Robust Design of Terminal ILC with an Internal Model Control Using $\mu$-analysis and a Genetic Algorithm Approach

Guy Gauthier, Member IEEE, Benoit Boulet, Senior Member, IEEE

Abstract—The thermoforming heater temperature setpoints can be automatically tuned with cycle-to-cycle control. Terminal Iterative Learning Control (TILC) is used to adjust the heater temperature setpoints so that the temperature profile at the surface of the plastic sheet converges to the desired temperature. Industrial thermoforming ovens generally have a large number of temperature sensors and heaters, which makes the design of TILC difficult. The proposed TILC design is based on Internal Model Control (IMC). The robustness of a closed-loop system with this TILC algorithm is measured using the $\mu$-analysis approach. A Genetic Algorithm (GA) is used to find the IMC exponential filter parameters giving the most robust closed-loop system. Simulation results are included to show the effectiveness of this robust TILC algorithm.

I. INTRODUCTION

In the thermoforming industry, the reheat phase is an important part of the process, since plastic sheets have to be heated to the right temperature before being molded [1-4]. Up to now, the heater temperature setpoints have been adjusted manually, by trial and error. This manual adjustment takes some cycles to complete, and results in monetary loss owing to the production of rejected parts.

To improve heater temperature setpoint tuning, we can use the Terminal Iterative Learning Control (TILC) algorithm [5-8]. This algorithm is an efficient cycle-to-cycle control technique which was introduced first in [9] and then in a PhD thesis [10]. The high-order TILC in [9-12] was proposed to improve robustness. TILC is a variant of Iterative Learning Control (ILC). The latter has access to measurements sampled during the entire cycle, while the former has access to measurements sampled only at the end of the cycle. This is the main difference between them [7, 13].

To use the TILC approach on a thermoforming machine, temperature sensors are installed for measuring the surface temperature of the plastic sheet at the end of the cycle are installed [4, 8, 14]. TILC adjusts the heater temperature setpoints so that the sheet surface temperature converges to the desired temperature at the end of the heating cycle [7]. However, it is not easy to design and tune the TILC algorithm for a big system.

One way to simplify the design is the use of IMC. IMC is used in the run-to-run control approach [15-18] and reduces the number of parameters to tune. To tune the TILC with IMC, a Genetic Algorithm (GA) is used select the best set of IMC filter parameters. The GA selects the best candidates by evaluating the robustness of the closed-loop system with the $\mu$-analysis. At the last generation of the GA, the individual with the genes giving the most robustness is selected to tune the TILC algorithm.

Section II presents the system used to design the TILC. Section III introduces a first-order TILC with IMC, and shows the stability of the system. The $\mu$-analysis concept and parameters are explained in Section IV. Section V presents the genetic algorithm used to select the best parameters for the TILC algorithm. Simulation results, using the TILC designed by this approach, are shown in Section VI. Section VII concludes the paper.

II. PROBLEM SETUP

The system on which we apply the TILC algorithm is a thermoforming machine which produces parts in a repetitive way. For the TILC design, the system is linearized [4-7, 14] to this state space system:

$$\dot{x}(t) = Ax(t) + Bu,$$
$$y(t) = Cx(t)$$

(1)

Time within the cycle and cycle number are expressed by $t \in \mathbb{R}$ and $k \in \mathbb{N}$ respectively. Matrices $A$, $B$ and $C$ are time-invariant. The state vector $x_k(t) \in \mathbb{R}^n$ expresses the temperature at $n$ points on the plastic sheet [4-7]. The input vector $u_k \in \mathbb{R}^m$ contains the temperature of the heaters, and those temperatures are maintained constant during the entire reheat cycle. The surface temperatures of the plastic sheet are in the output vector $y_k(t) \in \mathbb{R}^p$ [4-7].

