PSO-based intelligent integration of design and control for one kind of curing process

Xin Jiang Lu, Han-Xiong Li,*, Xiang Yuan

a School of Mechanical & Electrical Engineering, Central South University, Hunan, China
b Department of Manufacturing Engineering & Engineering Management, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, China
c College of Automobile & Mechanical Engineering, Changsha University of Science & Technology, Hunan, China

Article history:
Received 2 July 2009
Received in revised form 24 June 2010
Accepted 26 June 2010

Keywords:
Integration of design and control
Fuzzy modeling
Fuzzy control
Particle swarm optimization (PSO)
Curing process

ABSTRACT

In this paper, a PSO-based intelligent integration of design and control is proposed for one kind of non-linear curing process. This method combines the merits of both fuzzy modeling/control and PSO method, where fuzzy modeling/control is proposed to approximate/control the nonlinear process in a large operating region and the PSO-based intelligent optimization method is developed to solve non-convex and non-differential integration problem with design and control optimized simultaneously. Finally, the proposed method is compared with the traditional sequential method on controlling the temperature profile of a nonlinear curing process.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

The curing process is a very important process in semiconductor packaging industry to provide a desirable temperature profile for setting compounds, such as epoxy resin and encapsulation molding compound that are distributed onto electronic components [1], during the cutting or reflowing process. The key requirement for high quality packaging production is to maintain the uniform temperature and track the required temperature trajectory for the whole cured object. Since the process is of nonlinearly distributed nature and needs to work in a large operating region under uncertainty, it brings a challenge to high quality packaging.

In past decades, much effort has been dedicated to design and control of the curing process using the traditional sequential method, where design and control are optimized as separate step in a sequential fashion. On the one hand, many methods have been studied for the design of curing process, such as, heuristics and experiment based design [2–5] and model-based design [6,7]. On the other hand, there are many previous works that have contributed to control of the curing process, such as, the least-square control [6], optimization control [8–12], adaptive control approach [13], internal model control [14], iterative learning control [15,16], LQG [17,18], PI control [19]. A review with emphasis on temperature control was presented by Roozeboom and Parekh [20]. However, since the controllability and dynamic performance is little considered in the design stage, it can lead to the process difficult to control. In this sense, this traditional sequential method is often inadequate since the poor process design makes it difficult to control. A better method is to integrate design and control to improve the curing performance.

The integration of design and control has received much attention in recent decade because it aims to optimize process design and control simultaneously. Many control methods have been studied in this integration method, such as, PI control [21–23], IMC and MPC [24], LQG [25], $H_\infty$ sensitivity method [26] and Q parameterization method [27]. Though these control methods take stability and robustness of the process into account in the design stage, they are only valid in the vicinity of the operation condition since they are usually designed based on a local linear nominal model. Also, only an operating point is used to model the process, which is difficult to obtain an accurate process model when the process works in a large operating region. Sandoval et al. [28] proposed a robust modeling approach for stability and flexibility. However, this approach is also only valid in the vicinity of the operation condition. Recently, Lu et al. [29] proposed a fuzzy-modeling based integration method for the nonlinear system with parameter uncertainty in a large working region. However, this method is complex since it aims at parameter uncertainty. Parameters of many practical processes are often deterministic and thus this integration approach should be developed to consider the case with deterministic parameters.
Moreover, since the solver of this integration method is based on discrete approximation method, it needs large computational cost and is imprecise. This paper will extend the fuzzy-modeling based integration method to handle the process with the deterministic parameter and propose a new PSO-based intelligent approach to solve the integration problem.

The particle swarm optimization (PSO) method [30] is an intelligent optimization tool. Since this swarm intelligence method is inspired by social behavior of a flock of birds and insect swarms, PSO is not largely affected by the size, nonlinearity, non-convex and non-differential of the problem and can converge to the optimal solution in many problems where most analytical methods fail [31]. Therefore, it can be effectively applied to different optimization problems with advantages over other similar optimization techniques, such as, genetic algorithm (GA) [30–33].

