A Constrained Genetic Algorithm for Decentralized Control System Structure Selection and Optimization

D. R. Lewin* and A. Parag

PSE Research Group, Dept of Chemical Engineering, Technion, Haifa 32000, ISRAEL

* Corresponding author. Phone +972 4 8292006 Fax: +972 4 8295672 E-mail: dlewin@tx.technion.ac.il

Abstract

A genetic algorithm is applied to the automatic synthesis of decentralized MIMO control systems incorporating both feedback and feedforward components. The design approach relies on a robust performance objective formulation, which defines the limiting trade-off between performances of the output channels. The method is demonstrated on the synthesis of typical MIMO control systems.

Keywords: MIMO control synthesis; Decentralized control; Feedforward control; Robust control design; Constrained optimization; Genetic algorithms.

1. INTRODUCTION

This paper concerns the automated synthesis of regulatory and servo control for uncertain multivariable processes. Decentralized control, in which each manipulated variable is paired with a process output, is the most commonly implemented multivariable strategy in the process industries, because of its simplicity and inherent robustness to sensor and actuator failures. Several methods for decentralized control system synthesis have been proposed. Hovd and Skogestad (1994) present an iterative procedure for decentralized control system design, starting with the fastest loops and working out to those that respond slowest, in a similar fashion to the design approach for cascade controllers. Direct approaches include those of Mizumoto at al. (1999), Chen and Seborg (2001), and Van Antwerp et al. (2001).

All of the above tune the decentralized control system either for fixed performance, or according to frequency domain performance specifications to be defined a priori by the designer, even in cases involving model uncertainty. However, even if a design thus executed meets the given performance specification, there is no indication of whether improved performance could have been achieved. For uncertain multivariable systems, it is of interest to know what is the best performance that could have been achieved. For uncertain multivariable systems, this is of interest to know what is the best performance that could have been achieved. Furthermore, all of the proposed methods involve significant designer intervention and most are tested on relatively small example problems, usually 2×2 systems, and in a few cases, 3×3 systems.

With the above motivation in mind, the objective of work reported in this paper was to develop an automated synthesis method to derive the optimal decentralized feedback controller, possibly augmented by a multivariable feedforward controller, and to determine the best-possible performance that it can achieve, given a desired trade-off between the output channels.

The design of decentralized control requires the resolution of two design problems:

a) The pairings between control variables and process outputs must be defined.

b) The structure and parameters of the FB-FF (feedback-feedforward) control system must be selected and optimized.

The second of the above problems involves an objective function space that is multi-dimensional with potentially a large number of local optima. It is nonlinear in the parameters, non-convex and discontinuous in the parameter space. Often conventional first order (or higher) optimization methods relying on gradient descent from arbitrary initial guesses of the optimal solution may experience difficulties even converging in such cases. This is the motivation for considering reliable zero-order optimization methods, such as genetic algorithms.

Genetic algorithms (GAs) are modeled on the process of natural evolution found in nature (Goldberg, 1989). Whereas classical optimization methods often rely on local gradient search, genetic algorithms keep track of a population of potential solutions, and are therefore less sensitive to arbitrary initial guesses of the solution. It has been shown that genetic algorithms are appropriate optimization methods for the design of both feedback controllers (Lewin, 1994, Parag and Lewin, 1996) and feedforward controllers (Lewin, 1996b).

This paper is organized as follows. In Section 2, the constrained optimization problem pertaining to the synthesis of FB-FF multivariable control synthesis is formally defined. Next, a genetic algorithm is described, suitable for the solution of this problem. In Section 4, the algorithm is used to solve a representative 4×4 multivariable process control problem, and compared with solutions available in the literature.