The control task is to update the heater temperature setpoint $u_k$ so that the sheet surface temperatures converge to a desired terminal value vector $y(T) \in \mathbb{R}^p$ at time $T$. From (1), we can find the terminal output:

$$y(T) = \Gamma x(0) + \Psi u_k.$$  

(2)
In (2), matrix \( \Gamma \in \mathbb{R}^{p \times m} \) is used to obtain the zero-input response and is defined as
\[
\Gamma = Ce^{\Delta T} . \tag{3}
\]
Matrix \( \Psi \in \mathbb{R}^{p \times m} \) is used to obtain the zero-state response and is defined by
\[
\Psi = \int_0^T e^{A(t-\tau)} B d \tau . \tag{4}
\]
To put the emphasis on the cycle domain, expressed by \( \kappa \), the notation can be changed, and so (2) is rewritten as
\[
y_{\gamma}[k] = \Gamma x_0[k] + \Psi u[k] , \tag{5}
\]
where \( y_{\gamma}[k] := y_\kappa(T) \), \( u[k] := u_k \) and \( x_0[k] := x_\kappa(0) \).

Since it is a discretized system in the cycle domain, the z-transform is a useful tool for analyzing it. Then, the z-transform of (5) from the cycle domain is
\[
\hat{y}_\gamma(z) = \Gamma \hat{x}_0(z) + \Psi \hat{u}(z) \tag{6}
\]
where the z-domain variables have caret above then.

### III. TILC DESIGN USING IMC

A TILC algorithm is defined in z-domain, by
\[
\hat{u}(z) = C(z)(\hat{y}_\gamma(z) - \hat{\gamma}_\gamma(z)) . \tag{7}
\]

The closed-loop transfer function of the system (6) controlled with the TILC in (7) is expressed by
\[
\hat{y}(z) = (I_p + \Psi C(z))^{-1}\{\Psi C(z)\hat{y}_\gamma(z) + \Gamma \hat{x}_0(z)\} , \tag{8}
\]
or
\[
\hat{u}(z) = C(z)(I_p + \Psi C(z))^{-1}\{\hat{y}_\gamma(z) - \Gamma \hat{x}_0(z)\} . \tag{9}
\]

#### Lemma 1: The closed-loop system is internally stable if and only if the following matrix is invertible:
\[
\begin{bmatrix}
I_w & C(z) \\
-\Psi & I_p
\end{bmatrix} \tag{10}
\]
for all \( z \).

**Proof:** The proof can be found in [19]. \( \square \)

The TILC algorithm used in this paper is based on IMC. To tune the controller, IMC uses exponential filter parameters \( \alpha_i \), where \( 0 \leq \alpha_i < 1 \) and \( i \in \{1,2,\cdots,p\} \). The first-order TILC algorithm, using IMC, is defined by
\[
C(z) = (z-1)\Psi^+(I_p - Q) , \tag{11}
\]
where the + exponent represents the pseudoinverse operator and all filter parameters are on the diagonal of matrix \( Q \in \mathbb{R}^{p \times p} \):
\[
Q = \begin{bmatrix}
\alpha_1 & 0 & \cdots & 0 \\
0 & \alpha_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \alpha_p
\end{bmatrix} \tag{12}
\]

With the controller defined in (11) and the filter parameters in (12), the closed-loop transfer functions expressed by (8) and (9) become, respectively,
\[
\hat{y}(z) = (I_p(z-1) + \Psi \Psi^+(I_p - Q))^{-1}\{\Psi \Psi^+(I_p - Q)\hat{y}_\gamma(z) + \Gamma (z-1) \hat{x}_0(z)\} , \tag{13}
\]
or
\[
\hat{u}(z) = \Psi^+(I_p - Q)(I_p(z-1) + \Psi \Psi^+(I_p - Q))^{-1}\{\hat{y}_\gamma(z) - \Gamma \hat{x}_0(z)\} . \tag{14}
\]

Before showing that the closed-loop system is internally stable, we need to introduce some Lemmas.

In (13) and (14), the product \( \Psi \Psi^+ \) gives an idempotent matrix. The following Lemma is about the spectral decomposition of an idempotent matrix.