In this paper, a novel intelligent integration of design and control, which combines the merits of both fuzzy modeling/control and PSO, is proposed for one kind of nonlinear curing process. First, fuzzy modeling is used to approximate the process with deterministic parameter in a large operating region. Based on the developed model, fuzzy control rules are developed to stabilize the process and achieve the robust performance. Moreover, process design and controller design are integrated into a unified framework, where fuzzy-modeling based intelligent optimization tool. Since this swarm intelligence method is imprecise. This paper will extend the fuzzy-modeling based intelligent approach to solve the integration problem.

2. Process description and fuzzy modeling

2.1. Process description

The studied curing system [29] is shown in Fig. 1. The curing process has a motion mechanism inside the chamber, which moves a working plate up and down to adjust the curing temperature for curing the IC placed on the lead frame. A separate control system is working plate up and down to adjust the curing temperature for curing the IC placed on the lead frame. A separate control system is required to control the heaters embedded in the heater block. Working plate up and down to adjust the curing temperature for curing the IC placed on the lead frame. A separate control system is required to control the heaters embedded in the heater block.

The heat transfer of the heater block to the IC can be rewritten as

\[ q_{i,j}^{\text{wall}} = k_S \left( \frac{T_{i+1,j}(t) + T_{i-1,j}(t) - 2T_{i,j}(t)}{\Delta x} \right) + k_S \left( \frac{T_{i+1,j}(t) + T_{i,j-1}(t) - 2T_{i,j}(t)}{\Delta y} \right) \]

where \( k \) denotes thermal conductivity, \( S_x \) and \( S_y \) are the cross-sectional area of every zone as shown in Fig. 2.

The radiative heat of a zone is [6]:

\[ q_{i,j}^{\text{rad}} = \varepsilon S_{j,i} f_i(d) \mu(t) - \sigma T_{i,j}^4(t) \]

where \( \varepsilon \) is the emissivity of the lead frame and \( \sigma \) is Boltzmann constant. \( S_{j,i} \) is the surface area of the \((i,j)\) zone, \( \mu \) is the manipulated variable that offers power for heaters, the view factor \( f_i(d) \) from the \((i,j)\) zone to heater block is the function of the design parameters vector \( d = [\theta, H] \) with the detailed derivation given in [29], \( \theta \) is the curve angle of the heater block and \( H \) is the distance between the LF and the heater block as shown in Fig. 2.

Since the heat convection has a small effect compared with the other heat flux, it may be regarded as disturbance. Define

\[ \hat{w}_{i,j}(t) = \frac{1}{m_{i,j}c} (q_{i,j}^{\text{rad}} + q_{i,j}^{\text{wall}} + q_{i,j}^{\text{dist}}) \]

Inserting Eq. (2), Eq. (3) into Eq. (1), the heat transfer model (1) may be rewritten as

\[ m_{i,j}c \frac{dT_{i,j}(t)}{dt} = k_S \left( \frac{T_{i+1,j}(t) + T_{i-1,j}(t) - 2T_{i,j}(t)}{\Delta x} \right) + k_S \left( \frac{T_{i+1,j}(t) + T_{i,j-1}(t) - 2T_{i,j}(t)}{\Delta y} \right) + \varepsilon S_{j,i} f_i(d) \mu(t) - \sigma T_{i,j}^4(t) + \hat{w}_{i,j}(t) \]

Define

\[ x(t) = \begin{bmatrix} T_{1,1}(t) & \cdots & T_{1,p-1}(t) & T_{1,p}(t) & T_{2,1}(t) & \cdots & T_{n,p}(t) \end{bmatrix}^T \]

where \( c \) denotes specific heat, \( m_{i,j} \) and \( T_{i,j} \) are mass and temperature of the \((i,j)\) zone respectively, \( q_{i,j}^{\text{rad}}, q_{i,j}^{\text{wall}} \) and \( q_{i,j}^{\text{dist}} \) are heat flow rate into the \((i,j)\) zone via conduction, radiation from heater block and convection from air respectively. \( q_{i,j}^{\text{wall}} \) represents the unknown heat from the chamber wall, and \( q_{i,j}^{\text{dist}} \) is disturbance.

