A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

Author: Yeh-Liang Hsu, Tze-Chi Liu, Tso-Lung Liu (2005-04-15); recommended: Yeh-Liang Hsu (2005-09-12)

Note: This paper is published in *Engineering Optimization*, Vol. 37, No. 6, September 2005, p. 649 – 661.

A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

Abstract

This paper proposes a fuzzy proportional-derivative (PD) controller optimization engine for engineering optimization problems using optimality criteria approach. Traditional numerical optimization algorithms treat optimization problems as pure mathematical problems. Engineering knowledge about the problem is not utilized in the optimization process. The idea of using the fuzzy PD controller in engineering optimization is that, instead of using purely numerical information to obtain the new design point in the next iteration, engineering knowledge and human supervision process can be modeled in the optimization algorithm using fuzzy rules. The fuzzy PD controller optimization engine developed in this work appears to have stable performance in both structural optimization and blow molding parameter optimization examples.

*Keywords*: Engineering optimization; PD control; Fuzzy control; Optimality criteria
1. Introduction

Optimality criteria methods are based on the derivation of appropriate optimality criteria for specialized design (Rao 1996). Iterative numerical algorithms are then developed to find the design that satisfies the optimality criteria. For example, one of the simplest approaches for minimizing the weight of a statically loaded structure subject to stress constraints is the fully stressed design (FSD) technique. Combining with finite element analysis to calculate stress, the objective function of the fully stressed design is often in the following form:

\[
f = \text{min} \left( \frac{\sum_{j=1}^{n} (y_j - Y_j)^2}{n-1} \right)^{0.5}
\]

where \( y_j \) is the stress calculated at finite element node \( j \) and \( Y_j \) is the corresponding target stress at this node. The optimality criterion in the form of equation (1), to minimize the differences between certain system output and the target values, is also very common in many engineering optimization problems.

Most real-world engineering design problems exhibit complex phenomena and are often difficult to be expressed in a well-defined mathematical optimization model, which hinders the use of formal numerical optimization techniques to solve these problems. Moreover, design modifications in real-world engineering design problems are often based on engineering heuristics, experiences and knowledge. On the contrary, when solving an engineering optimization problem using numerical optimization techniques, the engineering problem is basically viewed as a pure mathematical optimization model. Design modifications in the optimization process rely purely on numerical information. Engineering heuristics are totally ignored in the numerical optimization algorithms.

This paper presents an optimization engine based on the concept of ‘fuzzy proportional-derivative (PD) controller’, which enables the use of engineering heuristics to generate the new design point of the next iteration in the optimization process. The
structure of an optimization algorithm is maintained to guide the engineering decision process and to ensure that an optimal solution rather than a trial and error solution can be obtained. Currently, this optimization engine is specifically developed for solving the problems with optimality criteria in the form of equation (1).

Arakawa and Yamakawa (1990) demonstrated an optimization method using qualitative reasoning, which makes use of the qualitative information giving an approximate direction of the optimum search. Hsu et al. (1995a, b) proposed a fuzzy optimization algorithm and applied it for determining the move limit, which is an important optimization process parameter in the sequential linear programming algorithm. Mulkay and Rao (1998) proposed a modified sequential linear programming algorithm using fuzzy heuristics to control the optimization parameters. Arabshahi et al. (1996) pointed out that many optimization techniques involve parameters that are often adapted by the user through trial and error, experience, and other insight. Instead, they applied neural and fuzzy ideas to adaptively select these parameters.

This paper first explains how to apply the concept of fuzzy PD controllers to optimization algorithms. Then a structural optimization example is used to demonstrate the optimization process using the fuzzy PD controller optimization engine. A blow moulding manufacturing parameter optimization problem is also solved by the fuzzy PD controller optimization engine. The fuzzy PD controller optimization engine appears to have stable performance in both the structural optimization and the blow molding parameter optimization examples.