#### Lemma 2: Suppose that matrix \( A \in \mathbb{R}^{p \times p} \) is idempotent and its rank is equal to \( r \). The spectral decomposition of \( A \) can be written as
\[
A = V \Lambda V^{-1} \tag{15}
\]
where \( V \in \mathbb{R}^{p \times p} \) is a matrix, each column of which is an eigenvector of \( A \), and \( \Lambda \in \mathbb{R}^{p \times r} \) is a diagonal matrix defined as
\[
\Lambda = \begin{bmatrix}
I_r & 0 \\
0 & 0
\end{bmatrix} \tag{16}
\]

**Proof:** Obviously, since the eigenvalues of an idempotent matrix are either 0 or 1, then the multiplicity of the 1 eigenvalue is equal to the rank of the idempotent matrix. \( \square \)

#### Lemma 3: Suppose that \( A \in \mathbb{R}^{p \times p} \) is a rank \( r \) idempotent matrix with the spectral decomposition defined in Lemma 2. Define the following matrix:
\[
\Theta = AQ + (I_p - A) \tag{17}
\]
where the matrix \( Q \) was defined in (12). Then, \( r \) of the eigenvalues of matrix (17) comes from the term \( AQ \) and the remaining \( p-r \) eigenvalues come from the term \( I_p - A \). Those \( p-r \) eigenvalues are all equal to 1.

**Proof:** The eigenvalues of \( \Theta \) are the solutions of
\[
\det(\lambda I_p - \Theta) = 0 \tag{18}
\]
From the spectral decomposition of \( A \) defined in (15) and (17), the term on the left-hand side of (18) becomes
\[
\det(\lambda I_p - \Theta) = \det\left(\lambda I_p - V \Lambda V^{-1} Q - (I_p - V \Lambda V^{-1})\right) . \tag{19}
\]
Since \( Q = QI_p \) and \( I_p = VV^{-1} \), we can write:
\[
\det(\lambda I_p - \Theta) = \det(V \left(\lambda I_p - \Lambda V^{-1} Q - (I_p - \Lambda)\right) V^{-1}) \tag{20}
\]
To shorten this expression, define \( P = V^{-1}QV \) and 
\( B = I_p - \Lambda \). Using the properties of the determinant, (20) reduces to
\[
\det(\lambda I_p - \Theta) = \det(\lambda I_p - \Lambda P - B). \tag{21}
\]

Since matrix \( A \) is idempotent and, from Lemma 2, where matrix \( \Lambda \) was defined in (16), we can see that the last \( p-r \) entries of the main diagonal of \( B \) are equal to 1 and the remaining entries are 0, since \( B = I_p - \Lambda \).

Now, matrix \( P \) must be partitioned, such that
\[
P = \begin{bmatrix}
P_{11} & P_{12} \\
P_{21} & P_{22}
\end{bmatrix}
\]

with \( P_{11}, P_{22} \in \mathbb{R}^{r \times r} \), \( P_{12}, P_{21} \in \mathbb{R}^{(p-r) \times r} \) and \( P_{22} \in \mathbb{R}^{(p-r) \times (p-r)} \).

Then, the structure of the matrix in the determinant (21) look like this
\[
\det(\lambda I_p - \Theta) = \det\begin{bmatrix}
\lambda I_p - P_{11} - P_{12} \\
0 & (\lambda - 1)I_{p-r}
\end{bmatrix}
\]

Using the determinant properties, (23) is simplified to
\[
\det(\lambda I_p - \Theta) = \det(\lambda I_p - P_{11}) \det((\lambda - 1)I_{p-r}) = (\lambda - 1)^{(p-r)} \det(\lambda I_p - P_{11})
\tag{24}
\]

Equation (24) shows that the \( p-r \) eigenvalues of \( B \) are all equal to 1. The remaining \( r \) eigenvalues come from the \( P_{11} \) entry that comes from the \( AQ \) term. \( \square \)

Because \( A \) is idempotent and \( Q \) is diagonal, both of them are Hermitian and positive semi-definite. For the following Lemma, it is assumed that the eigenvalues are arranged in decreasing order: \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \).

**Lemma 4:** Suppose that \( A \) and \( Q \) are positive semi-definite Hermitian matrices. The eigenvalues of the product \( AQ \) are defined by
\[
\min_{\lambda \in \mathbb{C}} \left[ \lambda(A) \lambda(Q) \right] \geq \min_{\lambda \in \mathbb{C}} \left[ \lambda(A) \lambda_{p+1-i}(Q) \right] \tag{25}
\]

when \( 1 \leq k \leq \min(r, \text{rank}(Q)) \) or
\[
\lambda_p(AQ) = 0 \tag{26}
\]

when \( p \geq k > \min(r, \text{rank}(Q)) \).