According to Fourier’s rule of heat conduction, heat conduction across a surface is expressed as

\[ q_{i,j}^{\text{w},i,j} = k_S \left( \frac{T_{i+1,j}(t) + T_{i-1,j}(t) - 2T_{i,j}(t)}{\Delta x} \right) + k_S \left( \frac{T_{i+1,j}(t) + T_{i,j-1}(t) - 2T_{i,j}(t)}{\Delta y} \right) \]
where \( T_i(t) \) is the reference and thus the expression of \( \dot{x}_r(t) \) and its derivative are known beforehand. The superscript \( T \) represents the transpose, the matrix \( \text{diag}([a \ b]^T) \) denotes \( \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \).

The Eq. (4) may be rewritten as the vector form:

\[
\dot{x}(t) = Ax(t) - Cx^4(t) + B(d)u(t) + w(t)
\]

where the matrix \( B(d) \) is the function of the design parameters vector \( d \) and the detailed derivation of the matrices \( A, B(d) \) and \( C \) are given in Appendix A.

Obviously, the process model (6) has strong nonlinearity and is affected by uncertainty and disturbance. Moreover, this process needs to work in a large operating region (Curing temperature range: 40–200°C) in order to track the required temperature profile. Thus, it is difficult to develop a controller directly based on this process model.

A simple linear model is usually used to approximate this process. However, this approximation will produce a large approximation error. This approximation error may lead to a poor performance since the controller is designed based on this simple model. Thus, an effective modeling method should be developed to approximate this nonlinear curing process in a large operating region, and the obtained model should be easy to be controlled.

### 2.2. Fuzzy modeling for the curing process

In this section, a fuzzy model will be used to approximate the nonlinear process in a large operating region. First, a simple state-space model is obtained at each operating point, which is usually called the local model. Then the fuzzy technique is used to connect these local models to form a global model, which can approximate any nonlinear process in a large operating region. Moreover, this fuzzy model is easy to be controlled by well-known fuzzy control because both of them can be designed using the same fuzzy theory.

Here, we will adopt local approximation [34] to construct an exact T–S fuzzy model for the nonlinear process model (6). The nonlinear term in the model (6) is rewritten as

\[
\dot{x}(t) = z(t) \cdot x(t)
\]

Then, the nonlinear process model is represented by the following fuzzy model:

**Model rule** \( \nu \):

IF \( x_1(t) \) is \( x_{1,1,v} \) and ... and \( x_{np,v} (t) \) is \( x_{np,v} (t) \)

THEN \( \dot{x}(t) = (A - C_r)x(t) + B(d)u(t) + w(t) \quad \nu = 1, 2, \ldots, r \)

where \( x_{(i,j), v} \) is the fuzzy set, \( C_r \), is equal to \( C \cdot z_{v} \), \( z_{v} = \text{diag}([z_{(1,1,v)}, \ldots, z_{np,v}]^T) \) can be calculated in each rule according to its definition in Eq. (5), \( r \) is the number of model rules.

The final state of the fuzzy system is inferred as follows:

\[
\dot{x}(t) = \sum_{\nu=1}^{r} h_{\nu}(x(t))[A - C_r]x(t) + B(d)u(t) + w(t)]
\]

where

\[
\mu_{\nu}(x) = \prod_{w=1}^{np} M_{(w,w),v}(x_w(t))
\]

The state-space model of each rule in the fuzzy model (8) describes the process behavior around a small neighborhood of a nominal operating point. Since there are many rules that represent the different nominal operating points, the fuzzy model (9) can approximate an arbitrary nonlinear process in a large working region.

### 3. Fuzzy control design

The robust tracking performance under external disturbance may be achieved by using the following \( H_{\infty} \) performance:

\[
\int_{0}^{T_f} \left[ e^T(t)Qe(t) \right] dt \leq e^T(0)P_0e(0) + \frac{1}{\gamma^2} \int_{0}^{T_f} \hat{w}^T(t) \hat{w}(t) dt
\]

where \( e(t) = x_r(t) - x(t) \) is the tracking error and \( T_f \) is the end time, and \( 0 \leq Q = C_r^T C_r \in R^{np \times np} \) is the weighting matrix that are specified beforehand according to the design purpose, \( 0 \leq P = R^{np \times np} \) is an unknown matrix, \( \hat{w}(t) \) is equal to \( [w(t) \ x_r(t) \ x_1(t)]^T \), \( \gamma > 0 \) is an attenuation level. The performance index (10) means that the effect of any nonzero \( \hat{w}(t) \) on tracking error \( e(t) \) must be attenuated below a desired level \( \gamma \) from the viewpoint of energy. In general, it is desirable to make \( \gamma \) as small as possible to achieve the optimal disturbance attenuation performance.