2. The fuzzy PD controller optimization engine

Fuzzy theory is primarily concerned with quantifying and reasoning using natural language in which many words have ambiguous meanings. As the introduction of the basic theory of fuzzy sets by Zadeh (1965), fuzzy theory has been extended and applied to many different fields. In the past two decades, the field of fuzzy controller applications has broadened to include many industrial control applications.
A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

Fuzzy control is similar to the classic closed-loop control approaches but differs in that it substitutes imprecise, symbolic notions for precise numeric measures. The fuzzy controller takes input values from the real world. These crisp input values are mapped to the linguistic values through the membership functions in the fuzzification step. A set of rules that emulates the decision-making process of the human expert controlling the system is then applied using certain inference mechanisms to determine the output. Finally the output is mapped into crisp control actions required in practical applications in the defuzzification step.

In a close-loop control system shown in figure 1, the measured response of the system process being controlled is fed back to be compared with a desired response. The control actions generated by the controller are determined in part by the system response in an attempt to fix this error. The output of a proportional controller is a control signal $u$ as shown in equation (2). The control signal is proportional to the error and $K_p$ is the gain. A proportional controller will have the effect of reducing the rise time and will reduce, but never eliminate, the steady state error.

$$u = K_p e$$  \hspace{1cm} (2)

![Figure 1. A close-loop control system](http://designer.mech.yzu.edu.tw/)

Derivative control is used to anticipate the future behavior of the error signal by using corrective actions based on the rate of change in the error signal. The output of a derivative controller is a control signal $u$ as in equation (3). The control signal is proportional to the derivative of the error and $K_d$ is the gain. Derivative control will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient
A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

response. A disadvantage of derivative control is that the controller output is zero when the error signal is constant.

\[ u = K_d \frac{de}{dt} \]  

(3)

The control action in a PD controller is in the form of equation (4), which combines proportional and derivative control modes. The PD controller makes a control loop respond faster and with less overshoot, and is the most popular method of control by a great margin.

\[ u = K_p e + K_d \frac{de}{dt} \]  

(4)

The fuzzy counterpart of the PD controller also has two inputs: system error \( e \) and error change \( \Delta e \). Fuzzy inference is used to compute the control signal \( u \). Table 1 shows a typical rule base for a fuzzy PD controller with 25 rules. Five linguistic terms are used for each variable, negative big (NB), negative small (NS), zero (ZE), positive small (PS), and positive big (PB).

<table>
<thead>
<tr>
<th>( \Delta e )</th>
<th>( e )</th>
<th>NB</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NB</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
<td>NS</td>
<td></td>
</tr>
<tr>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>NS</td>
<td></td>
</tr>
<tr>
<td>ZE</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>NB</td>
<td></td>
</tr>
<tr>
<td>PS</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>PB</td>
<td>NB</td>
<td></td>
</tr>
<tr>
<td>PB</td>
<td>PS</td>
<td>PS</td>
<td>PB</td>
<td>PB</td>
<td>NB</td>
<td></td>
</tr>
</tbody>
</table>

The iterative optimization process can be analogous to a close-loop control system. Figure 2 shows a general block diagram for an optimization process. Compared with figure 1, an optimization model in an optimization process is analogous to the system process in a control system; and an optimization algorithm is analogous to the controller. Initial
parameters are input to the optimization algorithm, which in turn generates a trial design point according to its search rules. The optimization model is then evaluated at this trial design point, and the information such as objective and constraint function values and sensitivity are feedback to the termination test. If the termination test fails, the optimization algorithm is triggered again to generate the next design point, using the numerical information from previous iterations. Finally, although a control system attempts to achieve a stable, predefined output, the optimization process pursues a converging objective function value.

![Figure 2. A general block diagram for an optimization process](image)

Traditional numerical optimization algorithms are analogous to direct digital controllers. The algorithms are usually crisply designed for well-defined mathematical models. Their numerical rules for generating the next design point are exact and definite, and they can usually be proved to have nice converging behavior when applied to well-defined mathematical models. However, in engineering optimization problems, we seldom have well-defined mathematical models. The objective and constraint functions often do not have exact algebraic forms in terms of the design variables, and they can only be evaluated through experiments or computer simulations, which are expensive and imprecise in nature. Usually, sensitivity calculation is also done through imprecise finite difference methods. Very often the cost of the number of function evaluations required to meet the crisp definition of the numerical algorithms is too high to be affordable.

