smaller fibres through the screen and to help prevent plugging of the screen. The added dilution water counters the tendency of the screen to increase the consistency on the rejects side, internal to the basket. The rejects pulp stream is then passed to a refining system where the longer fibres are further processed into more flexible fibres with better bonding characteristics.

Pulp screens have been traditionally used to separate wood impurities, such as shives, chop and coarse fibres, from the wood fibres that are usable in the paper making process [27]. Pressure screens can also be used to fractionate pulp streams according to other pulp properties such as fibre length. Fractionation according to fibre length occurs primarily as a probability function, dependent on the fibre characteristics and flexibility, as described in the previous paragraph, and screen operating conditions [22,27]. Slotted holed and smooth holed screen baskets fractionate according to fibre length in a similar way but with different probability functions for the separate fibre length fractions, i.e., fines, medium and longer [23,25].

In general, the probability of fibres of a given length passing through the screen is determined by a number of factors in the operation of the screen, including pulp and dilution flow rates, rotor speed and basket design. The usual way of manipulating screen operation for control of shive removal and mean fibre length of the accepts pulp is to adjust the ratio of the rejects flow to the feed flow. An increased *rejects ratio* will cause a greater portion of the total pulp feed flow to be passed on to the rejects refining system. As well, the change in rejects ratio affects the probabilities of individual fibre length fractions passing through the screen. Slotted holed screens and smooth holed screens different somewhat in the nature of this effect. The rejects ratio can therefore be used to manipulate the screen operation towards controlling optimum conditions involving fibre length distributions of the pulp flow to the paper machines.

### 3.4 Screen Controls

Figure 3.4 shows a graphic from a simulation of the control scheme for the main line screens. Both the shower flow and the rejects flow are proportioned volumetrically to the feed flow by ratio controllers. Typical ratio targets shown are for the main line screens, i.e., 9.2 % and 27.0 %, shower dilution and rejects respectively. All flows are shown in units of litres/min. and represent typical plant operating conditions. Normal operating pulp consistencies are shown in larger font text as 2.70 %, 2.16 % and 3.42 %, - feed flow, accepts flow and rejects flow respectively.

A differential pressure control scheme is used to maintain the accepts flow within reasonable limits, particularly to help prevent screen plugging. Should the differential pressure across the screen (feed to accepts) rise, indicating possible plugging, the controller will cause the accepts flow valve to close, further reducing the accepts flow. A momentary backwash effect should then clear the screen due to more feed and shower flow passing through to the rejects stream. Alternatively, if the differential pressure across the screen decreases, the accepts valve will open and the screen can return to a better flow balance.

The rejects system screens are controlled in the same manner as the main line screens, but operating conditions, particularly the flow ratio targets, have different nominal operating points.

### 3.5 Screen Modeling

Modeling of screen operation as to the probability of fibre length fractions passing through the screen is quantified by 'passage ratio', a function specified in the work of Gooding and Kerekes [21]. The passage ratio of a pulp, $P_p(l)$, as a function of fibre length $l$, has been defined as the ratio between the consistency of the pulp passing through an aperture of the screen, $c_s(l)$, and the feed or upstream consistency $c_u(l)$, i.e.,

$$P_p(l) = \frac{c_s(l)}{c_u(l)}. \tag{3.1}$$

A constant coarseness is assumed for fibres of all lengths.

A plug flow model of the screen flows is based on a material balance as derived by Gooding and Kerekes, and as shown in Figure 3.5.

Assuming perfect radial mixing between the screen basket plate and the rotor and no axial mixing on the feed side of the screen basket, the material balance in an annular element of thickness $dz$ is:

$$Q_z c_z = (Q_z - dQ_z)(c_z - dc_z) + P_p c_z dQ_z, \tag{3.2}$$

where $Q$ refers to *volumetric flow* and $c$ refers to *consistency* and the subscript $z$ is the axial direction, feed to rejects in the screening zone.

Assuming that $dc_z dQ_z$ approximates to zero, this equation can be rewritten as:

$$\frac{dc_z}{c_z} = (P_p - 1)\frac{dQ_z}{Q_z}. \tag{3.3}$$

Integrating Equation (3.3) for the axial length of the screening zone, i.e., from feed to rejects, yields the following:

$$\frac{c_r(l)}{c_f(l)} = \left(\frac{Q_r}{Q_f}\right)^{P_p(l)-1}, \tag{3.4}$$

where subscript *f* refers to *feed* and subscript *r* refers to *rejects*. Since the volumetric reject ratio, $RR_v$, is $\frac{Q_r}{Q_f}$, and the rejects thickening factor, *RTF*, is $\frac{c_r(l)}{c_f(l)}$, Equation 3.4 can be written as:

$$RTF(l) = (RR_v)^{P_p(l)-1}. \tag{3.5}$$

Equation (3.5) can be rewritten as

$$P_p(l) = \frac{\ln(RTF(l))}{\ln(RR_v)} + 1. \tag{3.6}$$

Following the work of Olson *et. al* [22,23,24,25], screen passage ratios as a function of fibre length, $P_p(l)$, have been determined experimentally for smooth holed and slotted screen basket types. An ideal passage ratio function, assuming that fibres less than 2.0 mm in length are to be accepted and fibres of length greater than 2.0 mm are to be rejected, would have a value of 1.0 for the shorter fraction and a value of 0.0 for the longer fraction with a sharp cutoff point at 2.0 mm. Passage ratio functions for both screen category types, slotted and smooth holed, have been determined in many experimental trials by Olson [23,25], and later in a specific trial for the reject screen at the Bowater site, by Hennings [18].

Olson determined that typical screen fibre passage ratio functions take the following equation form:

$$P_p(l) = e^{-\left(\frac{l}{\lambda}\right)^{\beta}}, \qquad (3.7)$$

where $l$ is fibre length in *mm* and $\lambda$ and $\beta$ are constants that shape the function. Olson further specifies that $\beta = 1.0$ for smooth holed screens and that $\beta = 0.5$ for slotted holed screens.

Given that passage ratio functions can be determined experimentally, a screen model can be formulated to calculate the fibre length distribution of the rejects pulp stream if the input feed fibre length distribution is known, as follows:

$$c_r(l) = c_f(l)(RR_v)^{P_p(l)-1}. \qquad (3.8)$$

It follows then that the accepts stream fibre length distribution can be computed as the difference between the feed and rejects streams distributions.

For the Bowater screening room, the constant $\lambda$ was estimated for the rejects screens as having a value of 6.67, taking results from the experimental trial by Hennings [18]. The main line slotted screens $\lambda$ was estimated through interpolation from the experimental results by Olson [25] as being 32.4.

Importantly, Olson also determined that passage ratio functions for 2.0 mm length fibres are independent of varying reject ratios for both smooth and slotted holed screens.

The Bowater main line screens currently have a slot width of 0.255 mm. The slot velocity was estimated as 3.0 m/s. The 2.0 mm passage ratio was then interpolated from Olson's work as being 0.78, and thus the $\lambda$ constant calculated to 32.4. This value was further vetted by interpolation of the wedge wire slotted screen data, again from Olson's trial work [25].

Figure 3.6 shows the passage ratio functions used in the modeling of the Bowater plant screening room.

### 3.6 Modeling the Plant

The overall plant model was simplified to be as shown in Figure 3.7. Transport and mixing lags in the rejects system pulp storage chests can be introduced into the simulation depending on the mill configuration to be modeled. Each connecting line in Figure 3.7 is modeled as pulp flow having variable properties of pulp consistency and pulp flow rate, symbolized by *C* and *Q* respectively. Consistency is defined as the total mass of pulp fibrous content as a percentage of the total mass of pulp content and liquid content in the pulp stream.

$$C = \frac{MassPulp}{(MassPulp + MassLiquid)} \times 100 . \qquad (3.11)$$

For the purpose of plant simulation, each pulp flow contains mass concentration data stored at discrete fibre length intervals and for a limited range of fibre lengths. This data storage mirrors that of the instrumentation found at the Bowater mill, where the on-line Pulp Quality Monitoring (PQM) systems measure various pulp quality properties at discrete intervals of fibre length. These systems provide data over the fibre length range of 0.05 mm to 7.0 mm in 140 discrete bins, where each bin interval is separated by a fibre length of 0.05 mm. In the following equations, where fibre length *l* is a variable, it is assumed that this variable *l* refers to discrete data in the above format.

The consistency of each pulp flow is also recorded as a fibre length distribution by concentration, where the fibre length *l* ranges from 0.05 mm. to 7.0 mm, and the relationship between concentration *c* and the consistency *C* is as follows:

$$C = \sum_{l=0.05}^{l=7.0} c(l). \tag{3.12}$$

The concentration $c(l)$ is the fraction by the percentage weight of the fibrous material of a given length *l* to the total weight of fibrous and liquid material in the pulp stream, i.e., for all fibre lengths in the measured range. The pulp flow $Q$ is the pulp flow rate measured as a volumetric rate, typically *litres/minute*, including all fibrous material and liquids. Pulp flows are also recorded as mass flow distributions by fibre length as follows:

$$m(l) = \rho Q c(l), \tag{3.13}$$

where $\rho$ is the density of the pulp.

Pulp flows can be combined by addition or distributed by subtraction using mass flow distributions. For example, if $Q_{Total} = Q_1 + Q_2$, then the total flow consistency $C_{Total}$ can be computed as follows:

$$C_{Total} = \frac{(C_1Q_1 + C_2Q_2)}{(Q_{Total})}. \tag{3.14}$$

The mass flow distribution is calculated as

$$m_{Total}(l) = m_1(l) + m_2(l), \tag{3.15}$$

and fibre length concentration distribution is computed as

$$c_{Total}(l) = \frac{m_{total}(l)}{\rho Q_{Total}}. \tag{3.16}$$

Each pressure screen is modeled with three pulp flows, i.e. feed, accepts and rejects, and a shower flow. Other screen attributes are the volumetric rejects ratio, $RR_v$ and the passage ratio lambda parameter as per Olson's screen models, i.e., as per Equation 3.7. The main line screens passage ratio function is

$$P_m = e^{-\left(\frac{l}{32.4}\right)^{0.5}} \tag{3.17}$$

while the rejects screen passage ratio function is

$$P_r = e^{-\left(\frac{l}{6.67}\right)^{1.0}}. \tag{3.18}$$

A screen balance uses the passage ratio function and the current operating point (volumetric rejects ratio) in calculating the rejects flow concentration distribution as in Equation 3.8 (i.e., $c_r(l) = c_f(l)(RR_v)^{P_p(l)-1}$), where $P_p$ is either $P_m$ or $P_r$ as above in Equations 3.17 or 3.18, and where $c_r$ is the rejects flow concentration and $c_f$ is the feed flow concentration for a given fibre length $l$.

The accepts flow concentration distribution is calculated as follows:

$$c_a(l) = \frac{(m_f(l) - m_r(l))}{\rho Q_a}, \tag{3.19}$$

where $c_a$ is the accepts flow concentration, $Q_a$ is the volumetric accepts flow rate, $m_f$ is the feed mass flow and $m_r$ is the rejects mass flow.

Pulp storage time in the rejects system pulp chests is modeled as a simple delay. Typical delays in terms of PQM sample times have been determined from nominal mill production rates and tank sizes to be in the range of five to nine PQM sample times, where a sample time is ten minutes, for each of the two storage tanks.

A literature search failed to find any significant data regarding the effects of TMP refiner operation on pulp fibre distributions. Therefore, a simple model is proposed whereby the refiner reduces the longer length fibre fraction concentration distributions and increases the shorter length fraction concentrations by a given percentage factor, while maintaining a material balance on the refiner input and output flows. For the longer fraction (i.e., 2.0 mm to 7.0 mm), each fibre length bin concentration is reduced by a given factor, typically 0.1 to 0.5, as follows:

$$c(l) = (1.0 - f)c(l), \tag{3.20}$$

where $f$ is the concentration reduction factor. The finer fibre fractions are categorized as the short fraction, i.e., up to 0.4 mm in length, and the medium fraction, i.e., between 0.4 mm and 2.0 mm in length. For the purpose of simulation, each discrete fibre length bin is located at intervals of 0.05 mm. Since the ratio of the number of bins in the longer fibre fraction to the sum of number of short and medium fibre fraction bins is 100 bins to 40 bins, or a ratio of 5 to 2, the total concentration loss for each set of five bins in the longer fraction is distributed to two bins in the short and medium fibre length fractions. For example, the two shortest fibre length bins are increased in concentration as follows:

$$c(0.05) = c(0.05) + \frac{1}{2}\sum_{l=2.00}^{l=2.25} fc(l), \text{ and} \qquad (3.21)$$

$$c(0.10) = c(0.10) + \frac{1}{2}\sum_{l=2.00}^{l=2.25} fc(l). \qquad (3.22)$$

This operation is repeated twenty times, for each pair of consecutive short and medium fibre length concentration bins, i.e, for fibre length bins 0.05 to 1.95 mm. For example, the next pair of bins are increased accordingly as $c(0.15) = c(0.15) + \frac{1}{2}\sum_{l=2.30}^{l=2.55} fc(l)$ and $c(0.20) = c(0.20) + \frac{1}{2}\sum_{l=2.30}^{l=2.55} fc(l)$. Since the bins contain mass concentrations and not fibre counts, the summation does not double the number of fibres in the shorter bins.

In terms of the control scheme manipulated variables, i.e., the screen reject ratios, the plant model can be seen to be further simplified as shown in Figure 3.8. Since the optimizer has been restricted to two dimensions, the main screens are to be manipulated together in parallel as the first controller output, and with the reject screen manipulation as the second controller output. This is normal operating practice for the mill. However, given that the pulp sources for the two main line latency chests may be different, the main line screens could be manipulated (in pairs) separately as the two controller outputs.

### 3.7 Screen Simulation

Figure 3.4 shows typical operating and quality conditions for each of the main line screens. All flows are in units of litres per minute. The feed flow normally has an overall consistency of about 2.7 %, which thickens to about 3.4 % in the rejects stream, if the volumetric rejects to feed flow ratio is 27.0 % and the volumetric shower to feed flow ratio is 9.2 %, as shown.

Figure 3.9 plots operational pulp quality conditions involving the operation of the main line screens. The plot shows typical pulp concentration fibre length distributions for the feed flow, as a percentage concentration, and the subsequent resultant concentrations for the rejects and accepts flows, again as percentages, given the same operating conditions as shown in Figure 3.4. Note that the rejects flow has the highest concentrations generally, while the accepts flow has the lowest concentrations. Figure 3.10 shows screen pulp (mass) flows by fibre length distribution, where the feed flow is the largest and the rejects flow is the smallest and where units are *kg/min*.