Given the fuzzy model (9), fuzzy control is a natural selection to obtain the system stability and robustness because both of them can be designed using the same fuzzy theory. The controller structure is shown in Fig. 3.

![Fuzzy model-based fuzzy control](image)
developed as

\[ u(t) = K_v e(t) \]

where \( K_v \in \mathbb{R}^{1 \times np} \) is the controller gain.

Obviously, the fuzzy control law may be represented by

\[ u(t) = \sum_{i=1}^{r} h_i(x(t))K_v e(t) \tag{11} \]

Inserting the control law (11) into (9), the closed-loop fuzzy system may be written as

\[ \dot{\hat{v}}(t) = \sum_{i=1}^{r} \sum_{w=1}^{r} h_i(x(t))h_w(x(t)) \left[ (A - C_v)(A - C_v)^T - \bar{B}(d)K_w \right] e(t) \]

\[ - (A - C_v)x_v(t) - w(t) \right] + \hat{x}_v(t) \tag{12} \]

Choose a Lyapunov function candidate as

\[ V(t) = e^T(t)P e(t) \tag{13} \]

Calculate the derivative of \( V(t) \) along the trajectory of the system (12) and yield:

\[ \dot{V}(t) = \sum_{i=1}^{r} \sum_{w=1}^{r} h_i(x(t))h_w(x(t)) \left[ e^T(t)(P\bar{A} + \bar{A}^T)P e(t) \right] \]

\[ - \bar{A}_v(t) - (A - C_v)(A - C_v)^T \dot{e}_v(t) - e^T(t)P(A - C_v)x_v(t) - w(t))Pe(t) \]

\[ - e^T(t)Pw(t) + \bar{A}_v(t)e^T(t)Pe(t) + e^T(t)P \dot{x}_v(t) \tag{14} \]

With

\[ \bar{A}_v = A - C_v - B(d)K_w \tag{15} \]

Adding and subtracting some terms on the right side of the equality (14), the equality (14) can be rewritten as

\[ \dot{V}(t) = \sum_{i=1}^{r} \sum_{w=1}^{r} h_i(x(t))h_w(x(t)) \left[ e^T(t)(P\bar{A} + \bar{A}^T)P e(t) \right] \]

\[ - \gamma^2 (x_v(t) + \gamma^{-2}(A - C_v)(A - C_v)^T \dot{e}_v(t)) \]

\[ + \gamma^2 \bar{A}_v(t)x_v(t) + \gamma^{-2}e^T(t)P^T(A - C_v)(A - C_v)^T \dot{e}_v(t) \]

\[ - \gamma^2 (w(t) + \gamma^{-2}Pe(t)) - \gamma^2 w^T(t)w(t) \]

\[ + \gamma^{-2}e^T(t)(P^TPe(t)) - \gamma^2 (x_v(t) - \gamma^{-2}Pe(t)) \]

\[ - \gamma^{-2}Pe(t)) + e^T(t)(P^TPe(t) \]

\[ \leq \sum_{i=1}^{r} \sum_{w=1}^{r} h_i(x(t))h_w(x(t)) \left[ e^T(t)(\zeta(t) e(t) + \gamma^2 \bar{A}_v(t) \dot{e}(t)) \right] \tag{16} \]

where \( \zeta(t) = P\bar{A} + \bar{A}^T + \gamma^{-2}P^T(A - C_v)(A - C_v)^T \bar{A}_v(t) + 2\gamma^{-2}P^TPe(t) \).

Obviously, if the following inequality holds:

\[ \zeta(t) < -Q \tag{17} \]

then, from the inequality (16), we get

\[ \dot{V}(t) \leq -e^T(t)Qe(t) + \gamma^2 \bar{A}_v(t) \dot{e}(t) \]

\[ \bar{A}_v(t) \dot{e}(t) \tag{18} \]

Therefore, we have the following theorem:

**Theory 1.** Consider the curing oven model (6) and the fuzzy model (9). Given the fuzzy controller (11), if there exists a common matrix \( P > 0 \) satisfying:

\[ \left[ \begin{array}{c} (A - C_v)X + X^T(A - C_v)^T - B(d)Y_w - Y_w^T B^T(d) \\ + \gamma^{-2}(A - C_v)(A - C_v)^T + 2\gamma^{-2}I \end{array} \right] X^T - Q^{-1} \] \[ < 0 \tag{19} \]

With

\[ X = p^{-1}, Y_w = K_w X. \]

Then the closed-loop model (12) is exponentially stable in the absence of \( \dot{w}(t) \) and the \( H_\infty \) control performance (10) is guaranteed in the presence of \( \dot{w}(t) \).