**Proof:** The proof can be found in [20]. \( \square \)

**Theorem 1:** Suppose that \( A \) is an idempotent matrix of rank \( r \) and \( Q \), defined by (12), with parameters \( \alpha_i \) such that \( 0 \leq \alpha_i < 1 \) for \( i \in \{1, 2, \ldots, p\} \). The matrix resulting from the product \( AQ \) has eigenvalues \( \lambda_i(AQ) \) such that \( 0 \leq \lambda_i(AQ) < 1 \) for \( i \in \{1, 2, \ldots, p\} \).

**Proof:** From (25), using \( k = 1 \), the limit of the maximum eigenvalue of \( AQ \) is
\[
\lambda_i(A)\lambda_i(Q) \geq \lambda_i(AQ) \geq \max_{i \leq i \leq p} \left[ \lambda_i(A) \lambda_{p+1-i}(Q) \right] \tag{27}
\]

Since \( \lambda_i(A) = 1 \) and \( \lambda_i(Q) = \max_{i \leq i \leq p} \), (27) is simplified to
\[
\max_{i \leq i \leq p} \lambda_i(A) \lambda_{p+1-i}(Q) \geq \lambda_i(AQ) \geq \max_{i \leq i \leq p} \lambda_i(A) \lambda_{p+1-i}(Q) \tag{28}
\]

Because the parameters \( \alpha_i \) in matrix \( Q \) are \( 0 \leq \alpha_i < 1 \) and matrix \( A \) is idempotent, then \( 0 \leq \lambda_i(AQ) < 1 \).

From (25), with \( k = p \), the limit of the minimum eigenvalue of \( AQ \) is:
\[
\min_{i \leq i \leq p} \left[ \lambda_i(A) \lambda_{p+1-i}(Q) \right] \geq \lambda_p(AQ) \geq \lambda_p(A) \lambda_p(Q) \tag{29}
\]

If either \( A \) or \( Q \) is rank-deficient, then \( \lambda_p(AQ) = 0 \). If both \( A \) and \( Q \) are full-rank, then \( \lambda_i(A) = \lambda_p(A) = 1 \) and \( \lambda_p(Q) = \min_{i \leq i \leq p} \alpha_i \), and (29) is simplified to
\[
\min_{i \leq i \leq p} \alpha_i \geq \lambda_p(AQ) \geq \min_{i \leq i \leq p} \alpha_i \tag{30}
\]

Again, since the parameters \( \alpha_i \) are in the range \( 0 \leq \alpha_i < 1 \), then \( 0 \leq \lambda_p(AQ) < 1 \).

Thus, all eigenvalues \( \lambda_i(AQ) \) of the product \( AQ \) are such that \( 0 \leq \lambda_i(AQ) < 1 \) for \( i \in \{1, 2, \ldots, p\} \). \( \square \)

Now we are ready to demonstrate the stability of the closed-loop system.

**Theorem 2:** Suppose that a system represented by the matrix \( \Psi \) is controlled with the TILC algorithm expressed by (11). Then the closed-loop system is internally stable.

**Proof:** From Lemma 1, the system is internally stable if (10) is invertible.
\[
\begin{bmatrix}
I_m \\
-\Psi
\end{bmatrix}
\begin{bmatrix}
C(z) \\
I_p
\end{bmatrix}
= \begin{bmatrix}
I_m \\
-\Psi
\end{bmatrix}
\begin{bmatrix}
(z-1)^{-1} \Psi \star (I_p - Q) \\
I_p
\end{bmatrix}
\tag{31}
\]

Using
\[
p(z) := \det\begin{bmatrix}
I_m \\
-\Psi
\end{bmatrix}
\begin{bmatrix}
C(z) \\
I_p
\end{bmatrix}
= \det(I_p + \Psi \Psi \star (I_p - Q)(z-1)^{-1})
\tag{32}
\]

then
\[
p(z) = (z-1)^{-r} \det(I_p z - \Psi \Psi \star Q - (I_p - \Psi \Psi \star)) \tag{33}
\]
The solutions of \( p(z) = 0 \), with \( p(z) \) defined in (33) correspond to the poles of the closed-loop system. From Lemmas 2, 3 and 4 and Theorem 1, we can show that the poles not cancelled by the \((z-1)\) denominator term are strictly inside the unit circle. \( \square \)

The system controlled with a first-order TILC defined in (11) is stable when the system is equal to its model used in the IMC. When this is not the case, we need to choose the filter parameters \( \alpha \) such that the system remains stable. The system must also perform well because we need to have fast convergence to limit the number of wasted plastic sheets.