The feed flow concentration, as shown in Figure 3.9, follows a log normal distribution, in which the logarithm of a variable has a normal distribution. Disturbances will be introduced into the system in later tests by adjusting the *mean* value for the feed flow log normal distribution for the pulp mass concentration. The feed flow concentration distribution is calculated from the feed flow consistency $C$ as follows:

$$c(l) = \frac{C}{l\sqrt{2\pi}\beta} e^{-\frac{(\ln(l)-\alpha)^2}{2\beta^2}}, \tag{3.23}$$

where $\alpha = \ln(mean) - \frac{\beta^2}{2}$. The variable $\beta$ is the *variance* of the distribution.

Removal efficiency [22,23,24,25] is defined as the mass flow rate of the fibres in the reject stream divided by the mass flow rate of fibres in the feed stream, as follows:

$$e(l) = \frac{Q_r c_r(l)}{Q_f c_f(l)} = RR_v \frac{c_r(l)}{c_f(l)}. \qquad (3.24)$$

It is a measure of how well the screen rejects the fibres at each fibre length. An ideal screen would have a removal efficiency of 0.0 for the finer fibre fractions and a removal efficiency of 1.0 for the longer fibre fraction.

Figure 3.11 shows the removal efficiency by fibre length for the slotted main line screens under the same conditions as above, i.e., with the same feed distribution and a volumetric rejects ratio, $RR_v$, set to 0.270. Figure 3.12 shows the removal efficiency function for the slotted main line screens with the same feed fibre length distribution, but where the volumetric rejects ratio has been set to 0.178, which is a typical operating point for the rejects system screen. In general, the overall removal efficiency improves as the volumetric rejects ratio is increased.

Figures 3.13 and 3.14 offer the same removal efficiency functions and using the same operating conditions respectively as Figures 3.11 and 3.12, except that the screens have been configured as holed screens, instead of slotted screens. It can be seen that holed screens provide higher removal efficiencies at all fibre lengths, and particularly higher at the longer fibre lengths. As expected, given the passage ratio functions for each type of screen, holed screens provide better fractionation of the feed pulp, i.e., a more efficient removal of the longer fibres.

## 1.7 3.8 Plant Simulation for Optimization Using Performance Functions

Plant simulation for performance function optimization uses the screening room configuration, as shown in Figure 3.7, with two differences. First, the main screens and rejects screen are always configured as the same type, i.e., all screens use the slotted screen passage ratio, or all screens use the holed screen passage ratio. This was done to enable a fair comparison of the system fractionation ability for each type of screen. Secondly, pulp storage delay in the rejects system was eliminated from the simulation. This was again done to provide a better fractionation picture and also because the adaptive optimizer is later applied to control the system fractionation, and delay is difficult for the optimizer to handle. One of the objectives of the later simulations is to compare the range-ability of fractionation control for each type of screen. Range-ability can be described as the range of manipulated variable values over which the manipulation causes a sufficient effect on the controlled variable. For example, control valves may saturate and provide no additional effect on the plant measured values even though the controller is asking for more.

For the performance function simulations, each of the manipulated variables, i.e., main line and reject screen volumetric reject ratios, are adjusted independently over a range of possible operating values. For each set of operating values, a performance function index value is calculated at system steady state. This performance function is then plotted against the volumetric reject ratio operating values, with the volumetric reject ratios as independent variables. In the following cases, the manipulated variables are not functionally constrained within the limits of the range of possible reject ratio values. Therefore, the performance functions, for optimization purposes, are considered unconstrained.

The main line screen volumetric reject ratios are manipulated independently (in parallel) over a range from 0.20 to 0.35 while the reject screen volumetric reject ratio is manipulated independently over a range of 0.15 to 0.30.

In the first set of simulations involving the performance function simulations, all of the screens are configured with holed baskets. The second set of performance function simulations repeat the conditions of the first set, but the screens are configured with slotted baskets.

### 3.8.1 Performance Function Simulations with Holed Screens

The first performance function, or performance index *PI*, is the sum of the longer fraction concentrations in the screening system out flow as a percentage of the total sum of all fibre length concentrations in the system out flow. This calculation is as follows:

$$PI_1 = \frac{\sum_{l=2.0}^{l=6.95} c(l)}{C} 100, \tag{3.25}$$

where $C$ is the consistency of the screening system accepts flow as per equation 3.12. This performance index ideally has a value of 0.0 percent for complete fractionation, i.e., complete removal of the longer fraction fibres.

Figure 3.15 shows the performance index $PI_1$ as the main line and rejects screens volumetric reject ratios are varied independently. As expected from the results of the steady state simulations, a minimum value for the performance index occurs when both screens have a maximum volumetric rejects ratio $RR_v$.

The second performance function simulation calculates the performance index as follows:

$$PI_2 = \left( \overline{C}_S - \frac{\sum_{l=0.05}^{l=0.35} c(l)}{C} 100 \right)^2 , \tag{3.26}$$

where $\overline{C}_S$ is a target total concentration percentage for the short fibre length fraction and has a value of 8.17 per cent. This is the error square of the shorter fraction, and minimally should be zero. The target value is determined from operating conditions, and in this case, is chosen as being the average value of the short fraction total concentration percentage for the whole range of reject ratio manipulation.

The simulation results are shown in Figure 3.16. Minimum performance index values occur when the main screens are operated with a volumetric rejects ratio in the mid-range from about 0.24 to 0.30. It is noted that as the main line screens reject ratio is increased, the rejects system reject ratio is decreased, in order to achieve a minimum performance index value. The main screens play the primary role in fractionation and shive removal since the rejects system handles less fibre overall.

The third performance function simulation calculates the performance index as follows:

$$PI_3 = \left( \overline{C}_M - \frac{\sum_{l=0.40}^{l=1.95} c(l)}{C} 100 \right)^2 , \tag{3.27}$$

where $\overline{C}_M$ is a target total concentration percentage, for the medium fibre length fraction, and it is chosen similarly to the short fraction target, having a value of 80.57 per cent. The simulation results are shown in Figure 3.17 and the landscape of the performance function is similar to the short fraction performance function.

The fourth performance function simulation calculates the performance index as follows:

$$PI_4 = \left( \overline{C}_L - \frac{\sum_{l=2.0}^{l=6.95} c(l)}{C} 100 \right)^2 , \tag{3.28}$$

where $\overline{C}_L$ is a target total concentration percentage, for the long fibre length fraction, and it is chosen similarly to the short fraction target, having a value of 11.25 per cent. The simulation results are shown in Figure 3.18 and the landscape of the performance function is similar to the short and medium fraction performance function.

The fifth performance function simulation calculates the performance index as follows:

$$PI_5 = W_3 PI_3 + PI_4 , \tag{3.29}$$

where $W_3$ is a weighting factor used to balance the individual medium and long fibre fraction error square values. In this case, this weighting factor is arbitrarily set to 10.0 to provide an even weighting of each fraction error square value. In practice, mill operations may have a different weighting factor. The simulation results are shown in Figure 3.19 and the landscape of this sum of error squares performance function is similar to previous error square performance functions.

### 3.8.2 Performance Function Simulations with Slotted Screens

The performance function simulations for the second set are identical to the first set except that the screens, both main line and rejects system, are configured as screens with slotted basket types. The results are shown in Figures 3.20 through 3.25 and these can be compared with the results from Figures 3.15 through 3.19 respectively, which are the results for the holed screens.

In general, the results for either screen types are similar. Comparing Figures 3.15 and 3.20 it can be seen that the holed screens can reduce the longer fraction total concentration percentage from the range of 10.8 % to 11.8 % produced by the slotted screens, to an approximate range of 10.4 to 11.4 %. The holed screens have some increased measure of longer fibre removal and thus provide better fractionation of the pulp. In both cases, the feed flow longer fraction total concentration percentage has a value of 12.9 per cent.

Also, it can be seen by comparing the individual fibre fraction performance function results, i.e., the short, medium and long fractions, for each screen type, that the performance function landscapes for the slotted screens are shallower than the corresponding landscapes for the holed screens. The holed screen performance function landscapes for the individual fibre fractions, as shown in Figures 3.16 through 3.18, show larger values at the extremes of the manipulated rejects ratios than for the slotted screen results as per Figures 3.21 through 3.23. This indicates that the errors are higher in value at these extremes and therefore the holed screens have a wider range of control than the slotted screens. Again, this is expected from the passage ratio effects for each screen type.

### 3.9 Fractionation Control of the Screening System with the Adaptive Optimizer

The adaptive optimizer is applied to fibre fractionation control in the screening system in the following simulation cases where the performance function is $PI_5$, as per Equation 3.29, i.e., the performance index is the sum of error squares for the total concentration percentages for the medium and long fibre fractions. In all cases, all screens are either holed or slotted, as indicated in each individual case, and the rejects system delay is removed for all but the last two cases. In all cases, the system feed flow fibre length distribution initially has a mean value of 0.600 mm, and is step changed sequentially to values of 0.595, 0.605 and 0.600 mm, for the purpose of providing disturbances to the system.

For each simulation case, the figure showing the results of the disturbance changes is a set of three time series plots, where each time unit is one PQM sample time, which is typically ten minutes in an industrial application. The upper plot shows the manipulated variables, i.e., the main line screens and the rejects system screen volumetric rejects ratios (*Main RR* and *Reject RR*), and the performance index (*Perf. Index*), plotted as a function of time. The performance index value has been scaled up, i.e., multiplied by a factor of 50.0, simply to make it significantly more visible on the same vertical axis scale. The middle plot shows the medium fraction total concentration percentage value (*Med. Frac.*), along with the medium fraction total concentration percentage target value (*Target*), plotted as a function of time. The lower plot shows the long fraction total concentration percentage value (*Long Frac.*), along with the long fraction total concentration percentage target value (*Target*), plotted as a function of time. The actual time of each respective disturbance change can be seen from the performance index chart, or either of the fraction target and measurement charts. When the disturbance is applied, the performance index immediately becomes non-zero and there is an immediate measurement deviation for each fraction total concentration percentage. For example, in Figure 3.25, the approximate times of each respective disturbance is at 60, 150 and 290 time units.

The adaptive optimizer always has an initial simplex size of 10.0 percent of the manipulated variable range, which is the volumetric rejects ratio range of 0.15 to 0.35. The refining factor is set to a value of 0.25, but is changed in one case, for comparison, to 0.5. Later simulation cases provide simplex contraction and expansion, using both adaptive optimizer methods for contraction and expansion. The last pair of simulation cases add delay to the rejects system.

The first simulation case results are shown in Figure 3.25. The main line and reject screens are all holed types. There is no simplex contraction or expansion. The adaptive optimizer is able to respond to the disturbances and maintain the fibre fraction measured variables on target in a repeatable manner.

The second simulation case results are shown in Figure 3.26. The main line and reject screens are all slotted types. There is no simplex contraction or expansion. For each disturbance step change, the resulting change in manipulated variable, i.e., the volumetric rejects ration for each screen, is somewhat larger than the corresponding manipulated variable changes for the holed screens, as shown in Figure 3.25. Again, this is evidence of a lower control range for the slotted screens, as compared to the holed screens.

From an operating point where the system feed flow fibre length distribution has a mean value of 0.600 mm, the system was first disturbed to a lower mean value and then to a higher mean value for both types of screens in order to ascertain the range over which control can be effected. For holed screens, the lower limit for control is a mean value of 0.585, where the rejects screen rejects ratio saturated near a value of zero. The higher limit for control is a mean value of 0.615, where the main line screens rejects ratio leveled out at a value of 0.40 and the medium fraction sustained an offset error. For slotted screens, the respective limits in mean disturbances are 0.590 and 0.610, which constitutes a smaller control range than for holed screens.

The third simulation case results are shown in Figure 3.27. The main line and reject screens are all holed types. There is no simplex contraction or expansion. The refining factor is changed from a value of 0.25 to a value of 0.50. The refining system is able to convert more of the longer fraction fibre to the shorter fractions, and the fraction targets have changed somewhat. The longer fraction target has decreased in value, while the medium fraction target has increased in value, indicating this extra conversion. The volumetric rejects ratio manipulated variable excursions have decreased in value under these conditions, since less of the longer fibre fraction is passed to the rejects system and more of the longer fibre fraction is converted to the shorter fractions in the rejects system.

The fourth simulation case results are shown in Figure 3.28. The main line and reject screens are all holed types. The method of simplex contraction or expansion is the threshold method and the performance index threshold value for contraction or expansion is 0.001. If the threshold value is smaller than this value, the simplex is not able to contract from its initial size, since all simplex point reflections around the optimum point produce performance index values that are greater than this threshold value.

The refining factor is 0.25. Comparing Figure 3.28 to Figure 3.25, it can be seen that simplex contraction and expansion, using the threshold method, has lowered the rise time of the step response of the control system to the disturbances. It is also noted in this case that the rejects screen manipulation is more aggressive under these conditions, and as a result, on larger disturbances, i.e., when the feed distribution mean changed from 0.595 to 0.605, the manipulated variables settled to a different position than for the case without simplex contraction or expansion, as shown in Figure 3.25. This result is not unexpected, since the general landscape of the performance function, as shown in Figure 3.19, is a flat valley where similar minimal values can be found at various combinations of the manipulated variable positions.

The fifth simulation case results are shown in Figure 3.29. The main line and reject screens are all holed types. The method of simplex contraction or expansion is the ranking method. On the first disturbance, i.e., a change in fibre length distribution mean of 0.600 to 0.595, there is more overshoot than seen with the threshold method in Figure 3.28. In the larger disturbance, i.e., a change in fibre length distribution mean of 0.595 to 0.605, the optimizer simplex tends to drift more in the performance function valley than for the previous case, since the ranking method of contraction and expansion, as outlined in Chapter 2, does not provide the same tracking ability as the threshold method.

The last simulation case results are shown in Figures 3.30 and 3.31 where delays are introduced in the rejects system in the form of the pulp storage chests. In the former of these two cases, the delay amounts to four sample periods, while in the latter case the delay is eight sample periods. The other conditions are the same as the first case in this series, i.e., with holed screens, no simplex contraction and the same step changes in disturbances. The delay in the rejects system causes some drifting of the manipulated variables and the errors are larger at each step change in disturbance. However, most of the fibre is still processed by the main line screens, where there is no delay in the path, and the fractionation is sufficiently controlled.