**Proof.** See Appendix B. \[ \square \]

In order to obtain the optimal disturbance attenuation performance, bound of the \( H_\infty \) tracking performance \( \int_0^T |e^T(t)Qe(t)| \) dt in (10) should be minimized, which can be achieved by solving the following constraint optimization problem:

\[ \min_{X, Y_1, \ldots, Y_r} \gamma^2 \]

subject to \[ \gamma > 0, X > 0, \text{ the inequality } (19) \tag{20} \]

where the inequality (19) is the requirement of the stability. This optimization problem (20) can be solved by LMI, and its solution can guarantee that the system is stable and has a good robust tracking performance. After obtaining \( X, Y_1, \ldots, Y_r \) from (20), the controller gain is expressed as

\[ K_w = Y_w X^{-1} \tag{21} \]

### 4. Integration of design and control and its optimization method

#### 4.1. Integration of design and control

The integration cost \( J(x, d, u) \) which simultaneously considers the design performance and the control performance can be formulated as

\[ \min_{d, x} J(x, d, u) \]

s.t. \[ \dot{x}(t) = Ax(t) + C_0(e(t) + B(d)u(t) + w(t) \text{ (Process model)} \]

the equalities (11) and (21) \text{ (Controller design)}

the inequality (19) \text{ (Stability)}

\[ \gamma < \gamma_0 \text{ (Robust performance)} \]

\[ l(x, u, d) \leq 0 \text{ (Other constraint)} \]

where equalities (11) and (21) are the control rule and the control gain respectively, the inequality (19) is the requirement of the stability, the attenuation level \( \gamma \) less than \( \gamma_0 \) (\( \gamma < \gamma_0 \)) can guarantee the robust performance, \( \gamma_0 \) is a specified tolerance that is decided by users in proportion to the objective’s relative importance in the context of the problem, \( l(x, u, d) \) is constraint from other design aspects. The integration optimization (22) can achieve the desirable tracking performance, and guarantee the system stability, robustness and flexibility as well.

#### 4.2. PSO-based intelligent optimization method

Obviously, the integration (22) is a nonlinear constrained optimization problem. The detection of the controller parameters and the design parameters with mathematical programming algorithms is often divergent due to discontinuities and non-convergencies in this integration problem space [35]. Thus, it is very complex and difficult to optimize design parameters and fuzzy controller gains together using analytical optimization methods.

In order to make the integration problem easier to tackle, a hierarchical optimization framework proposed by Malcolm et al.
4.2. Embedded control optimization

Given the design parameters, the embedded control optimization (inner loop) is to guarantee the stability and robustness. This optimization strategy is shown in Fig. 4, and summarized as follows:

1. Initialization (Given design parameters)
2. Process model (Eq. 6)
3. Fuzzy modeling (Eq. (9))
4. Model verification
5. Satisfactory approximate error?
   - Yes
     a. Fuzzy controller design (Eq. (20) and (21))
     b. Integration cost (22)
   - No
     a. Embedded control optimization
     b. PSO based design (Eq. 24 and Eq. 25)

Fig. 4. Embedded control optimization.

4.3. Master design optimization

4.3.1. Design framework

This master design optimization (outer loop) as shown in Fig. 5 is developed to find the optimal design parameters by solving the integration problem (22), where the controller gains, the control input \( u \) and the attenuation level \( \gamma \) have been decided in the embedded control optimization.

Usually, this optimization problem is complex and difficult to be solved by analytical methods since it may be non-convex or non-differential. Here, an intelligent PSO method is proposed to solve this master optimization.