2. FORMULATION OF THE PROBLEM

The classical feedforward-feedback structure shown in Figure 1a can also be expressed in terms of the more conveniently handled perturbation structure of Figure 1b. In Figure 1a, \( P \), is the nominal process, \( \zeta \) and \( \zeta' \), are the feedback and feedforward controllers, respectively, which depend on a vector of tunable parameter values, \( \Gamma \).
the additive model uncertainty, and \( P_m \) models the effect of external disturbances. The weight \( W \) permits frequency dependent upper bounds to be specified for the worst-case closed loop sensitivity function vector. Unity-norm-bounded perturbations \( \Delta_u \) and \( \Delta_w \), appear as multipliers of the model uncertainty and on the artificial performance weight. The model, its uncertainty, and the disturbance model are all assumed stable, linear, and may represent full transfer function matrices with or without time delays and RHP zeros.

![Figure 1](image)

**Figure 1.** (a) The classical feedback-feedforward structure; (b) The perturbation feedback structure.

**Feedback and feedforward controller transfer functions.**

The decentralized feedback controller is limited to second order (i.e. PI or PID):

\[
c_{ij} = \text{sign}(p_{ij}(0))K_{ij}\frac{(\tau_i s + 1)(\tau_j s + 1)}{s(\tau_s s + 1)}
\]

where each row, \( i \) of \( C \) has a single non-zero element in column \( j \). Note that each transfer function element in Eq. (1) has integral action to ensure offset-free response, and a proportional gain, \( K_{ij} \), with its sign equal to that of the corresponding steady-state process gain, which is a necessary condition to ensure stability of each SISO control loop. The orders of the numerator and denominator are selected by the user, and are equal to ensure proper controllers, with the simplest instance being the PI controller (with \( \tau_2 = \tau_1 = 0 \)). To improve regulatory performance, a feedforward controller may be considered, whose components are limited to delayed first order lead-lag terms.

For a square system of dimension \( n \), the total number of possible pairings of a decentralized control scheme is \( n! \). It is important to be able to guarantee that a decentralized control system (including integral action to eliminate offset) remains closed loop stable in the event of one or more loops switching from manual to automatic, or vice versa, or in the event of an actuator or sensor failure. This property is referred to as **Decentralized Integral Controllability** (DIC). Morari and Zafiriou (1989) present sufficiency conditions for DIC depending only on the nominal steady state gain matrix of the process that need to be satisfied by a decentralized controller, which identify feasible pairings. Having determined which pairing configurations satisfy the DIC tests, the parameters of the controllers for each configuration must be optimized, as discussed next.

**Optimization problem formulation.** Transformation of the feedforward-feedback structure in Figure 1a to the perturbation structure in Figure 1b, gives the block matrix:

\[
G = \begin{bmatrix}
\hat{P}^{-1}H L \\
W \hat{E} L \\
\end{bmatrix}
= \begin{bmatrix}
\hat{P}^{-1}H L d \\
W \hat{E} L d \\
\end{bmatrix}
\]

where \( \hat{E} \) and \( \hat{H} \) are the sensitivity and complementary sensitivity function matrices, respectively. Note that the uncertainty perturbation \( \Delta_u \) causes additional feedback, leading to potential destabilization of the control system. As stated previously, only those decentralized controller pairings that satisfy DIC conditions are candidates for investigation, starting with the nominal stability test, in which the nominal process and proposed feedback controller (with its parameters) are tested for closed loop stability. If the proposed controller satisfies nominal stability, robust stability is guaranteed if and only if:

\[
\mu_\Delta(G(\omega)) < 1 \forall \omega
\]

where \( \mu_\Delta(M) \) is the structured singular value (SSV) of the matrix \( M \) (Doyle, 1982), and is the least conservative measure of its magnitude. As shown in Figure 1b, the perturbation structure also allows the treatment of the robust performance objective in the same manner as robust stability (see Doyle, 1982, and Morari and Zafiriou, 1989), which is satisfied provided

\[
\mu_\Delta(G(\omega)) < 1 \forall \omega
\]