One way to measure the robustness of the TILC algorithm with IMC is to use \( \mu \)-analysis.

\[ \mu \text{-analysis is used to measure the robustness} \] \[ [19, 21] \text{of a given TILC algorithm. Figure 1 shows the detailed block diagram of the system matrix} \Psi \text{with its uncertainties. That system is related to the nominal one,} \Psi_0 \text{, with the weighting functions and matrices necessary to define the uncertainty of the system} [7]. \]

Each matrix appearing in Figure 1 has to be defined before going further.

The system to control is expressed by its nominal matrix \( \Psi_0 \in \mathbb{R}^{pm \times pm} \), and the uncertainty amplitudes on each entry of the system matrix \( \Psi \) are expressed by each entry \( \Delta_{\psi,ii} \in \mathbb{R}_+ \), \( \forall \{1,2,\cdots,r\} \) of the real diagonal matrix \( \Delta_\psi \). The size of matrix \( \Delta_\psi \) depends on the number of uncertain real entries of the system, but the maximum size is \( pm \times pm \).

The real diagonal matrix \( \Delta \) is such that each entry on the main diagonal is strictly smaller than 1 \( (|\Delta| < 1) \). Therefore, the size of the matrix \( \Delta \) is the same as \( \Delta_\psi \). To associate each entry \( \Psi_{a,ii} \) of \( \Psi_0 \) with the corresponding uncertainty amplitude \( \Delta_{\psi,ii} \), we need two real matrices identified by \( \Delta_\psi \) and \( \Psi_0 \). Then, the uncertain system can be written as
\[
\Psi = \Psi_0 + \Psi_\Delta \Delta_\psi W_U
\]

When all parameters of the system \( \Psi \) are uncertain, the matrix \( W_U \in \mathbb{R}^{pm \times pm} \) is
\[
W_u = \left[ I_m \quad I_m \quad \cdots \quad I_m \right]^T,
\]
where the identity matrix \( I_m \) is repeated \( p \) times and matrix \( W_\gamma \in \mathbb{R}^{pm \times pm} \) is
\[
W_\gamma = I_p \otimes [1 \quad 1 \quad \cdots \quad 1]
\]
where the 1 is repeated \( m \) times.

For the parameters with no uncertainty, the corresponding lines and columns have to be removed from matrices as shown in Figure 2 [7].

The uncertain system \( \Psi \) is connected to a cycle-to-cycle control to close the loop in the cycle domain.

\[ \text{Figure 4 shows the TILC. The uncertainty and TILC} \]
blocks are put into distinct blocks and all the other blocks
are grouped into a block named $N$. From the block diagram in Figure 4, we can write
\[
N := \begin{bmatrix}
0 & 0 & \Delta \Psi W_L \\
-W_L W_r & W_r & -W_L \Psi_0 \\
-W_r & I & -\Psi_0
\end{bmatrix}. \tag{38}
\]
This $N$ matrix is the main component of the following relationship between input and output shown in Figure 4:
\[
\begin{bmatrix}
p \\ z \\ e
\end{bmatrix} = N \begin{bmatrix}
q \\ y_d \\ u
\end{bmatrix}. \tag{39}
\]

\[
\text{Figure 4: Block diagram of the closed-loop system}
\]

To be able to perform the $\mu$-analysis, the controller is attached to the system with the following lower linear fractional transformation (LFT):
\[
P := P_f(N, C) = N_{11} + N_{12} C (I - N_{22} C)^{-1} N_{21}. \tag{40}
\]

In (40), $N$ is partitioned as follows:

\[
N_{11} = \begin{bmatrix}
0 & 0 \\
-W_L W_r & W_r
\end{bmatrix}, \tag{41}
\]

\[
N_{12} = \begin{bmatrix}
\Delta \Psi W_L \\
-W_L \Psi_0
\end{bmatrix}, \tag{42}
\]

\[
N_{21} = \begin{bmatrix}
-W_r \\
I
\end{bmatrix}, \tag{43}
\]

\[
N_{22} = -\Psi_0. \tag{44}
\]