4.3.2. Particle swarm optimization based optimal design

In PSO, each individual possible solution can be modeled as a particle that moves through the problem hyperspace. The position of the \( i \)th particle at a certain iteration time \( t \) is determined by the vector of the coordinates:

\[
d_i(t) = [d_{i1}(t), \ldots, d_{im}(t)]
\]

where the component \( d_{ij} \in [d_{jmin}, d_{jmax}] \) represents the \( j \)th design parameter of the \( i \)th particle, \( \beta \) is the total number of design parameters namely the dimension of the search space.

The information available for each individual is based on its own experience and the knowledge of other individuals. Since the relative importance of these two factors can vary from one decision to another, it is reasonable to apply random weights to each part. Therefore, the velocity \( v_i(t) \) of the \( i \)th particle will be determined by

\[
v_i(t) = \Delta v_i(t-1) + \varphi_1 \cdot \text{rand}_1 \cdot (p_{\text{best},i}(t-1) - d_i(t-1)) + \varphi_2 \cdot \text{rand}_2 \cdot (g_{\text{best}}(t-1) - d_i(t-1))
\]

where \( \Delta v_i(t) \) is an inertia weight, \( \varphi_1 \) and \( \varphi_2 \) are two positive numbers and \( \text{rand}_1, \text{rand}_2 \) are two random numbers with uniform distribution in the range of \([0,1]\). \( p_{\text{best},i} \) and \( g_{\text{best}} \) is the best position of the \( i \)th particle itself and the best position of the entire swarm respectively.

Then, the position of the \( i \)th particle is updated according to

\[
d_i(t+1) = d_i(t) + v_i(t)
\]

when the value of a coordinate of a particle lies outside the interval of acceptable values, it means that the particle leaves the search space. The velocity and position of such a particle should be modified to bring it back inside the search space [32]:

\[
d_{ji} \notin [d_{jmin}, d_{jmax}] \Rightarrow \begin{cases} v_{ji}(t) = 0 \\ d_{ji} < d_{jmin} \Rightarrow d_{ji} = d_{jmin} \\ d_{ji} > d_{jmax} \Rightarrow d_{ji} = d_{jmax} \end{cases}
\]

where \( v_{ji}(t) \) is the \( j \)th element of \( v_i(t) \).

The following procedure can be used for implementing the PSO algorithm [31]:
Step 1: Initialize the swarm by assigning a random position in the problem hyperspace to each particle.

Step 2: Evaluate the fitness function \( J(x, d, u) \) for the \( i \)th particle and judge the feasibility of the \( i \)th particle. The feasibility test is to check all constraint conditions when given the design variable \( d \). If all constraint conditions are feasible, then such a particle will be accepted. Otherwise, this particle is unaccepted and then the program goes to step 5.

Step 3: For each accepted particle, compare the particle’s fitness value \( J(x, d, u) \) with its \( p_{best,i} \). If the current value is better than \( p_{best,i} \) value, then set this value as \( p_{best,i} \) and the current particle’s position \( d_i \) as \( p_i \).

Step 4: Identify the particle with the best fitness value. It is to compare the integration function \( J(x, d, u) \) of all particles with \( g_{best} \) in order to find the minimal integration function, which is regarded as the best fitness value. Then set this best fitness value as \( g_{best} \) and its corresponding position as \( p_g \).

Step 5: Update the velocities and position of all the particles using Eqs. (24) and (25).

Step 6: Repeat steps 2–5 until a stopping criterion is met. Finally, the optimal design parameters are taken from \( p_g \).

### 4.4. Summary

The proposed integration of design and control is shown in Fig. 6 with two optimizations below.

1. **Embedded control optimization (inner loop):**
   - Under the given design parameters, this optimization is to guarantee the stability and robustness of the system. It mainly includes two key parts:
     - A fuzzy model is first developed to approximate the original nonlinear process in a large operating region.
     - Based on this fuzzy model, a fuzzy controller is derived to stabilize the system and minimize the robust tracking performance.

2. **Master design optimization (outer loop):**
   - This optimization is to obtain the optimal design parameters by solving the integration problem (22). An intelligent PSO method is proposed to solve this integration problem. Even if it is non-convex or non-differential, this integration problem can be effectively solved by the intelligent PSO method.