On the other hand, when numerical optimization algorithms are applied to an engineering problem, the engineering problem is treated as a pure mathematical problem. Engineering heuristics are totally ignored in the numerical optimization algorithms. This
motivates the idea that, in addition to crisp numerical rules, the human supervision process should also be modeled in an optimization algorithm using fuzzy rules; the “controllers” in the optimization process may as well be fuzzy controllers!

Figure 3 shows the block diagram of using the fuzzy PD controller as the optimization engine in an optimization process. Currently, the fuzzy PD controller optimization engine minimizes the differences between system output and the target values, that is, it only handles the specific type of objective function in equation (1). Comparing the initial system output $y$ with the target values, or the set point $Y$, initial error $e$ and change of error $\Delta e$ are input to the fuzzy PD controller, which generates the change of design variable $\Delta x$ for the next iteration. Then, the system input is updated ($x^{q+1} = x^q + \Delta x^q$) and the new system process output $y^{q+1}$ is fed back to compare with the set point $Y$ again.

![Block Diagram](http://designer.mech.yzu.edu.tw/)

Figure 3. Using the fuzzy PD controller in an optimization process

System output $y$ are functions of design variables $x$. In order to correctly update the design variables in the next iteration, the relation between $\Delta x$ and $y$ should be known empirically and modeled in the fuzzy PD controller optimization engine. The optimization process terminates when $e$ and $\Delta e$ approach zero, that is, when no change in design variables $\Delta x$ will be generated by the fuzzy PD controller optimization engine.

In the following section, a structural optimization example is used to demonstrate the optimization process using the fuzzy PD controller optimization engine.

3. A structural optimization example

In axisymmetric shell structures, the discontinuity at the intersection of two shells causes stress concentration, which often causes failure of the shell structure. Thus how to
A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

reinforce the discontinuities to minimize stress concentration while keeping the weight of the shell structure at the lowest possible value is an important task in designing shell structures. It is, therefore, an optimization problem on how to obtain a fully stressed thickness profile that keeps the stress at the discontinuities at the nominal stress value. That is, to minimize the deviation of the stress from the target nominal stress.

Figure 4 shows the loading and boundary conditions of a shell structure with a cylinder-cone intersection. Simply supported boundary conditions are applied at the end of the shell and an internal pressure load of 0.2 MPa is applied. The material is steel with an initial uniform thickness of 10 mm. The shell structure can be discretized using finite elements. The thickness at 35 control points indicated by the arrows in Figure 4 are manipulated to obtain fully stressed thickness profile of 10 MPa. This problem was studied by Hsu et al. [8] using another optimization method to find the constant stress boundary.

Figure 4. Loading and boundary conditions of the cylinder-cone example
Given a set of thicknesses at the nodes, it is possible to obtain the stress at all nodes from the simulation results by finite element software. Therefore in the objective function equation (1), \( n \) is the total number of nodes, \( y_j \) is the stress at the \( j \)th node in the finite element model, and \( Y_j \) is the corresponding target stress, which is constant for all nodes. The thicknesses at discrete nodes are the design variables \( x_j \). Obviously, at the \( j \)th node, the stress \( y_i \) is a monotonic function of thickness \( x_j \), that is, stress reduces if the thickness increases.

This type of monotonicity prevails in engineering design problems. In a fuzzy system, the engineering heuristics for modifying the thickness can be expressed in linguistic statements in the form of simple IF-THEN rules:

Rule 1: ‘IF stress is high, THEN increase thickness.’

Rule 2: ‘IF stress is low, THEN reduce thickness.’

This set of rules indicates that if \( e_j = y_j - Y \) (the difference between the stress at a certain node and the target stress) is positive, then \( \Delta x_j \) (change in thickness in corresponding node) should be positive and vice versa.

On the other hand, \( \Delta e_j \) reflects the ‘trend’ of system output \( y_j \). In the \( q \)th iteration, the change in error at the \( j \)th node is defined as

\[
\Delta e_j^q = (y_j^q - Y) - (y_j^{q-1} - Y)
\]  

(5)

In the case of an axisymmetric shell, a positive \( \Delta e_j \) reflects that the stress is increasing, and a negative \( \Delta e_j \) reflects that the stress is decreasing at the \( j \)th node. From the optimization point of view, Rule 1 and Rules 2 can be extended into the following heuristics for modifying the design variables:
A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

Rule 1(a): ‘IF stress is high AND stress is decreasing, THEN increase thickness softly.’