### 3.10 General Conclusions

- With a screening room configured as per the scheme at the Bowater Mill, Thunder Bay, the main line screens and the rejects system screen volumetric rejects ratios can be manipulated together to provide control of fibre length fractionation in the screening room outflow, as measured by the total concentration percentages of the medium and long fibre fractions.
- The volumetric rejects ratios manipulation for the screens have limited ranges, since a lower value restricts rejects system refining and a higher value may cause screen plugging.
- Fibre length fractionation control can be achieved with screens having baskets with apertures of either the holed or slotted types. However, the holed type screen baskets provides better fractionation, i.e., removal of the longer fibre fraction, than the slotted screen type. Thus, the holed screen type has more control range than the slotted screen type.
- In the rejects system refining, a better conversion rate of longer fibre fraction to the shorter fibre fractions will increase the control range of the fibre length fractionation for the screens.

- The adaptive optimizer can be effectively applied in controlling the screening system outflow fibre length fractionation using a performance function that is the sum of weighted error squares for the total concentration percentages for the medium and long fibre fractions.
- Using the contraction and expansion methods with the adaptive optimizer can improve the speed of response to step changes in disturbances. For a performance function of the type having error squares from targets, the threshold method of contraction and expansion provides more effective control than the ranking method of contraction and expansion. The choice of threshold value is critical in that it must be large enough to allow contraction after a normal simplex reflection and small enough to allow expansion after a contraction. The use of the ranking method of contraction and expansion for this system gives larger steady state errors and thus causes the simplex to drift in the performance function valley.
- For a performance function of the type having error squares from targets, the volumetric rejects ratios may settle at different steady state values for similar disturbances. This is due to the nature of the flat performance function landscape, which is a valley of similar values for a range of main screen and rejects system screen volumetric rejects ratio values. Therefore, the rate at which one measured variable changes, or the rate at which one manipulated variable is applied, affects the final position of both manipulated variables at steady state.
- Delay within the system, in the form of pulp storage chests in the rejects system, provides some difficulty for the adaptive optimizer. However, the optimizer is still able to control the outflow fractionation in a system having significant delays

of these types. This may be attributed to the fact that the main line screen processing has no delay and the majority of fibre passes to the system outflow in this manner.

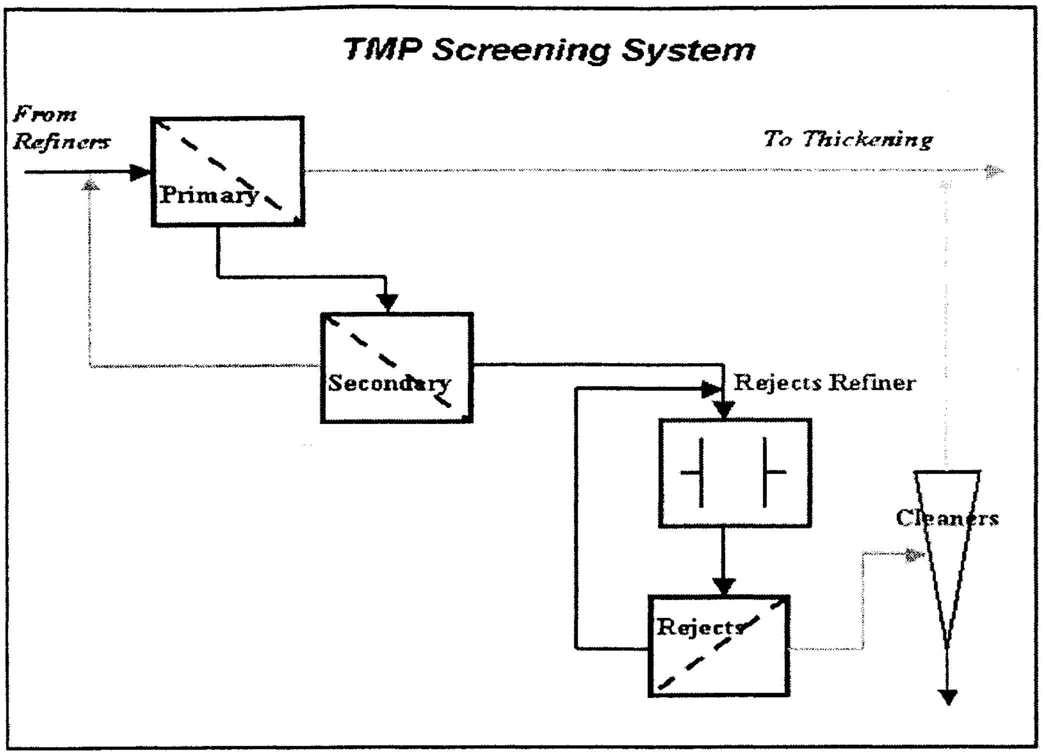


Figure 3.1 Traditional Screening Room Schematic

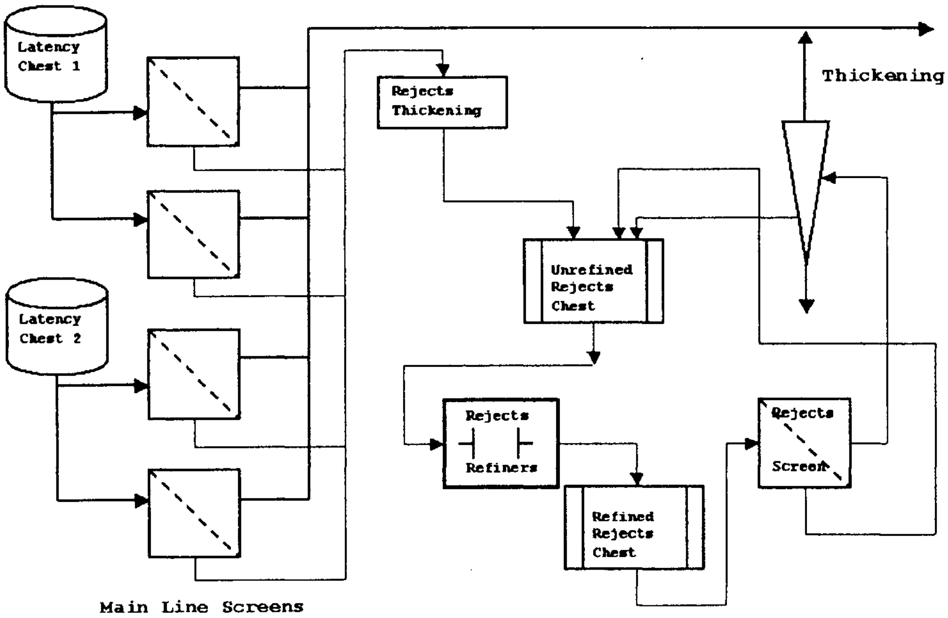


Figure 3.2 Bowater Screening Room Schematic

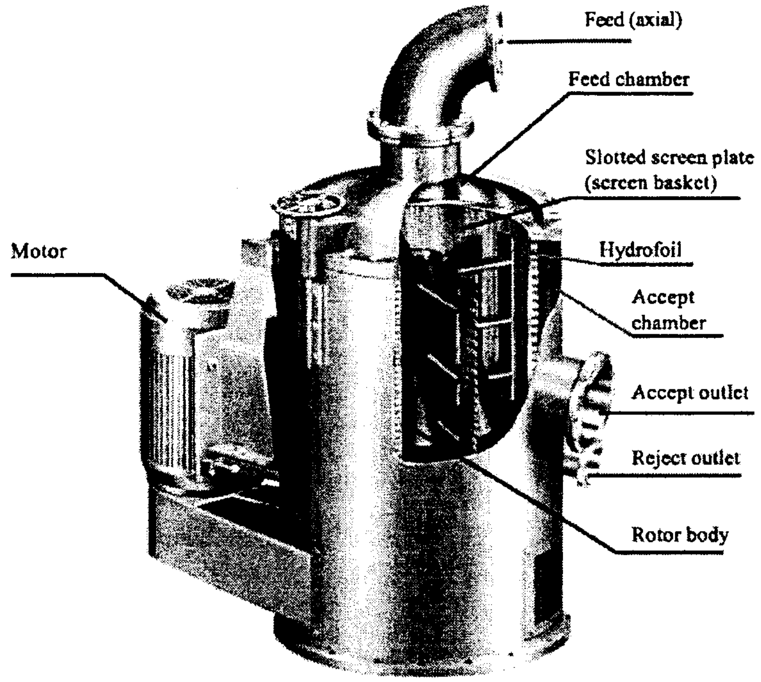


Figure 3.3 Modern Pressure Screen

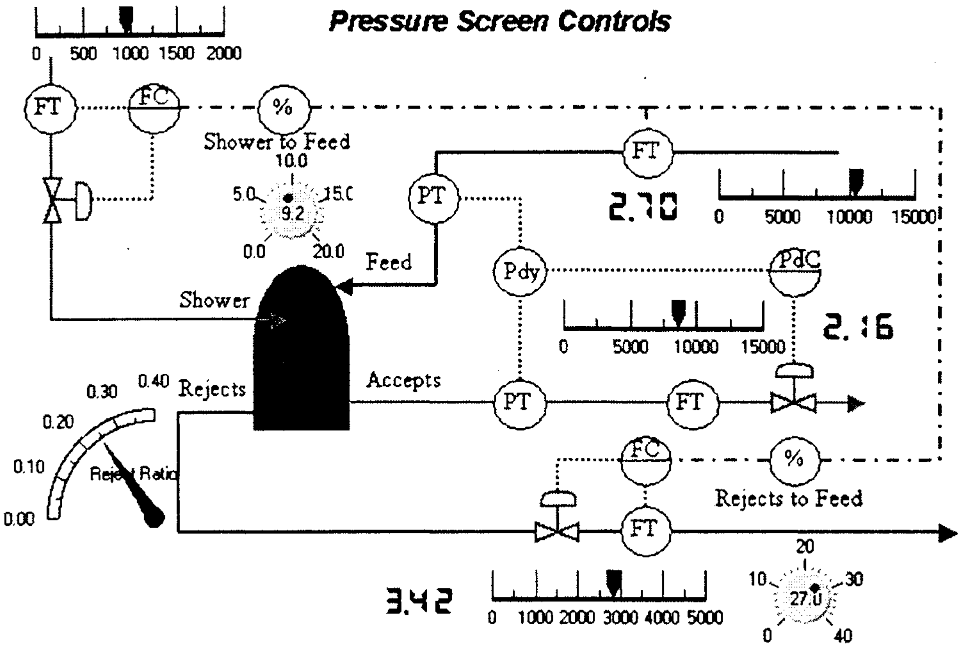
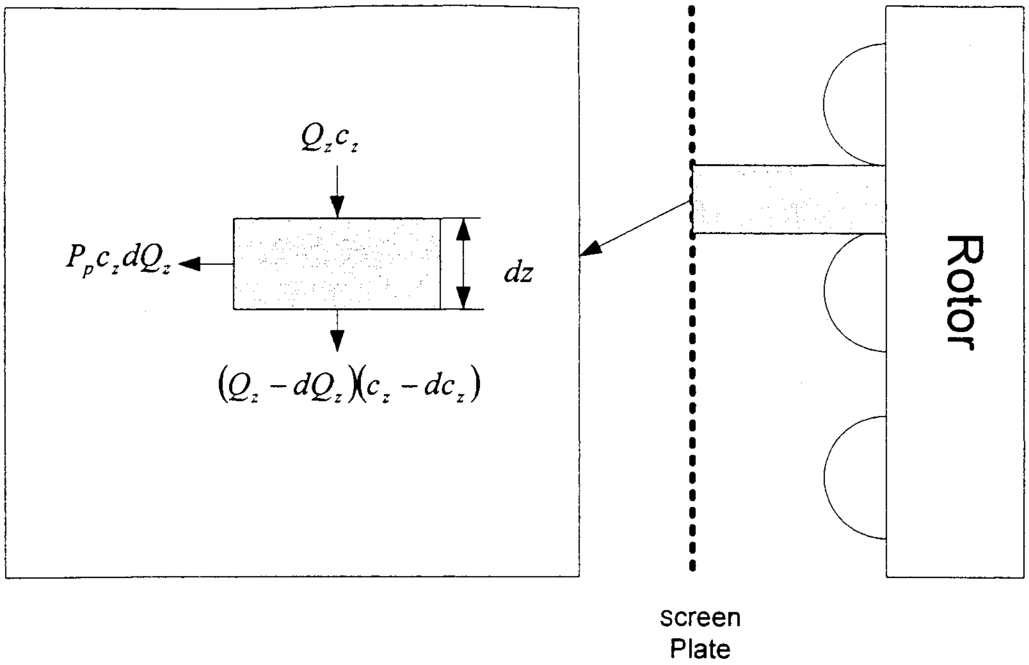


Figure 3.4 Screen Controls



Material balance around an annular element, shown in cross-section.