Note that fulfilling the robust performance condition (4) implies compliance with the robust stability condition (3). The selection of the parameters of performance weight matrix, \( W \), is critical, noting that when Eq. (4) is satisfied at the limit, this implies that the performance weight constitutes a tight upper bound on the peak sensitivity function at a critical frequency. A commonly used formulation for the performance weight is a diagonal matrix, \( W \), comprising of elements of the form:

\[
w_{ij}(s) = \alpha_i(\beta_i s + 1)/\beta_i s
\]

This formulation allows the designer to indirectly specify time domain specifications for the response of each output channel, \( i \), with the permitted overshoot being inversely proportional to \( \alpha_i \), and the desired bandwidth inversely proportional to \( \beta_i \) (Lewin, 1994). Thus, the definition of performance specifications involves fixing a value for \( \alpha_i \) and a desired ratio between the settling times of the process output channels. Note that \( \alpha_i \) must be set to a value less than unity, since \( W^{-1} \) imposes a bound on the peak value of the sensitivity function \( \hat{E} \), which for adequate performance is usually permitted to exceed unity. This can be conveniently specified as a vector containing normalized channel closed-loop time constants:

\[
\bar{\beta}_i = [\beta_1, \beta_2, \ldots, \beta_n]
\]
Henceforth, $\bar{\beta}_i$ is referred to as the bandwidth trade-off basis, permitting the actual values of the individual channel bandwidth specifications to be expressed in terms of a single trade-off parameter, $\gamma$:

$$\beta_i = \gamma \cdot \beta^*_i, i = 1, \ldots, n$$  \hspace{1cm} (7)

In principle, an appropriate bandwidth trade-off basis could be estimated on the basis of the nominal process model resiliency limitations, such as process delays, right half plane zeros and model uncertainty, using the approach proposed by Morari and Zafiriou (1989). In practice, this ratio could be altered from the achievable ratio that might be suggested by analysis of the model, but potentially at the price of significant deterioration of the performance of the less important channels (Lewin, 1994). The proposed parameterization of Eq. (7) reverts the robust performance of $\bar{\beta}_i$ and $\alpha_i$ since for a given controller structure, smaller values of $\gamma$ should be expected (i.e., more aggressive control action and faster closed-loop dynamic response) when smaller values are selected for $\alpha_i$ (i.e., more oscillatory responses are tolerated).

3. A GENETIC ALGORITHM FOR ROBUST CONTROL SYNTHESIS

The solution of optimization problem P1 above involves an objective function space that is multi-dimensional with potentially a large number of local optima, nonlinear in the parameters and non-convex and discontinuous in the parameter space. Difficulties expected when using gradient-based methods motivate the use of a stochastic optimization method, such as a genetic algorithm. However, genetic algorithms, which are unconstrained methods by nature, are difficult to apply directly to problem P1. One possibility would be to include the constraint into the objective function using a penalty function, but this approach is extremely sensitive to the penalty function weight. The approach used reformulates problem P1 to the following equivalent statement:

$$\min \sum J = \sup_{\alpha} \mu \left( G \left( \alpha, \Sigma, \gamma \right) \right)$$  \hspace{1cm} (9)

with $\gamma_{k+1} = \gamma_k - K_\mu \left| \mu \left( G \left( \alpha, \Sigma, \gamma \right) \right) \right|^{-1}$

Problem formulation P2 relies on the fact that for large values of $\gamma$, the minimized value of $J$ will be less than unity, implying relaxed performance specifications. In contrast, for small values, the minimized $J$ would yield values above unity, indicating unachievable specifications. Thus, the unconstrained minimization of Eq. (9) is in fact implicitly constrained by the value of $\gamma$ derived using Eq. (10), which guarantees that the converged solution meets the constraint. Clearly, $K_\mu$ must be greater than unity, so that when the median value of $\mu$ is less than unity, the value of $\gamma$ is decreased, and vice versa, with larger values giving correction that is more aggressive. A value of $K_\mu = 1.5$ is selected from experience to provide smooth convergence. Problem P2 is solved using a constrained genetic algorithm (CGA), which is concisely expressed in the pseudo-code shown in Figure 2.

```
Procedure CGA;
Begin
k = 0; Initialize population $\pi(0)$;
Evaluate $\pi(0)$;
While (~termination condition)
  Update $\gamma$ using Eq. (10);
  Select $\pi(k)$ from $\pi(k-1)$;
  Reproduce or recombine $\pi(k)$;
  Evaluate $\pi(k)$;
End
End.
```

Figure 2. Pseudo-code for the constrained genetic algorithm.