Rule 1(b): ‘IF stress is high AND stress is increasing, THEN increase thickness strongly.’

Rule 2(a): ‘IF stress is low AND stress is increasing, THEN decrease thickness softly.’

Rule 2(b): ‘IF stress is low AND stress is decreasing, THEN decrease thickness strongly.’

Rules 1(a) and 2(a) try to prevent overshoot, while Rules 1(b) and 2(b) try to increase the convergence rate. These engineering heuristics are further interpreted into the fuzzy rule base with 25 rules shown in Table 2. The linguistic terms NB, NS, ZE, PS, and PB are to be defined in individual cases.

<table>
<thead>
<tr>
<th>Δe</th>
<th>e</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PB</td>
<td>PS</td>
<td>ZE</td>
<td>NS</td>
<td>NB</td>
</tr>
<tr>
<td>PB</td>
<td>PB</td>
<td>PB</td>
<td>ZE</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>PS</td>
<td>PB</td>
<td>PS</td>
<td>ZE</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
<td>ZE</td>
<td>NS</td>
<td>NB</td>
</tr>
<tr>
<td>NS</td>
<td>PS</td>
<td>PS</td>
<td>ZE</td>
<td>NS</td>
<td>NB</td>
</tr>
<tr>
<td>NB</td>
<td>PS</td>
<td>PS</td>
<td>ZE</td>
<td>NB</td>
<td>NB</td>
</tr>
</tbody>
</table>

The fuzzy PD controller optimization engine is used to find a constant stress profile for this shell structure. In the objective function equation (1), $Y=10$ and $n=35$. The quantization table table 3 gives quantitative definitions for PB, PS, ZE, NS and NB for the two inputs $e$ and $\Delta e$, and the output $\Delta x$ for this example. Moreover, in this example, the maximum allowable thickness is 50 mm, and the minimum allowable thickness is 5 mm.
A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

Table 3. The quantization table

<table>
<thead>
<tr>
<th>Quantized Level</th>
<th>$e$</th>
<th>$\Delta e$</th>
<th>$\Delta x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Target stress</td>
<td>Target stress/2</td>
<td>50% of initial thickness</td>
</tr>
<tr>
<td>1</td>
<td>Target stress/2</td>
<td>Target stress/4</td>
<td>50% of initial thickness/2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>-Target stress/2</td>
<td>-Target stress/4</td>
<td>-50% of initial thickness/2</td>
</tr>
<tr>
<td>-2</td>
<td>-Target stress</td>
<td>-Target stress/2</td>
<td>-50% of initial thickness</td>
</tr>
</tbody>
</table>

The values of quantitative definitions for $e$, $\Delta e$, and $\Delta x$ in table 3 reflect the designer’s subjective perception of ‘high stress’, ‘low stress’, and so on. These values will affect the sensitivity of the algorithm. If the algorithm is too sensitive that a small error in stress triggers a large change in thickness, the algorithm will become unstable. On the other hand, if a large error in stress only triggers a small change in thickness, the algorithm will be slow. In table 3, the target stress is used as the reference for ‘high stress’ and ‘low stress’. The maximum change in thickness in one iteration is set at 50% of initial thickness.

As shown in figure 5, the fuzzy PD controller terminates after 42 iterations. The termination criterion is that the iteration stops when the difference of objective function values in consecutive iterations is < 1%. Figure 6 shows the initial and final thickness distributions. Figure 7 compares the stress distributions along the shell in the initial and final designs. There is a serious stress concentration (stress concentration factor larger than 8) at the neighborhood of the cylinder-cone intersection in the initial design, and the stress appears to be constant at target stress in the final design. Note that only 42 analyses are required to obtain this result and the sensitivity information is not needed. The fuzzy PD controller code is completely external to the analysis code.
A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

Figure 5. Iteration history of the cylinder-cone example

Figure 6. Initial and final thickness distributions of the cylinder-cone example
4. A blow moulding process parameter optimization example

Blow moulding is the forming of a hollow part by ‘blowing’ a mould-cavity-shaped parison that is made by thermoplastic molten tube. It is the most popular and efficient process for manufacturing commodity hollow plastic parts such as bottles, containers, and toys. More recently, this forming process has been applied to the manufacture of complex automotive parts such as fuel tanks, seat backs, air ducts, windshield washers and cooling reservoirs.