**Figure 3.5 Material Balance – Plug Flow Model**

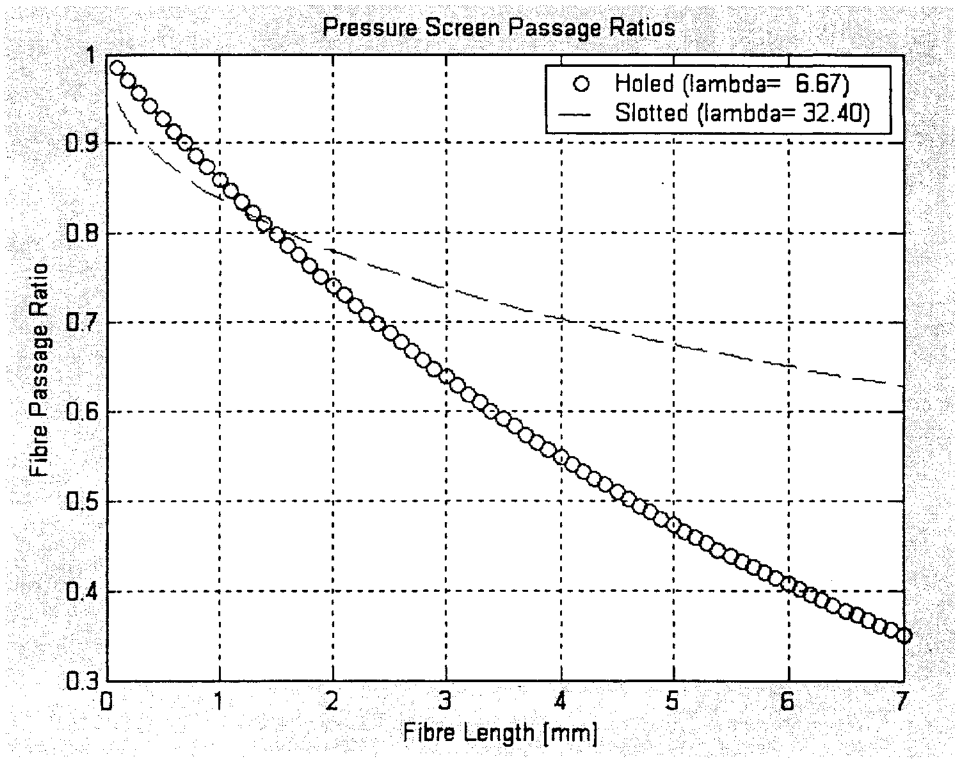


**Figure 3.6 Main Line and Rejects Screen Passage Ratio Functions**

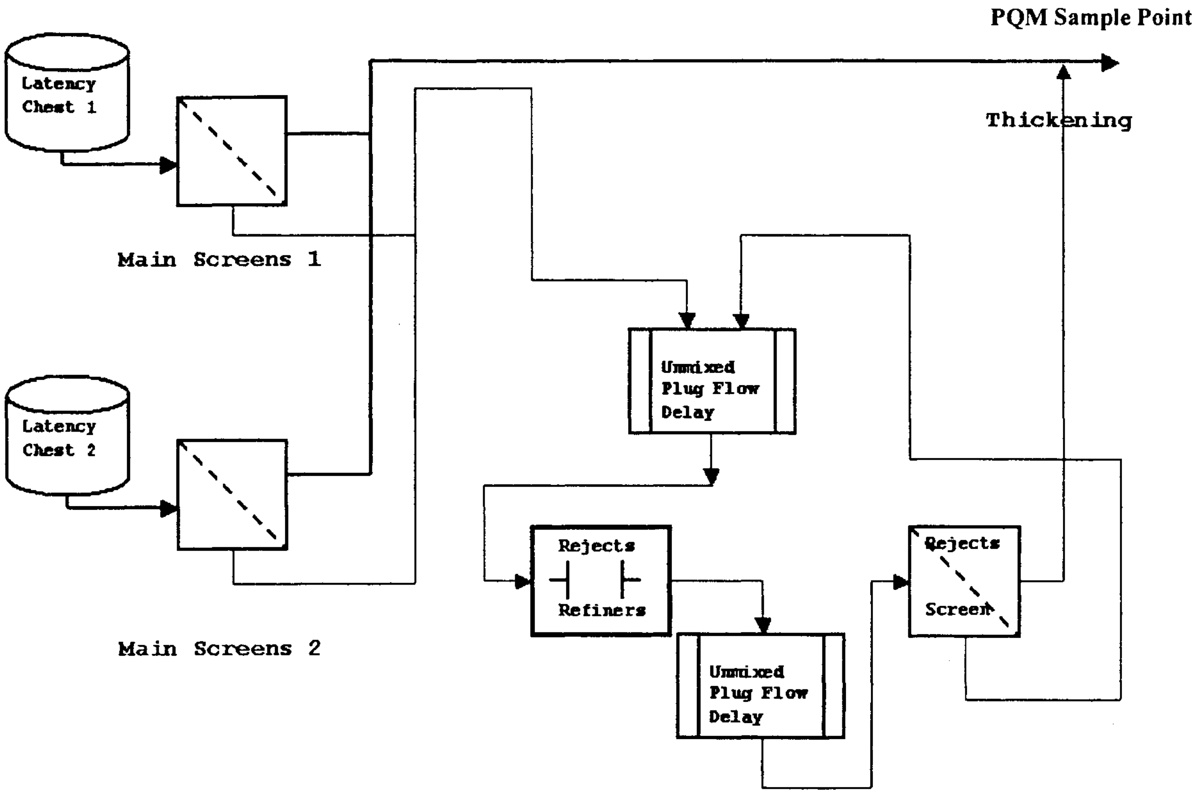


Figure 3.7 Overall Plant Model of Bowater Screening Room

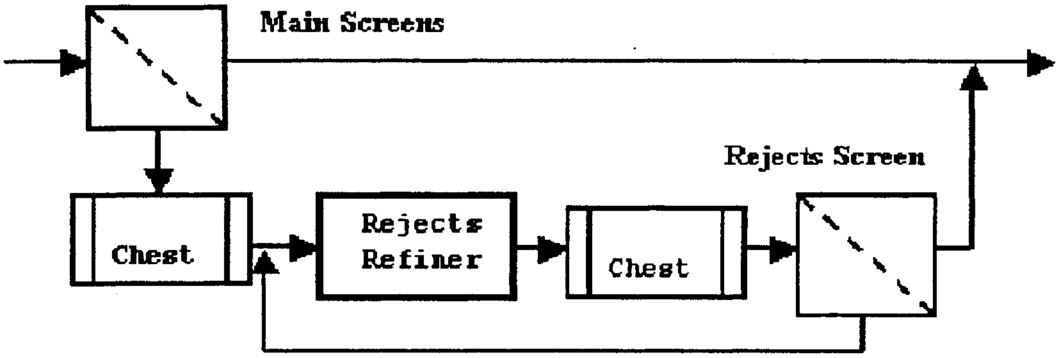


Figure 3.8 Control Structure of Bowater Screening Room

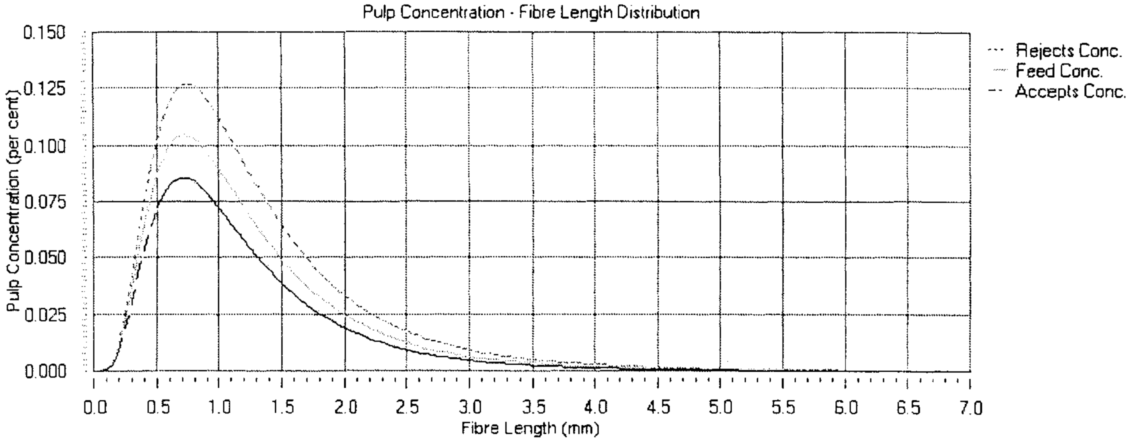


**Figure 3.9 Typical Main Line Screen Flow Concentrations**

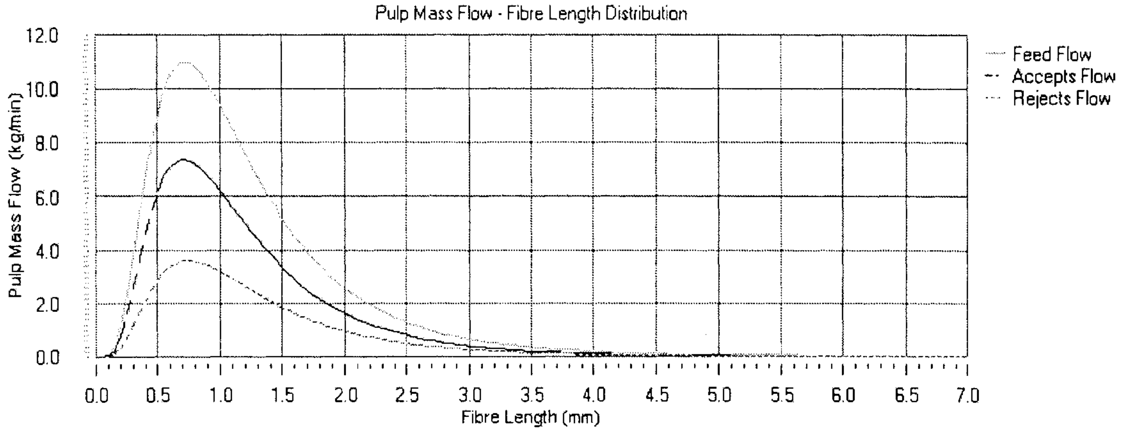


**Figure 3.10 Typical Main Line Screen Mass Flows**

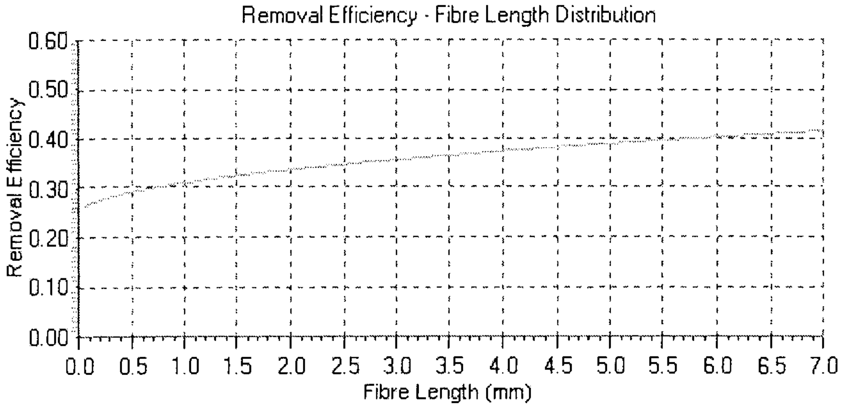