Then, the lower LFT can be rewritten as
\[
P = \begin{bmatrix}
-\Delta \Psi W_L U_r W_r & \Delta \Psi W_L U_0 \\
W_r S_r W_r & W_r S_0
\end{bmatrix}. \tag{45}
\]

In (45), the nominal input and output sensitivities ($S_0$ and $U_0$) of the closed-loop system are given by
\[
S_0 = (z - 1) I_p z - Q)^{-1} \tag{46}
\]
\[
U_0 = \Psi_0^* (I_p - Q)(I_p z - Q)^{-1} \tag{47}
\]

The performance specification is expressed by a fictitious uncertainty matrix $\Delta \in \mathbb{C}^{n \times p}$ connecting the output $z$ of the weighting function $W_r$ to the input $y_d$.

This fictitious uncertainty matrix $\Delta$ is such that
\[
\|\Delta\|_\infty = \sigma(\Delta(x)) < 1 \tag{48}
\]
at all frequency. The matrix $\Delta$ is a full matrix of complex values. Both uncertainty matrices, $\Delta$ and $\Delta p$, are combined into this block diagonal matrix:
\[
\Delta = \begin{bmatrix}
\Delta & 0 \\
0 & \Delta_p
\end{bmatrix}. \tag{49}
\]

Note that $\Delta$ satisfies
\[
\|\Delta\|_\infty = \sigma(\Delta) = \max \{ \sigma(\Delta(x)), \sigma(\Delta_p) \} < 1 \tag{50}
\]
at all frequencies.

Since the norm of $\Delta$ is smaller than 1, combining the system expressed by matrix $N$ and the controller $C$, using the small gain theorem, must gives a matrix $P$ having a gain (or norm) smaller than 1 at all frequencies.

The measure of robustness is obtained by the structured singular value $\mu_\Delta(P)$ . This measure is defined by [19, 21]
\[
\mu_\Delta(P) := \frac{1}{\min \{ \sigma(\Delta(x)), \det(I - PA) \}} \tag{51}
\]

$\mu_\Delta(P)$ is evaluated with the mu function in Matlab.

V. GENETIC ALGORITHM

The GA is used to find the values of the filter parameters $\alpha_i$ that minimize the value of $\mu_\Delta(P)$ and made the closed-loop system robust. This GA is shown in Figure 5.

An initial population, of size $N$, is generated with a number of randomly generated filter parameters $\alpha_i$ to cover the space of all possible $\alpha_i (i \in \{1, 2, \cdots, p\})$. Each chromosome has a length equal to $p$ genes and contains the real $\alpha_i$ values corresponding to the main diagonal of matrix $Q$.

At each generation, the robustness measure $\mu_\Delta(P)$ is evaluated for all chromosomes, and the fitness function is equal to the inverse of $\mu_\Delta(P)$ . Once the fitness value has been determined, the choice of which of the $N$ individuals of the population will be used for mating is performed by roulette-wheel selection [22].

Each pair of selected individuals generates an offspring by real-value recombination. The algorithm for recombination is the intermediate recombination [23] defined by
\[
\alpha_i^{\text{offspring}} = a_i \alpha_i^{\text{parent} \_1} + (1 - a_i) \alpha_i^{\text{parent} \_2} \tag{52}
\]
with $a_i \in [-0.25, 1.25]$ an uniform random variable, for $i \in \{1, 2, \cdots, p\}$.
Following real-value recombination, every chromosome of all offspring can have a mutation with a probability of 5%. The real-value mutation is defined by [23]:

$$ \alpha_{Mutated} = \alpha_{Offspring} + s_i \cdot 2^{-u} $$  \hspace{1cm} (53)

with $u \in [0,1]$ a uniform random variable and $s_i \in (-1,+1)$ another uniform random variable. Note that all random variables are independents.

The first TILC was designed for an oven configuration of two heaters and two sensors. The weighting function parameters of $W_i$ in this case are $M_i = 2$, $c_i = 0.01$ and $\omega_i = 0.5$.

The initial population is 50 randomly generated individuals having chromosomes containing 2 parameters each. At each generation, 25 of offspring are created and the best 50 of the 75 individuals are kept at each generation.

After 50 generations, the best individual has the following chromosome: $\alpha_1 = 0.3032$ and $\alpha_2 = 0.3031$. That gives a closed-loop system with $P_\mu = 0.8487$. The square markers in Figure 7 show the simulation results with the controller using the filter parameters coded in the chromosome.