The blow moulding process consists of three phases: parison extrusion, parison inflation and part solidification. The extrusion phase involves the extrusion of a polymer melt through an annular die to form a hollow cylindrical parison with a non-uniform material distribution and consequently non-uniform parison thickness along its length. Once the parison is extruded to the desired length, it is inflated to take the shape of an enclosing mould. The part then solidifies as a consequence of heat transfer to the cooling mould. The parison thickness distribution is modified significantly by the inflation and the solidification stages to yield the final part thickness distribution.

Figure 7. Initial and final stress distributions of the cylinder-cone example
Blow moulded parts often require a strict control of the thickness distribution in order to achieve the required mechanical performance and final weight. For example, in industrial applications, thickness of the part is often required to be larger than a specified value. Therefore a uniform thickness distribution at this specified value is the minimum weight design. Manipulation of the die gap programming points can lead to an optimal part thickness distribution (Diraddo and Garcia-Rejon 1993, Lee and Soh 1996). Figure 8 shows the forming of an axisymmetric bottle. As illustrated in figure 8(a), the die gap can be adjusted as a function of time in order to obtain the desired thickness profile along the extruded parison, which determines the thickness of the blown hollow part. For example, in order to obtain a uniform thickness distribution for the hollow part, the thickness of programmed parison must be non-uniform. As shown in Figure 8(b), the parison thickness for the largest expansion area must be thicker than that of the other areas.
The blow moulding process can also be simulated using the finite element type of software. BlowSim developed by the Industrial Materials Institute (IMI) of the National Research Council (NRC), Canada is used in this research (Laroche et al. 1996, 1999). In the blow moulding process, it is desirable to manipulate the die gap programming to minimize the material of the final part, whereas the thickness of the final part should be larger than a prescribed thickness. The optimality criterion is therefore to obtain a final part of constant thickness.

Figure 9 shows the geometry of a bottle. In this example, the die gap openings at seven evenly distributed discrete programming points are to be manipulated to obtain a uniform part thickness of 2 mm. It is an optimization problem on how to control the die gap openings in order to minimize the deviation in the thickness of the final part from the target thickness. The blow moulding process simulation software provides the ‘average weighted thickness’ of all nodes affected by the die gap opening of a certain programming point. Given a set of die gap openings, it is possible to extract the average weighted thickness of all programming points from the simulation results.
A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

Therefore the objective function of this problem is again in the form of equation (1). Here $n$ is still the total number of programming points (7), $y_j$ is the average weighted thickness of all nodes affected by the die gap opening of the $j$th programming point, and $Y$ is the target thickness. The die gap openings at discrete programming points are the design variables $x_j$. Obviously, at the $j$th node, thickness $y_j$ is a monotonic function of die gap opening $x_j$, that is, thickness increases if the die gap opening increases. This engineering heuristics for modifying the gap openings can be expressed in linguistic statements in the form of simple IF-THEN rules:

Rule 1: ‘IF thickness is large, THEN reduce the die gap opening.’

Rule 2: ‘IF thickness is small, THEN increase the die gap opening.’

In the blow moulding process, a positive $\Delta e$ reflects that the thickness is increasing, and a negative $\Delta e$ reflects that the thickness is decreasing. Considering $\Delta e$, Rule 1 and Rule 2 can be extended into:

Rule 1(a): ‘IF thickness is large AND thickness is decreasing THEN reduce die gap opening softly.’

Rule 1(b): ‘IF thickness is large AND thickness is increasing THEN reduce die gap opening strongly.’

Rule 2(a): ‘IF thickness is small AND thickness is decreasing THEN reduce die gap opening strongly.’