Here, $\pi(k)$ is the solution population at the $k$’th generation, with each individual solution being the parameter values pertaining to a given control configuration. As in all genetic algorithms, the CGA requires that each solution be expressed as a chromosome, which is represented as a binary bit string (i.e., lists of 0’s and 1’s). For $n_{\text{pars}}$ parameters discretized by $n_{\text{disc}}$ bits, each potential design is represented by a binary word of length $n_{\text{pars}} \times n_{\text{disc}}$. The pseudo-code calls for a number of functionalities, again all of them characteristics of all genetic algorithms:

(a) An evaluation function. In this study, the peak value of the constrained SSV is used to rate fitness. Note that the value of $\gamma$ is updated in each generation using Eq. (10), to gradually force the robust performance condition to its unity limit.

(b) A way to initialize a population of solutions. The initial population, of $n_{\text{pop}}$ solutions, is generated with randomly selected parameter values, but such that all solutions are nominally stable and are not excessively in violation of the initial robust performance specifications; an upper limit on the peak SSV of 1.5 is selected for this purpose. Note that initially, $\gamma$ is set to be a multiple of the dominant open-loop time constant.

(c) A selection procedure. In fitness-proportional selection, copies of solutions are made in proportion to their fitness.
their fitness, and inserted into a pool, from which the next generation of solutions will be formed.

(d) **Genetic recombination.** Two types of operators are common to most genetic algorithms: crossover and mutation. The former involves the random selection of a crossover point, followed by an exchange of information in the “tails” of the chromosomes (beyond the crossover point) thus procreating two children. Mutation allows a small proportion of the child’s chromosomal information to switch (e.g. 0 to 1) at random, which often assists to overcome problems associated with convergence to local optima (Goldberg, 1989). In the implemented algorithm, \( n_{\text{pop}} \) new solutions are generated by crossover of selected pairs of “parents,” modified by mutation. Subsequently, all of the solutions are then pooled (both parents and children), and the best \( n_{\text{pop}} \) solutions carried over to the next generation.

(e) **Convergence criteria:** The CGA is terminated when the following two conditions are satisfied: (a) The value of \( \mu_{\text{Median}} \) lies between 0.995 and 1; and (b) at least half of the population has peak values of \( \mu \) in the range \( 1 \pm 0.01 \). In all runs of the CGA performed in this study, these criteria converged the algorithm successfully in well under 100 generations.

In the overall algorithm, at the highest level (Level 1: the GA Master), all possible pairings of a decentralized controller featuring integral action which will guarantee closed loop stability are determined, using the DIC test procedure. This level also handles the management of the additional multivariable FF controller, as necessary. For each possible pairing, a second level of procedures (Level 2: the GA Slaves) are spawned to optimize controller parameters for each pairing, \( i \). On convergence of each of the Level 2 optimizations the GA Master can then rate the alternative pairings in decreasing order of performance as indicated by increasing values of the minimized \( \gamma \) for each case.

**Implementation details.** Typical convergence of the CGA is shown in Figure 3, where it is noted that the update formula of Eq. (10) acts to gradually drive the median value of the peak SSV to its unity limit by nonlinear feedback control, and consequently, brings \( \gamma \) to its minimum value. It is also noted in Figure 3 that the number of new solutions accepted in each generation gradually decreases as convergence is approached, since the chances of better solutions being generated by genetic operators decreases as the true performance constraint is approached. As this limit is approached, the number acceptable solutions, for which \( \mu = 1 \pm 0.01 \), increases.