**Figure 3.11 Slotted Main Line Screen, RRv = 0.270**

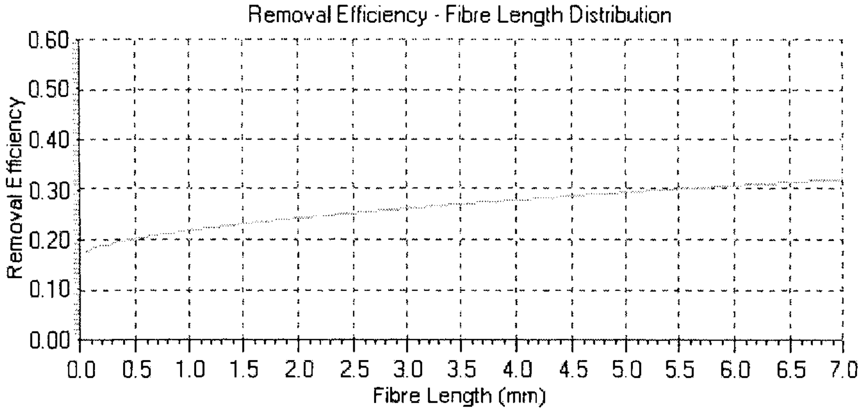


**Figure 3.12 Slotted Main Line Screen, RRv = 0.178**

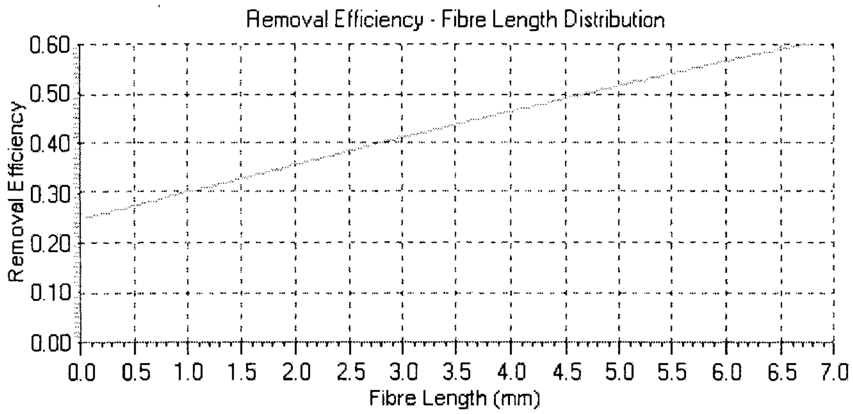


**Figure 3.13 Holed Main Line Screen, RRv = 0.270**

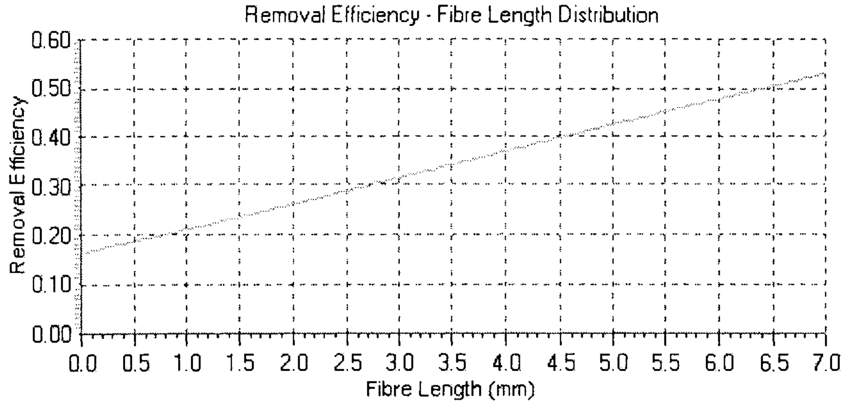


**Figure 3.14 Holed Main Line Screen, RRv = 0.178**

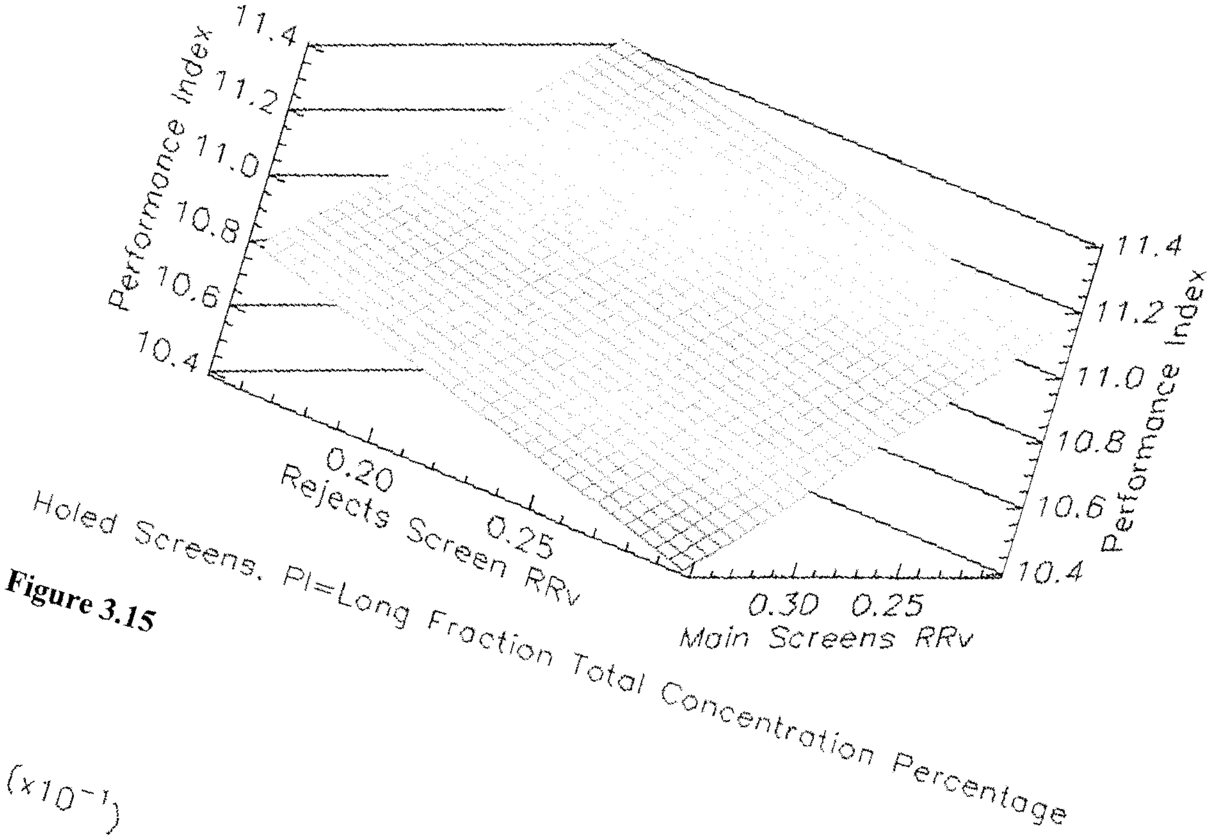


Figure 3.15

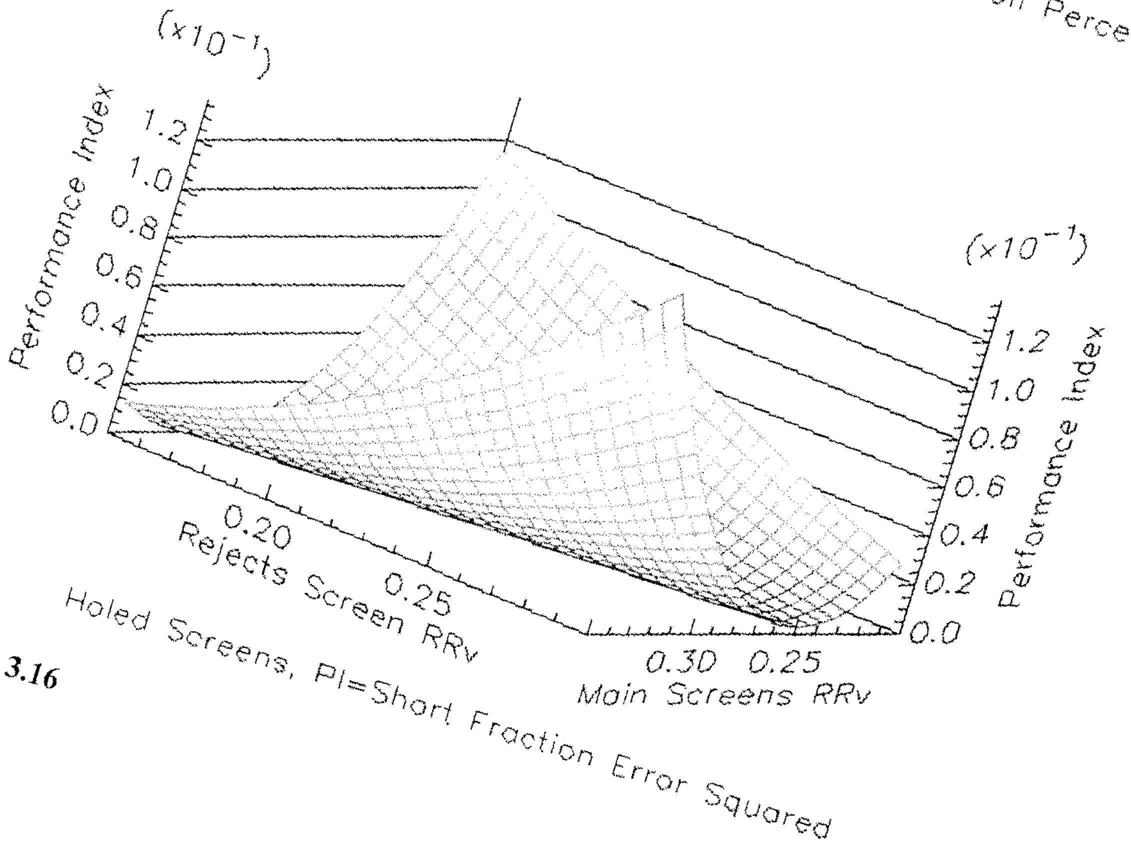


Figure 3.16

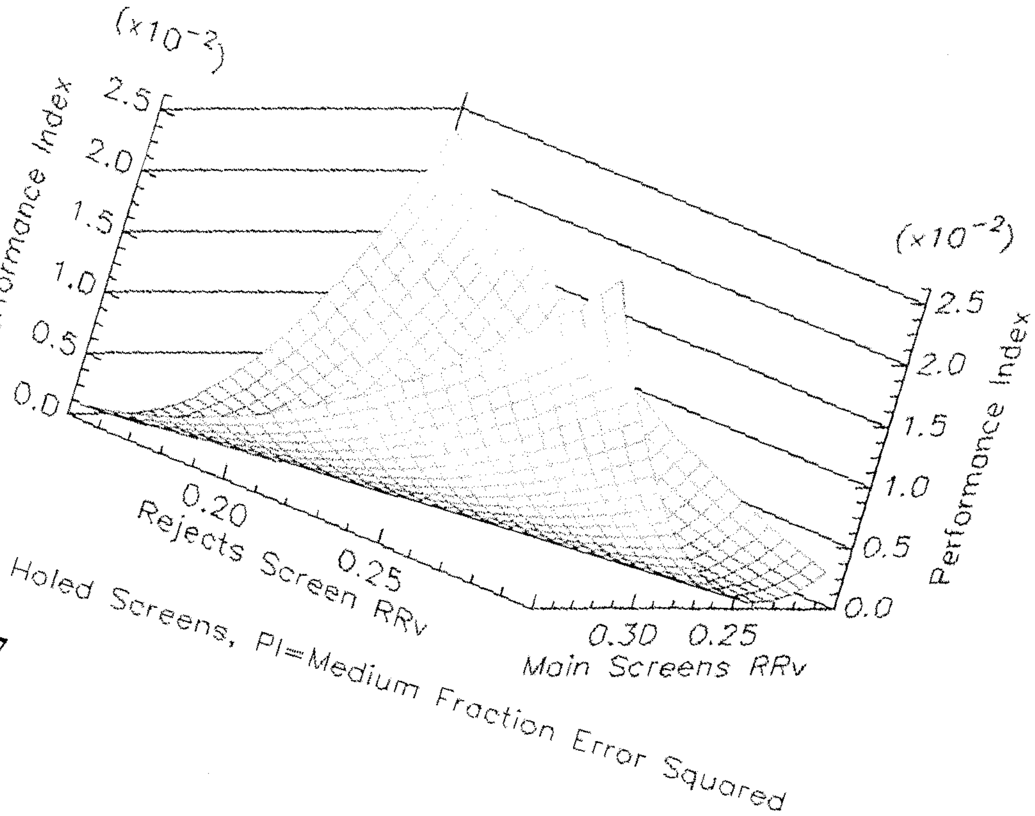


Figure 3.17

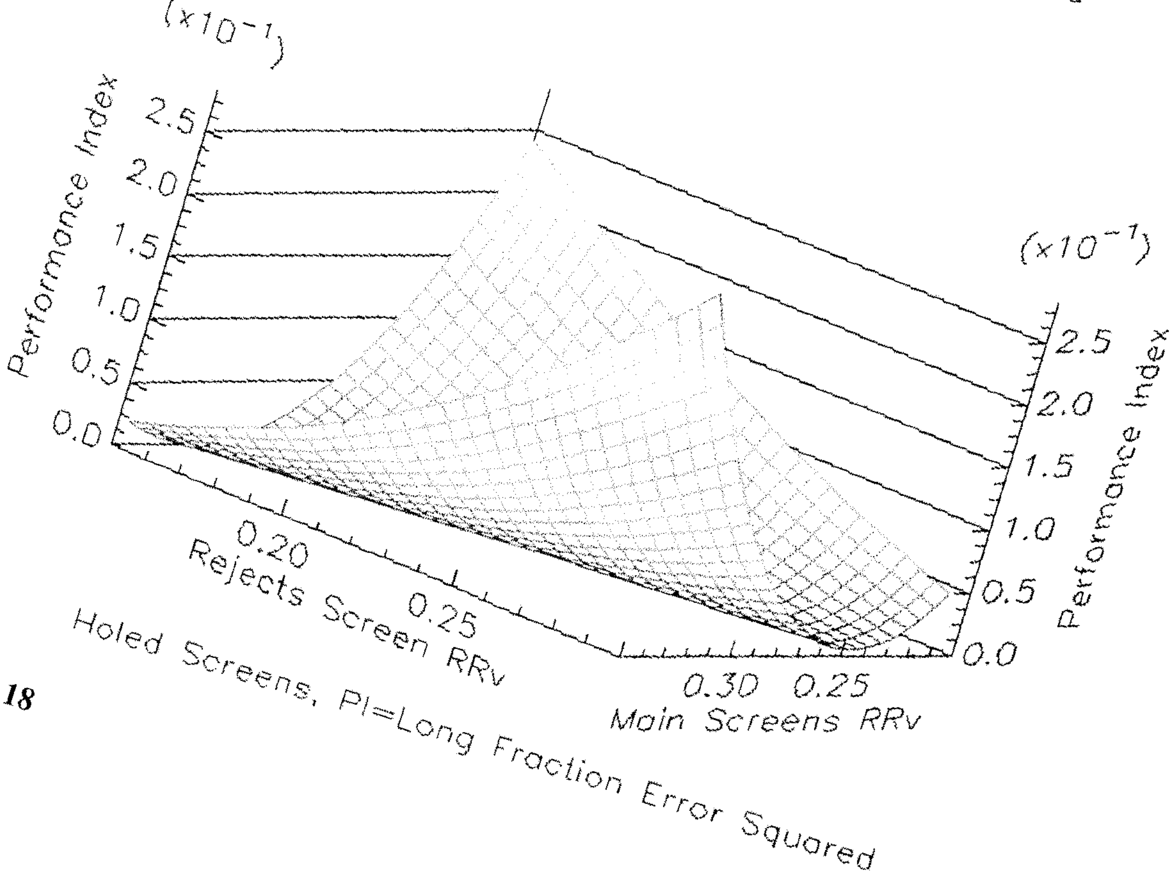
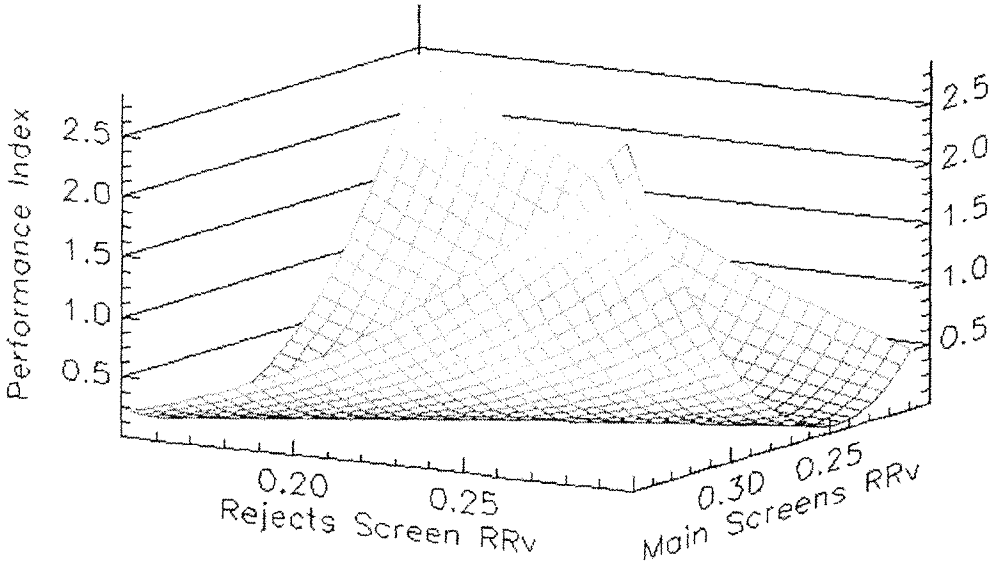
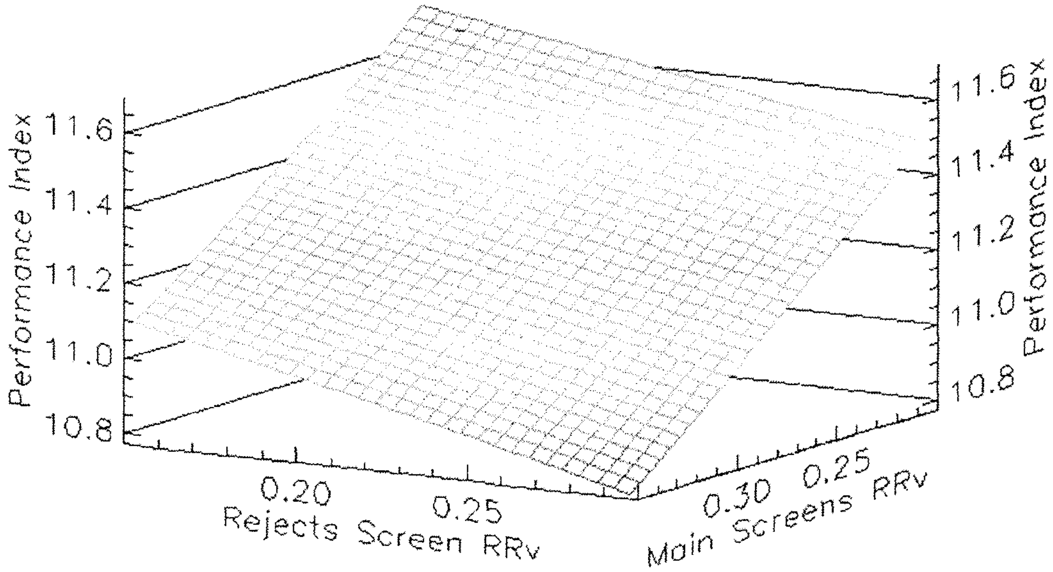