Rule 2(b): ‘IF thickness is small AND thickness is increasing THEN reduce die gap opening softly.’

These rules can then be interpreted into the fuzzy rule base of blow moulding process with 25 rules as shown in Table 4.
A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

Table 4. Fuzzy rules of blow moulding process

<table>
<thead>
<tr>
<th>$\Delta e$</th>
<th>PB</th>
<th>PS</th>
<th>ZE</th>
<th>NS</th>
<th>NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB</td>
<td>NB</td>
<td>NB</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>PS</td>
<td>NB</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>ZE</td>
<td>NB</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PB</td>
</tr>
<tr>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PB</td>
</tr>
<tr>
<td>NB</td>
<td>NS</td>
<td>NS</td>
<td>ZE</td>
<td>PB</td>
<td>PB</td>
</tr>
</tbody>
</table>

Figure 9. Geometry of the bottle

The fuzzy PD controller optimization engine will then generate a set of changes in the die gap openings $\Delta x_i$ for the next iteration according to the current average weighted thickness and the quantization table defined by the user as in Table 5. The initial die gap openings are 50%, the maximum die gap opening is 95%, and the minimum die gap opening is 5%. The optimization process terminates when the change in objective function value in consecutive iterations is $< 0.1\%$.

Table 5. The quantization table

| Quantized Level | $e$ | $\Delta e$ | $\Delta x$ |
A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

<table>
<thead>
<tr>
<th>PB</th>
<th>Target Thickness</th>
<th>Target Thickness</th>
<th>25% of die gap opening</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS</td>
<td>Target Thickness/2</td>
<td>Target Thickness/2</td>
<td>25% of die gap opening/2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NS</td>
<td>-(Target Thickness)/2</td>
<td>-(Target Thickness)/2</td>
<td>-(25% of die gap opening)/2</td>
</tr>
<tr>
<td>NB</td>
<td>-(Target Thickness)</td>
<td>-(Target Thickness)</td>
<td>-(25% of die gap opening)</td>
</tr>
</tbody>
</table>

The bottle example in Figure 9 terminated after 16 iterations. Figure 10 shows the iteration history of the objective function value. The initial objective function value is 0.8226, and the final objective function value is 0.4586. Ideally the objective function should converge to zero upon obtaining a part with uniform thickness. Figure 11 compares the average weighted thickness of the initial and final design at the seven programming points. Figure 12 shows the initial and final die gap openings. As shown in figure 12, the die gap openings of control points 1, 6, and 7 have arrived the lower bound 5%, whereas their corresponding average weighted thickness are still higher than the target value.

Figure 10. Iteration history of the blow moulding example
A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach

Figure 11. Initial and final average weighted thickness

Figure 12. Initial and final die gap openings
5. Conclusions

This paper has proposed a fuzzy PD controller optimization engine for engineering optimization problems using an optimality criteria approach. The optimality criterion in the form of equation (1) is very common in engineering optimization problems. This paper shows how the engineering knowledge and heuristics, such as the monotonicity between design variables and certain system output, can be modeled in the optimization algorithm using the fuzzy PD controller to obtain the new design point in the next iteration.

The monotonicity between design variables and objective and constraint functions prevail in engineering design optimization problems. The designer can identify which constraints must be active at the optimum design point using monotonicity analysis (Papalambros and Wilde 2000). In the fortunate ‘constraint-bound’ case, the number of design variables equals to the number of active constraints. Therefore, the optimization problem becomes finding the design point that satisfies all active constraints with strict equality. An optimality criteria in the form of equation (1) can be formed.

The fuzzy PD controller optimization engine developed in this work appears to have stable performance in both the structural optimization and blow molding parameter optimization examples. In both examples no interactions was assumed between design variables. That is, one active constraint contains only one design variable and one design variable appears in only one active constraint. Therefore, the updating of a design variable is decided by only one fuzzy rule. The full potential of the fuzzy PD controller optimization engine is not realized yet. For more general engineering design optimization problems, future research should develop algorithms using the fuzzy PD controller optimization engine for design variables with strong interactions.
6. References


A fuzzy proportional-derivative controller for engineering optimization problems using an optimality criteria approach