For convenience, the entire code is implemented, uncompiled, in MATLAB, making the algorithm rather slow. Typically, a single run of the CGA on the 4×4 system described in this paper takes a few hours on a Pentium III, 1 GHz machine. Obviously, this calculation time will increase significantly with system size. The main bottleneck is the time it takes to perform a single computation of the structured singular value, in which all extremum uncertainty combinations are tested at each frequency. Calculation times can be speeded up, noting that: (a) the CGA is inherently a parallel algorithm, with its implementation on a parallel processor having the potential for near linear speed-up (Parag, 1996); (b) compilation of the code can significantly reduce computation times (by up to two orders of magnitude).

---

**Figure 3.** Typical characteristics of the CGA for Configuration A, \( \mathbf{\alpha} = [2, 2, 2, 2] \): (a) Convergence properties, showing updated \( \gamma \) values (dashed) and median (solid) and best (dotted) \( \mu \), with generation; (b) The number of new solutions (dotted) and number of good solutions (dashed), with generation.

<table>
<thead>
<tr>
<th>Loop</th>
<th>Variable</th>
<th>Average</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loop 1</td>
<td>( K )</td>
<td>-1.45</td>
<td>0.057</td>
</tr>
<tr>
<td></td>
<td>( \tau_1 )</td>
<td>1.87</td>
<td>0.297</td>
</tr>
<tr>
<td></td>
<td>( \tau_2 )</td>
<td>8.44</td>
<td>0.230</td>
</tr>
<tr>
<td></td>
<td>( \tau_3 )</td>
<td>5.84</td>
<td>0.794</td>
</tr>
<tr>
<td>Loop 2</td>
<td>( K )</td>
<td>1.49</td>
<td>0.0098</td>
</tr>
<tr>
<td></td>
<td>( \tau_1 )</td>
<td>4.65</td>
<td>0.462</td>
</tr>
<tr>
<td></td>
<td>( \tau_2 )</td>
<td>7.20</td>
<td>1.22</td>
</tr>
<tr>
<td></td>
<td>( \tau_3 )</td>
<td>4.70</td>
<td>2.13</td>
</tr>
<tr>
<td>Loop 3</td>
<td>( K )</td>
<td>-0.373</td>
<td>0.0009</td>
</tr>
<tr>
<td></td>
<td>( \tau_1 )</td>
<td>1.05</td>
<td>0.178</td>
</tr>
<tr>
<td></td>
<td>( \tau_2 )</td>
<td>0.748</td>
<td>0.084</td>
</tr>
<tr>
<td></td>
<td>( \tau_3 )</td>
<td>0.235</td>
<td>0.021</td>
</tr>
<tr>
<td>Loop 4</td>
<td>( K )</td>
<td>-0.938</td>
<td>0.0021</td>
</tr>
<tr>
<td></td>
<td>( \tau_1 )</td>
<td>8.37</td>
<td>0.253</td>
</tr>
<tr>
<td></td>
<td>( \tau_2 )</td>
<td>9.18</td>
<td>0.707</td>
</tr>
<tr>
<td></td>
<td>( \tau_3 )</td>
<td>9.04</td>
<td>0.378</td>
</tr>
</tbody>
</table>

Table I gives typical statistics of the control parameters derived by the CGA are presented for the same case shown in Figure 3. It is noted that in this case, where the entire population of 50 individuals was acceptable (\( \mu = 1 \pm 0.01 \)), the standard deviations for the gains is at least one order of magnitude smaller than for the time constants, indicative of the relative importance of correctly setting the controller gains. Furthermore, because of the relatively high values of standard deviation for the controller time constants, the
The possibility of pole-zero cancellation should be considered; in Table I, for example, it is noted that $\tau_2$ is approximately equal to $T_1$ in Loop 4. On convergence of all of the spawned GA Slaves, the GA Master rates the alternative pairings/structures in decreasing order of performance as indicated by increasing values of the minimized $\gamma$ for each case.