Figure 3.18



Holed Screens, PI=Medium+Long Fractions Error Squared

**Figure 3.19**



Slotted Screens, PI=Long Fraction Total Concentration Percentage

**Figure 3.20**

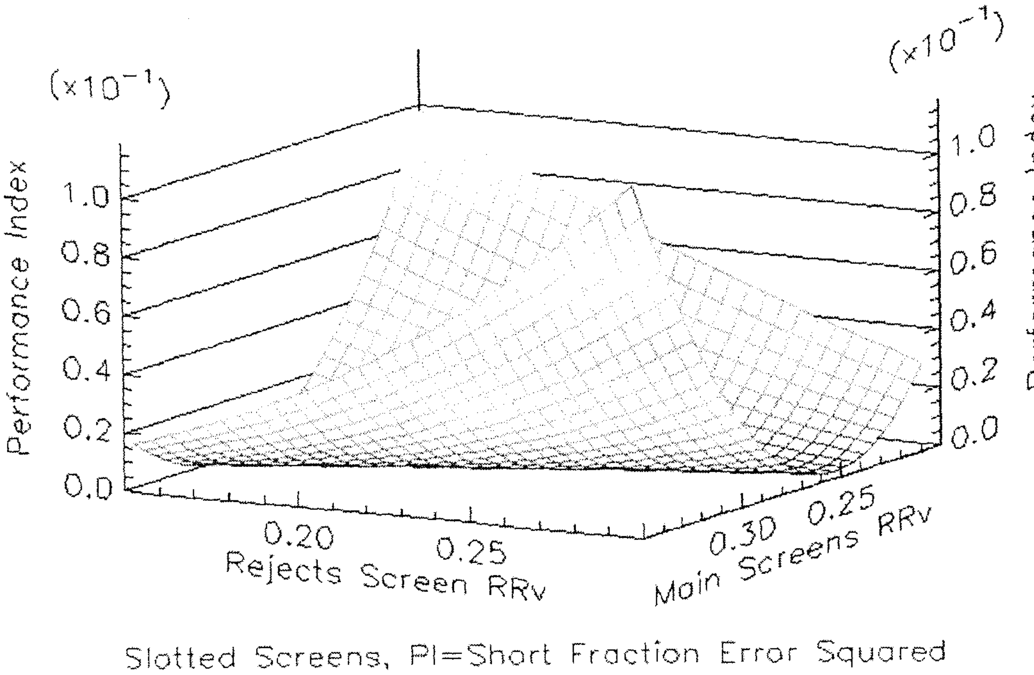


**Figure 3.21**

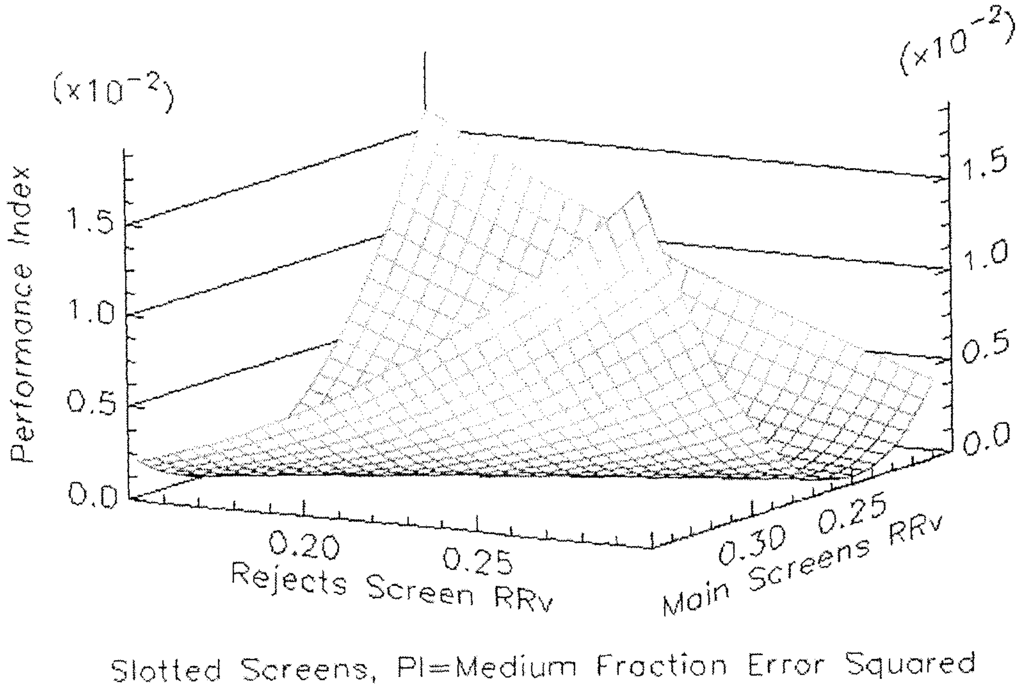


**Figure 3.22**

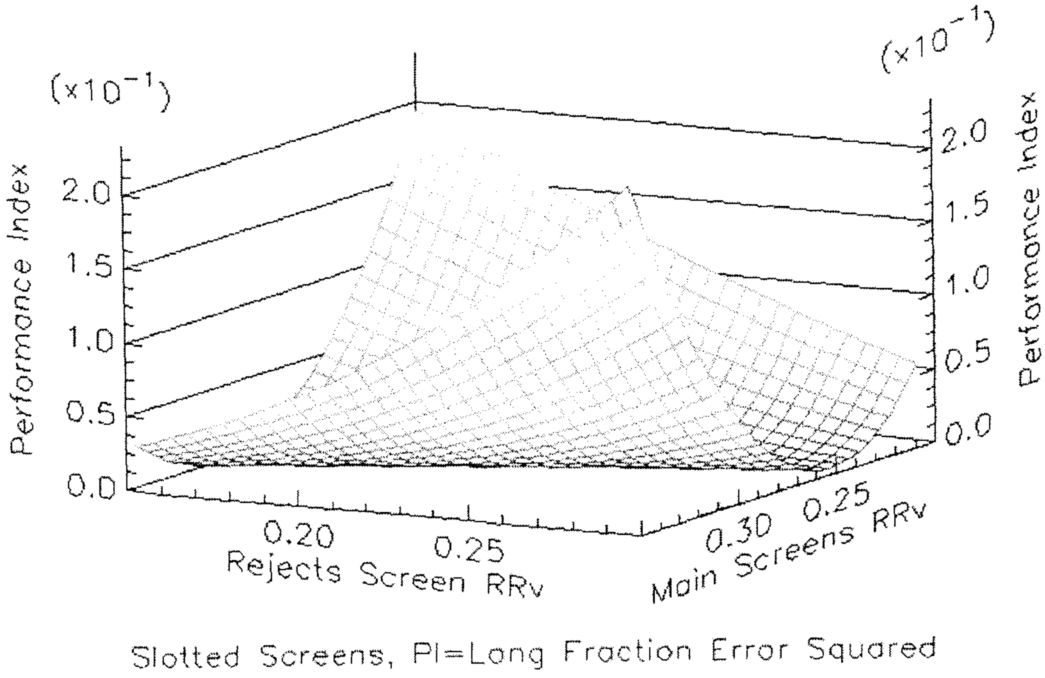


**Figure 3.23**

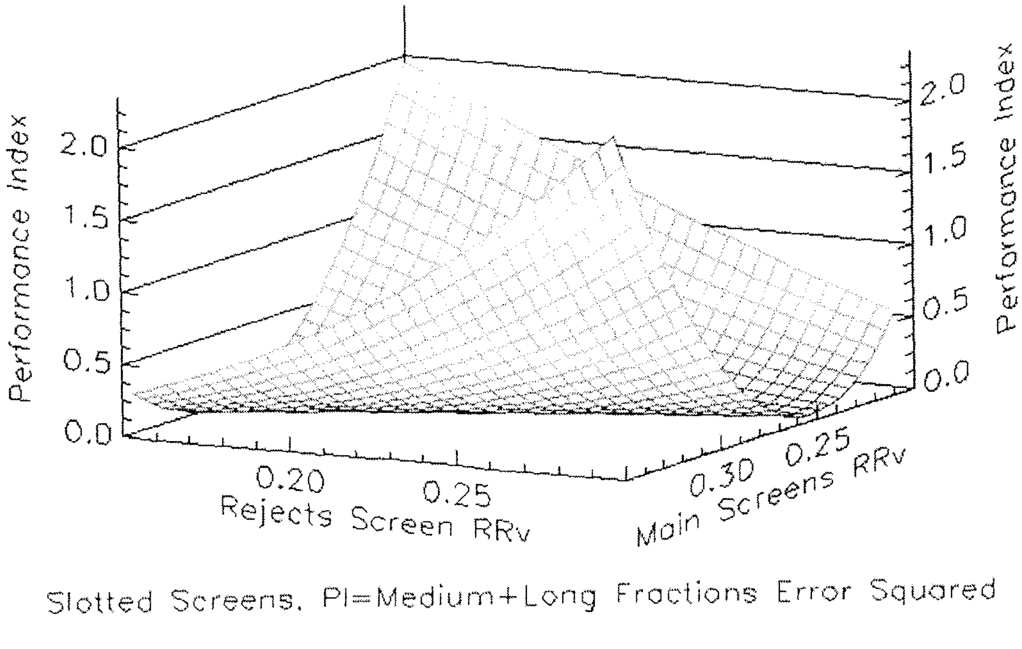


**Figure 3.24**

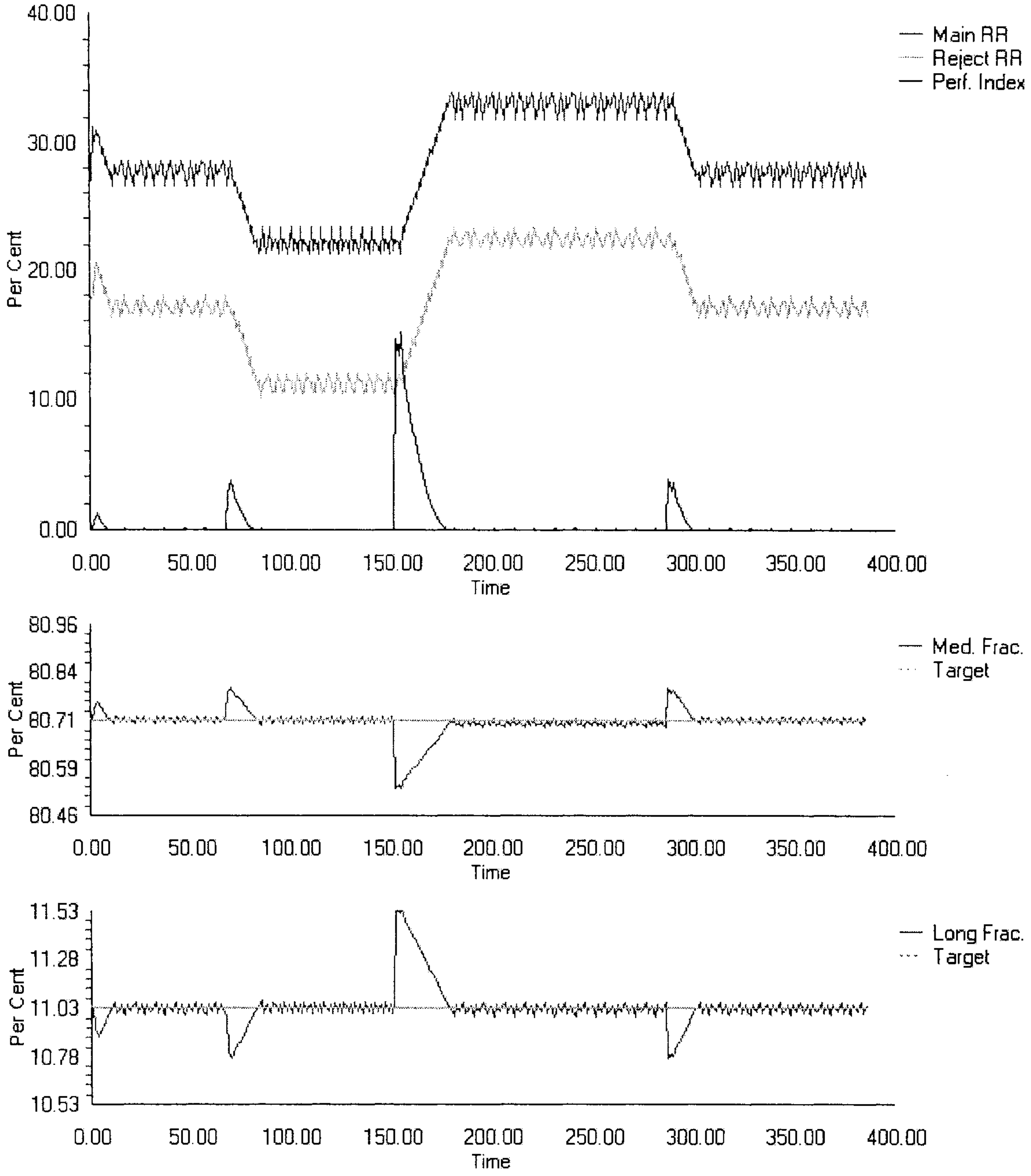


**Figure 3.25**

Adaptive Optimizer, Fibre Length Distribution Mean Step Changes, Holed Screens, No Simplex Contraction or Expansion, Refining Factor = 0.25