The targets surface temperatures are 150°C for the top surface of the plastic sheet and 151°C for the bottom.

Figure 5: Genetic algorithm

The initial population of N individuals
Fitness evaluation of each individual
Selection of individuals for mating
Creation of N offspring by recombination
Mutation of some offspring
Reinsertion of offspring
Fitness evaluation of N+N offspring
Natural selection of the N best individuals

Figure 6: Heaters and sensors location (bottom heaters and sensors at the same location, with subscript B)

The first TILC was designed for an oven configuration of two heaters and two sensors. The weighting function parameters of $W_i$ in this case are $M_i = 2$, $c_i = 0.01$ and $\omega_i = 0.5$.

The initial population is 50 randomly generated individuals having chromosomes containing 2 parameters each. At each generation, 25 of offspring are created and the best 50 of the 75 individuals are kept at each generation.

After 50 generations, the best individual has the following chromosome: $\alpha_1 = 0.3032$ and $\alpha_2 = 0.3031$. That gives a closed-loop system with $P_\mu = 0.8487$. The square markers in Figure 7 show the simulation results with the controller using the filter parameters coded in the chromosome.

The targets surface temperatures are 150°C for the top surface of the plastic sheet and 151°C for the bottom.

The temperature error falls below 5°C at the third iteration. The targets surface temperatures are 150°C for the top surface of the plastic sheet and 151°C for the bottom.
surface. The noise in the measurement had a standard deviation of 1°C. The initial plastic sheet temperature was subject to a slow variation.

The second TILC was designed for an oven configuration with four heaters and four sensors. The approach used was the same as for the first design, but the chromosomes contain 4 parameters each. The weighting function parameters of $W_i$ are the same as previously.

After 50 generations, the best individual has the following chromosome: $α_1 = 0.2992$, $α_2 = 0.2992$, $α_3 = 0.2991$, and $α_4 = 0.2998$, which gives a robust closed-loop system with $μ_5(P) = 0.8688$. The circle markers in Figure 7 show the maximum temperature error. The target temperatures were 150°C for $IRT_1$, 161°C for $IRT_2$, 151°C for $IRB_2$ and 160°C for $IRB_3$.

Finally, the third TILC was designed with the complete oven, a 12 sensor/14 heater configuration. In this case, all filter parameters $α_i$ are set to the same value $α$, each chromosome therefore containing only one parameter. From the GA, the filter parameter obtained by elitism is $α = 0.2804$ and gives $μ_5(P) = 0.9627$.

In this simulation, the desired surface temperatures are (from $IRT_1$ to $IRB_3$): 165 °C, 160 °C, 150 °C, 150 °C, 150 °C, 150 °C, 150 °C, 160 °C, 166 °C, 161 °C, 151 °C, 151 °C, 151 °C, 151 °C, 161 °C. The diamond markers in Figure 7 show the maximum temperature error. This error remains over 5°C.

The simulation is repeated with a different desired surface temperature profile (from $IRT_1$ to $IRB_3$): 170 °C, 156 °C, 146 °C, 152 °C, 146 °C, 146 °C, 162 °C, 171 °C, 157 °C, 147 °C, 154 °C, 147 °C, 147°C, 163 °C. The maximum temperature error converges below 5°C in 6 cycles (triangles markers in Figure 7). This profile is feasible by the oven, since the previous one was not, explaining the higher level of error in for the curve with diamond markers in Figure 7.

VII. CONCLUSION

The design of a TILC, by tuning the IMC exponential filter parameters, leads to a robust closed-loop system as shown by the simulation results. This approach greatly reduces the number of parameters to tune, e.g. from 168 to 1 in the last design. The GA is used to find the system giving the lowest $μ_5(P)$. The initial population of the GA has to cover the space of the possible $Q$ matrix to be able to find the minimal $μ_5(P)$. The closed-loop robustness analysis and TILC design are automated. The price to pay is the length of calculation, which becomes very long for a big system. One way to reduce the calculation burden is to reduce the number of filter parameters (or genes) to tune. It is possible to do this for the thermoforming machine, since the best individual has identical genes on its chromosome.

In future work, we will have to determine whether or not this approach is applicable for higher-order TILCs.

REFERENCES