**Figure 4. The Distillation Column System of Lang and Gilles (1989).**

**4. CASE STUDY: THE LANG PROBLEM**

The proposed approach is applied to the design of decentralized control for two coupled distillation columns (Lang and Gilles, 1989), shown schematically in Figure 4. It is desired to be able to adequately track step changes in the set points to temperatures on the 11th, 30th, 34th and 48th trays, using a decentralized control scheme. The process is approximated by:

\[
\begin{bmatrix}
T_e \\
T_{e1} \\
T_{e2} \\
T_{e3} \\
T_{e4}
\end{bmatrix} =
\begin{bmatrix}
0.092 + 1 & -2.11(0.063 + 1) & \cdots \\
0.049(0.063 + 1) & 0.571 & \cdots \\
0.049(0.063 + 1) & 0.571 & \cdots \\
2.7 & -0.571(0.063 + 1) & \cdots \\
1.75 & -0.571(0.063 + 1) & \cdots \\
0.216 & -0.571(0.063 + 1) & \cdots \\
\end{bmatrix}
\begin{bmatrix}
\gamma_1 \\
\gamma_2 \\
\gamma_3 \\
\gamma_4 \\
\gamma_5 \\
\gamma_6 \\
\gamma_7 \\
\gamma_8
\end{bmatrix}
\]

where time units are hours. Since Lang and Gilles (1989) required a design to perform adequate set point tracking of step changes, this is equivalent to rejection of steps in disturbances, so the disturbance model is set to $P_d = \mathcal{G}(\lambda s + 1)^1$ with $\lambda$ set a suitable response time (with an arbitrary value of 1 hr). A value of $\alpha = 0.5$ is used, and additive static gain uncertainty of 20% is assumed. The speed of response of the first and third output channels, which directly affect product streams, are assumed the most important. To reflect this, two values of $\bar{P}_g$ are compared, $\bar{P}_g = [2, 2, 2, 2]$ and $\bar{P}_g = [2, 10, 2, 10]$. The latter implies tighter specifications on the important channels at the expense of relaxing those of the less important ones.

**Configuration selection by the GA Master.** Since the original problem definition does not include a disturbance model, the GA Master considers only decentralized FB control in this case. For an example application of the CGA for the design of a FB-FF multivariable control, see Parag and Lewin (1996). Only three out of the 24 combinations of possible pairings satisfy the DIC conditions:

A. 2-1-3-4: namely $T_{30}$-QE, $T_{11}$-SAB, $T_{34}$-RL1 and $T_{48}$-RL2. This strategy was attempted by Lang and Gilles (1989), who were unable to generate stabilizing integral control for all four outputs.