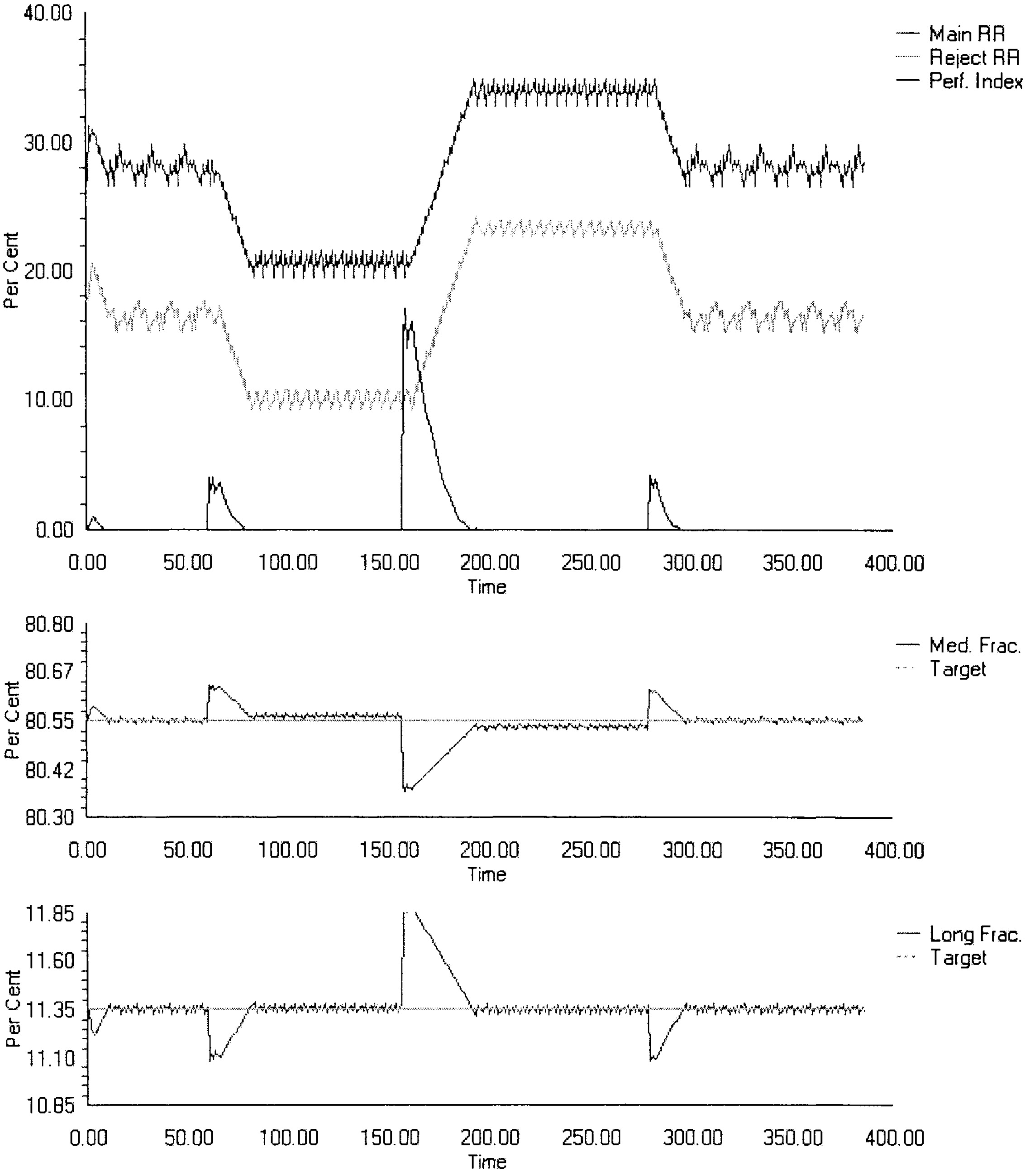


**Figure 3.26**

Adaptive Optimizer, Fibre Length Distribution Mean Step Changes, Slotted Screens, No Simplex Contraction or Expansion, Refining Factor = 0.25

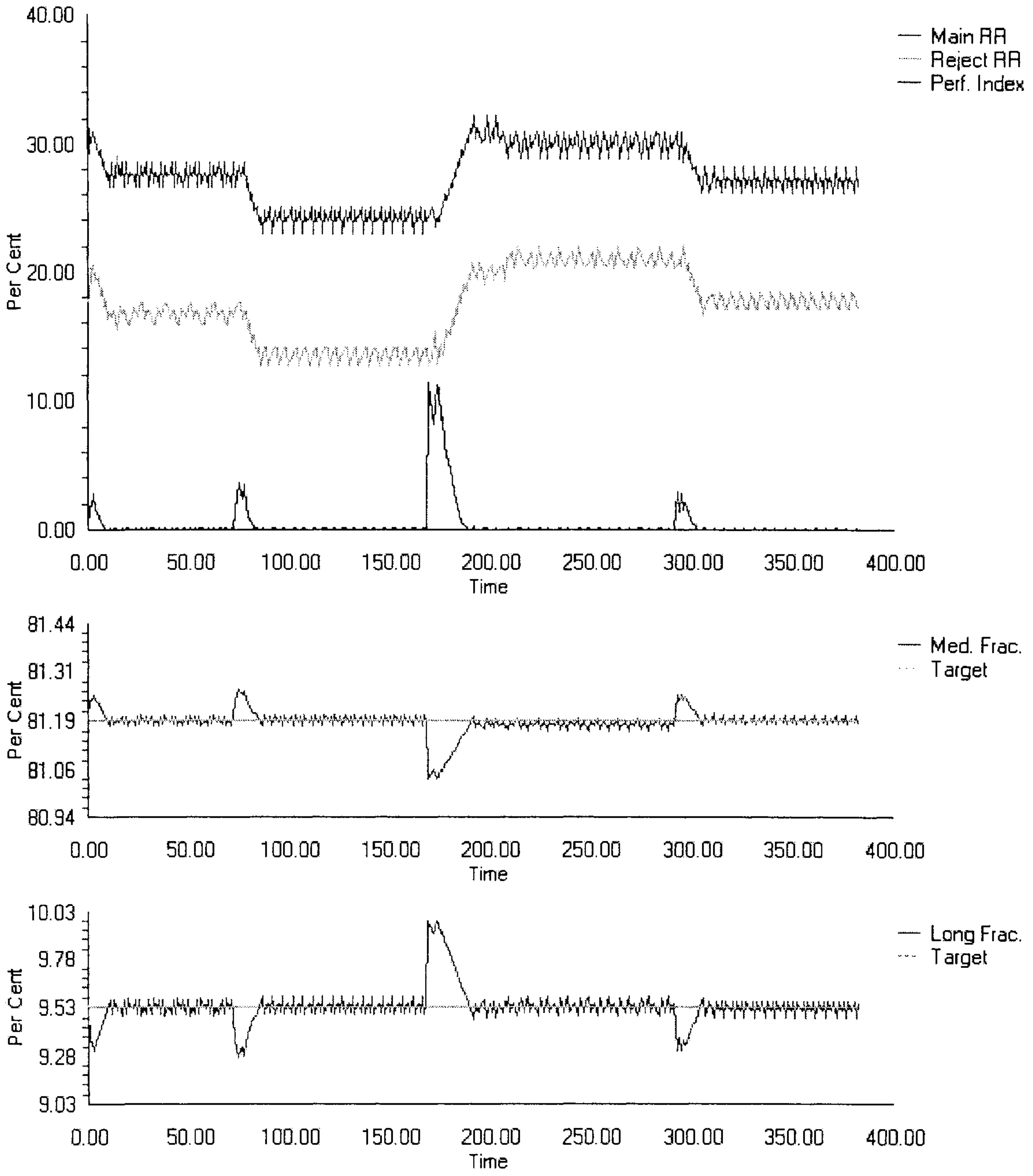


**Figure 3.27**

Adaptive Optimizer, Fibre Length Distribution Mean Step Changes, Holed Screens, No Simplex Contraction or Expansion, Refining Factor = 0.5

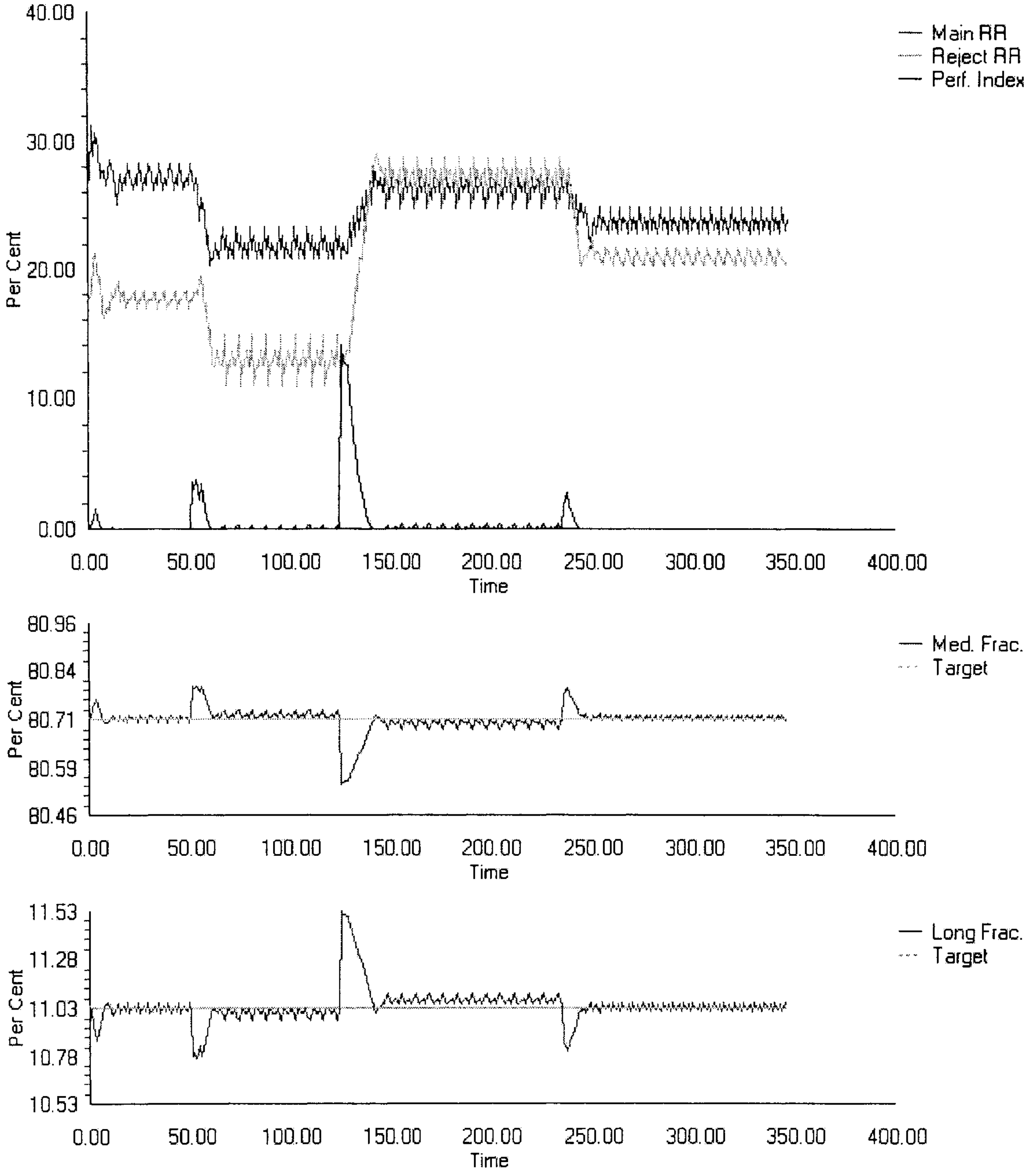


**Figure 3.28**

Adaptive Optimizer, Fibre Length Distribution Mean Step Changes, Holed Screens, Threshold Method of Simplex Contraction or Expansion, Refining Factor = 0.25

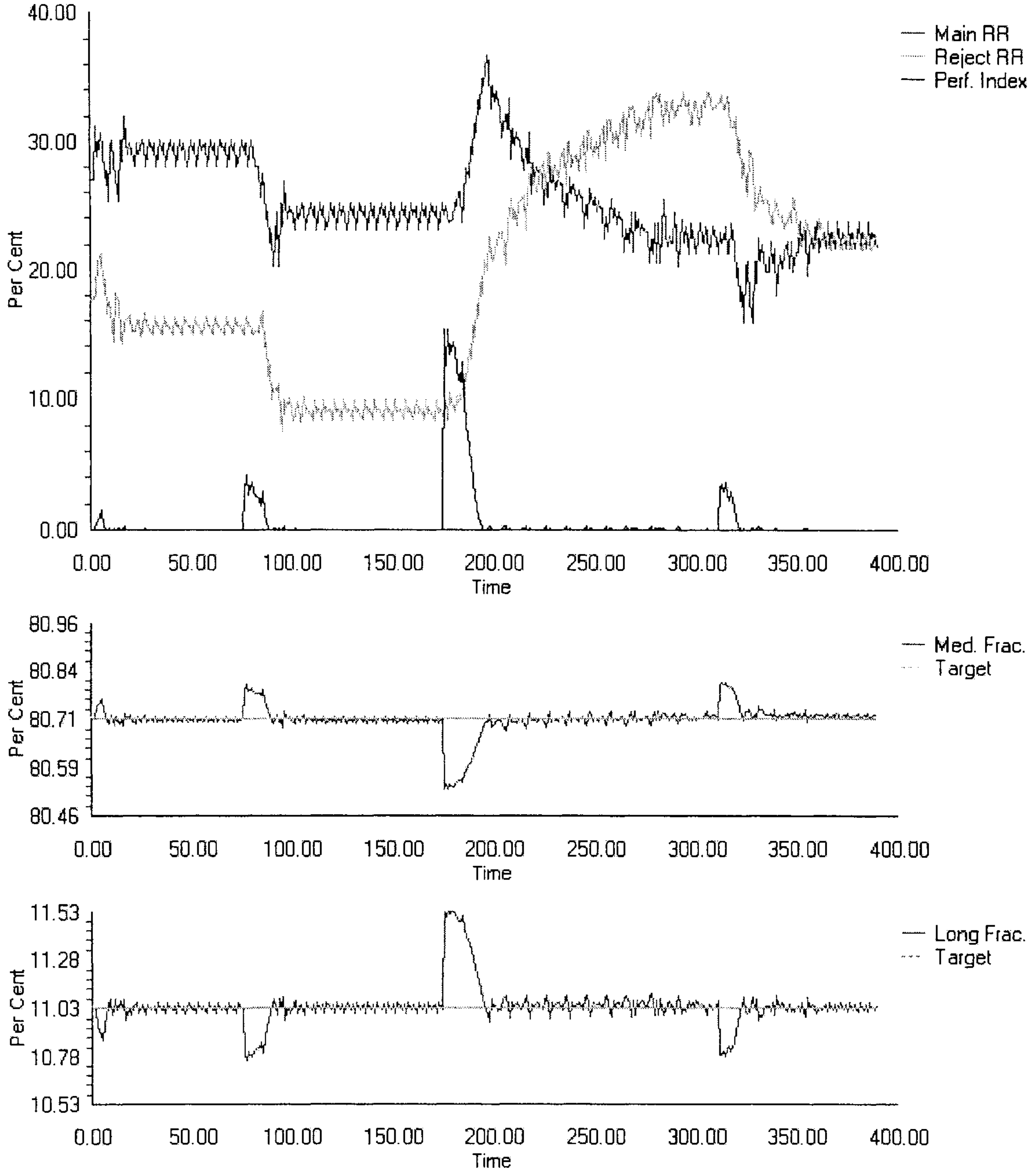


**Figure 3.29**

Adaptive Optimizer, Fibre Length Distribution Mean Step Changes, Holed Screens, Ranking Method of Simplex Contraction or Expansion, Refining Factor = 0.25

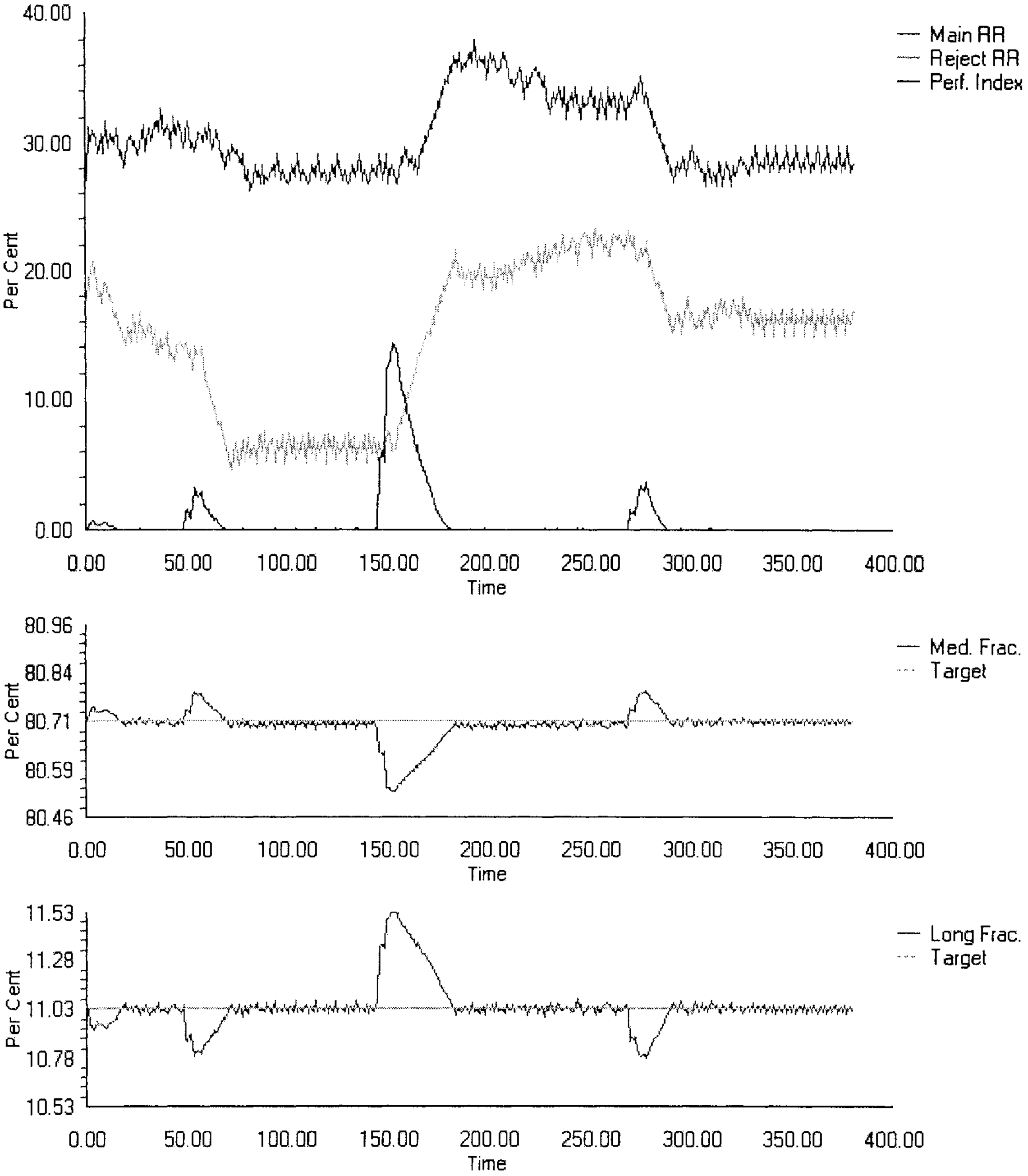


**Figure 3.30**

Adaptive Optimizer, Fibre Length Distribution Mean Step Changes, Holed Screens, No Simplex Contraction or Expansion, Refining Factor = 0.25, Rejects System Delay = 4

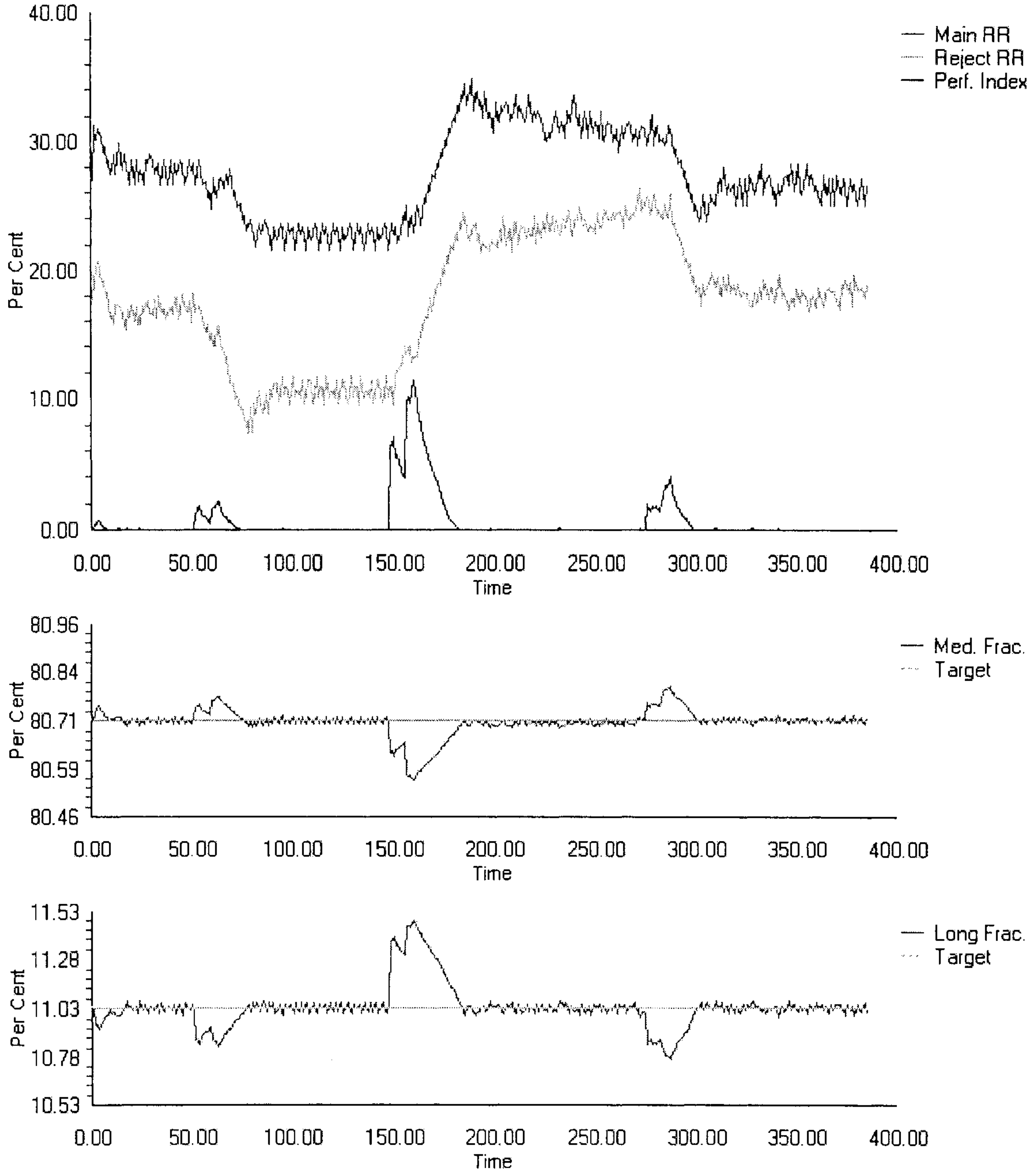


**Figure 3.31**

Adaptive Optimizer, Fibre Length Distribution Mean Step Changes, Holed Screens, No Simplex Contraction or Expansion, Refining Factor = 0.25, Rejects System Delay = 8

## Recommendations

1. Delay in the control loop presents a difficulty for the adaptive optimizer. It is recommended that further study be done in applying the adaptive optimizer to systems with dead time. As a starting point, the adaptive optimizer techniques could be tested with a discrete Smith Predictor control structure. While a model of the plant, including delay time, is required for the Smith Predictor, and this need of a model is contrary to the reasons for using the direct search method of the adaptive optimizer, insight into the possibility of dead time compensation for the adaptive optimizer may be gained. Another control structure, which may have promise in combination with the adaptive optimizer and system delay, is Model Predictive Control (MPC). Many different types of models are possible for calculating the predicted values of the process outputs with MPC, and discrete models can be used, where adaptive updating of the models is possible. Dead time compensation could be added to this discrete controller.

2. While the adaptive optimizer is able to control the pulp fractionation properties in the screening room simulations, the manipulated variables, i.e., the volumetric rejects ratios for the screens, have limited operational range. Studies on incoming variations in screening room feed flow fibre length distributions would be useful in determining if this method of control is able to counter typical feed flow fibre length distribution disturbances. Screen baskets with holed type apertures have more control range than screen baskets with slotted type apertures, and the former are preferred if pulp fractionation is a desired objective.

3. The effects of refining on pulp fibre length distributions is another facet of the screening room operation that is important to the control scheme outlined in this work. Further study in this area could help determine if refining could be used for fractionation control, and would also determine the role that the rejects system refining plays in the changes in fibre length distribution in the screening room.

4. The adaptive optimizer can be used as a controller in many other situations. It is recommended that a test trial of the adaptive optimizer be made with a physical process, such as a distillation column, in a laboratory environment, for the purpose of determining if the adaptive optimizer, with its inherent simplex reflection and re-measurement delays, can provide effective control with other processes and other objective functions. Control performance comparisons against traditional Proportional, Integral and Derivative (PID) controllers would also be important, in this regard.

5. Finally, some work in the area of optimization of the contraction and expansion methods for the adaptive optimizer would be useful. The adaptive optimizer always implements a regular simplex that only moves by reflection. Elongation of the simplex, or a complete translation of the simplex, may be effective in response to step changing disturbances or targets.

## References

[1] W. Spendley, G. R. Hext, and F. R. Himsworth, *Sequential application of simplex designs in optimization and evolutionary operation*, Technometrics, 4 (1962), pp. 441-461.

[2] P.E. Gill, W. Murray and M. Wright, *Practical Optimization*, Academic Press (1981), pp. 46-52

[3] P.E. Gill, W. Murray and M. Wright, *Practical Optimization*, Academic Press (1981), pp. 83-104

[4] P.E. Gill, W. Murray and M. Wright, *Practical Optimization*, Academic Press (1981), pp. 105-126

[5] R. Fletcher and C.M. Reeves, *Function minimization by conjugate gradients,* Computer Journal, 7 (1964), pp. 149-154

[7] R. Hooke and T. A. Jeeves, *Direct search solution of numerical and statistical problems*, Journal of the Association for Computing Machinery, 8 (1961), pp. 212-229.

[8] J. A. Nelder and R. Mead, *A simplex method for function minimization*, The Computer Journal, 7 (1965), pp. 308-313.

[9] K. I. M. McKinnon, *Convergence of the Nelder-Mead simplex method to a nonstationary point*, SIAM Journal on Optimization, 9 (1998), pp. 148-158.

[10] C.G. Broyden, *The convergence of a class of double-rank minimization algorithms,* J. Inst. Maths. Applics., 6 (1970), pp. 76-90

[11] R. Fletcher, *A new approach to variable metric algorithms,* Computer Journal, 13 (1970), pp. 392-399

[12] D. Goldfarb, *A family of variable metric methods derived by variational means,* Mathematics of Computation, 24 (1970), pp. 23-26

[13] D.F. Shanno, *Conditioning of quasi-Newton methods for function minimization,* Mathematics of Computation, 34 (1970), pp. 647-657

[14] Matlab - The Language of Technical Computing, *fminsearch*, Copyright © 1984, 2002, The Mathworks, Inc., Natick, Mass., USA

[15] *GNU Scientific Library (GSL)*, http://www.gnu.org/software/gsl, Copyright © 1989, 1991 Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

[16] *Numerical Recipes in C, The Art of Scientific Computing*, Cambridge University Press, (1992), Chapter 10

[17] A. Jutan and Q. Xiong, *Continuous Optimization using a Dynamic Simplex Method*, University of Western Ontario, (2000)

[18] J. Hennings, *On-line Optimization of Thermomechanical Pulping Rejects Screen using Modified Sequential Simplex Method,* Lakehead University, (2003)

[19] R.M. Lewis, V. Torczon and M.W. Trosset, *Direct Search Methods: Then and Now*, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, (2002)

[20] E. Bristol, *On a new measure of interaction for multivariable process control*, IEEE Transaction on Automatic Control, 11 (1966) , p. 133.

[21] R.W. Gooding and R.J. Kerekes, *Consistency changes caused by pulp screening,* Tappi Journal, Nov. (1992), p. 109

[22] J. Olson, B. Allison, N. Roberts and R. Gooding, *Fibre Length Fractionation Caused by Pulp Screening,* Journal of Pulp and Paper Science, 24.12 (1998), p. 395

[23] J. Olson, B. Allison and N. Roberts, *Fibre Length Fractionation Caused by Pulp Screening, Smooth-Hole Screen Plates,* Journal of Pulp and Paper Science, 26.1 (2000), pp. 12-16

[24] B. Allison and J. Olson, *Optimization of Multiple Screening Stages for Fibre Length Fractionation: Two-Stage Case*, Journal of Pulp and Paper Science, 26.3 (2000), p. 114

[25] J. Olson, *Fibre Length Fractionation Caused by Pulp Screening, Slotted Screen Plates*, Journal of Pulp and Paper Science, 27.8 (2001), pp. 255-261

[26] A. Ammala, *Fractionation of Thermomechanical Pulp in Pressure Screening,* University of Oulu, Finland, (2001)

[27] A.W. Hooper, *Screening of Mechanical Pulp – Effect of Control Parameters and Screen Operation on Pulp Quality*, Pulp and Paper Manufacture, Third Edition, Volume 2, pp. 181-194

[28] Hautala J, Hourula I, Jussila T & Pitkänen M *Screening and Cleaning. In Mechanical Pulping* (Ed. Sundholm J) Fapet Oy, Jyväskylä, Finland, (1999)