B. 2-4-1-3: $T_{30}$-QE, $T_{48}$-SAB, $T_{11}$-RL1 and $T_{34}$-RL2.

C. 2-4-3-1: $T_{30}$-QE, $T_{48}$-SAB, $T_{34}$-RL1 and $T_{11}$-RL2.

**Optimization of each configuration.** The three possible decentralized control configurations are sent to three independent GA Slaves, each of which perform constrained optimization (problem P2) using a CGA for $\bar{P}_g$, each managing a population of 50 individuals. The closed loop responses attained with each of the configurations with the controller tuning parameters determined by the CGAs are shown in Figure 5. The servo responses are for the set point change $\gamma_5 = [1, -1, 1, -1]^T$, which is computed to be the most difficult to track, and the worst-case gain uncertainty is simulated. Comparing the responses with the values of $\gamma$ obtained using the CGA, it is apparent that reduced values of the trade-off parameter indicate that faster responses can be expected in more of the channels. Configuration A, with $\gamma = 1.22$, permits rapid tracking of $T_{30}$ and $T_{48}$, acceptable tracking of $T_{11}$, and sluggish tracking of $T_{34}$. In contrast, Configuration B, with $\gamma = 2.02$, succeeds in achieving rapid settling in only two channels ($T_{11}$ and $T_{34}$), and Configuration C, with $\gamma = 1.06$, manages to quickly settle all of the channels except $T_{34}$, whose settling time and oscillatory behavior are unacceptable. Based on these results, it would appear that Configuration A is the most desirable. Furthermore, the speed of response of the more important channels $T_{11}$ and $T_{34}$ are improved by selecting $\bar{P}_g = [2, 10, 2, 10]$. In this case, the CGA gives a value of $\gamma = 1.18$, an indication that the important channels can be speeded up at the expense of the lesser important ones, which is confirmed by simulation as shown in Figure 6. Lang and Gilles (1989) were unable to design stable decentralized PID control system for the complete 4×4 system in Configuration A, and resort to reducing the $T_{34}$-RL1 to proportional control only.

The controller settings generated for each of the two bandwidth trade-off bases tested are compared in Table II. It is noted that the controller gains for the second and fourth channels are significantly decreased when the performance specifications for these are sacrificed in favor...
of the first and third channels. As shown by Lewin (1996b), the components of the control system that do not contribute significantly to the closed loop performance can be eliminated using statistical hypothesis testing, using descriptive statistics generated from the population of solutions satisfying $\mu = 1 \pm 0.01$. Doing so results in the conclusion that specific controllers can be reduced from the PID to the PI form with no loss of performance, which can be confirmed by simulation.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
Loop & $K_c$ & $\tau_I$ \\
\hline
1 & -11.2 & -16.6 \\
2 & 7.11 & 15.3 \\
3 & 1.42 & 3.82 \\
4 & 8.43 & 3.58 \\
\hline
\end{tabular}
\caption{PID controller settings for Configuration A.}
\end{table}

\section{CONCLUSIONS}

The synthesis of robust MIMO control including feedback and feedforward components has been posed as a constrained minimization problem. In this formulation, the maximum closed loop bandwidth is sought along a constraint line defined by the unity value of the supremum of the structured singular value of the perturbation structure matrix defining robust performance. It has been shown that this constrained optimization problem can be reformulated as a pseudo-unconstrained problem, in which an internal control mechanism recursively updates the value of a single robustness-performance trade-off parameter, which gradually drives the solution to meet the constraint. This modified problem can be solved efficiently and robustly using a multilevel genetic algorithm. The implementation of this algorithm enables the synthesis of the most appropriate control structure, given a process model, its uncertainty and the desired relative importance of the performance of each output channel.

\begin{figure}[h]
\centering
\includegraphics[width=0.7\textwidth]{figure5.png}
\caption{Closed loop responses to step changes in all set points for $\beta = [2,2,2,2]$ for maximum gain uncertainty for the three DIC configurations: $T_{ij} = $ solid lines, $T_{im} = $ dashed lines, $T_{id} = $ dot-dashed lines, $T_{id} = $ dotted lines: (a) Configuration A, $\gamma = 1.2192$; (b) Configuration B, $\gamma = 2.019$; (c) Configuration C, $\gamma = 1.0675$.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.7\textwidth]{figure6.png}
\caption{Closed loop responses to step changes in all set points for maximum gain uncertainty values for Configuration A, with $\beta = [2,10,2,10], \gamma = 1.1785$ (Compare with Figure 5a).}
\end{figure}

\begin{thebibliography}{99}
\bibitem{Bibliography}
\bibitem{Hovd, M. and Skogestad, S. (1994): Sequential Design of Decentralized Controllers, Automatica, 30(10), 1601-1607.}
\bibitem{Van Antwerp, J., Featherstone, A. P., and Braatz, R. D. (2001): Robust Cross-directional Control of Large-scale Sheet and Film Processes, Jnl. of Process Control, 11, 149-177.}
\end{thebibliography}