Since the control optimization is embedded into the master design optimization, successive iterations of the master design problem will gradually improve the integration performance. Since the PSO-based design is integrated with the fuzzy modeling/control, the proposed method has combined the merits of both fuzzy modeling/control and PSO. Thus, it has ability to deal with the complex nonlinear problem in a large operating region. Its solution can achieve the tracking performance, and guarantee the stability, robustness and flexibility as well.

### 5. Case study

The parameters of the curing system are shown in Table 1. The length, breadth and height of the lead frame are 240 mm, 90 mm and 0.2 mm respectively. The lead frame is uniformly divided into 36 zones.

The desired temperature profile (°C) is given as

\[
T_i(t) = \begin{cases} 
40 + 7t & \text{for } 0 \leq t \leq 20 \\
180 & \text{for } 21 \leq t \leq 50 
\end{cases}
\]  

(27)

Moreover, the desirable temperature uniformity needs to guarantee that the steady-state error \( e_{ij}(t_f) \) defined as \( e_{ij}(t_f) = T_i(t_f) - T_{ij}(t_f) \) should be limited in ±5 °C. Thus, the design constraint \( l(x, u, d) \) is considered as

\[
-5 \leq e_{ij}(t_f) = T_i(t_f) - T_{ij}(t_f) \leq 5
\]

(28)

The integration cost \( J(x, d, u) \) and the unknown disturbance are

\[
J(x, d, u) = \int_0^{t_f} e^T(t)e(t)dt
\]

(29a)

\[
\hat{w}_{ij}(t) = 2\cos(t)
\]

(29b)

The objective is to design a fuzzy controller and optimize the design parameters \( \theta \) and \( H \) from \( \theta \in (160°, 180°) \) and \( H \in (4 \text{ mm, } 14 \text{ mm}) \) to obtain the satisfactory steady and dynamic performance.

#### 5.1. Proposed integration method for nonlinear curing process

Since the curing oven is fully symmetric about \( x \) and \( y \) axis, only a quarter of the LF is required to use for design and control. The nonlinear curing model with uncertainty is firstly approximated

| \( c \) Density, \( \rho \) | 385 J/(kg K) | 8780 kg/m³ |
| \( \varepsilon \) | 0.2 |
| \( b \) | 90 mm |
| \( l \) | 240 mm |

Table 1 Parameters of the curing system.
by the fuzzy model (9). The membership function $h(x(t))$ for the fuzzy sets of $x(t)$ is shown in Fig. 7.

The fuzzy T–S controller (11) is used to control the fuzzy model (9) and twenty particles are employed for PSO. The velocity and the position of the particle $v_i$ is determined by (24) and (25) with $\delta = 0.7$, $\varphi_1 = \varphi_2 = 1.47$, which is suggested by [31]. The iteration process is shown in Fig. 8.

The optimal design parameters $d_I^* = [166^\circ, 4.5\text{ mm}]$ by the proposed integration method under the weighting matrix $Q = 0.1 I$. The controller gains under the optimal design parameters $d_I^*$ are listed as follows:

$$K_1 = [20.6532 \ 20.7640 \ 20.7147 \ 21.2814 \ 21.3480 \ 21.3938 \ 21.3532 \ 21.4510 \ 21.3317]$$

$$K_2 = [20.2428 \ 20.3526 \ 20.3033 \ 20.8615 \ 20.9271 \ 20.9734 \ 20.9323 \ 21.0295 \ 20.9072]$$


$$K_5 = [20.5347 \ 20.6452 \ 20.5959 \ 21.1601 \ 21.2265 \ 21.2724 \ 21.2316 \ 21.3293 \ 21.2091]$$

5.2. Verification and comparison

The robust tracking performance and the uniform temperature performance obtained by the proposed integration method are shown in Fig. 9. From Fig. 9(a), the maximal and minimal temperature trajectory on the whole surface of the LF are very close to the reference signal. From Fig. 9(b), all steady-state errors $e_{i,j}(t_f)$ are limited in $\pm 3^\circ \text{C}$, which satisfies the requirement of the temperature uniformity. Thus, the proposed integration method can obtain the desirable robust tracking performance and uniform temperature distribution.

For a better comparison, two design methods are selected as follows: (i) the traditional sequential design method and (ii) the proposed integration method.

The traditional sequential design method is to firstly obtain the design parameters by solving the following steady-state temperature uniform design problem [2,5]:

$$\min_d \left( \frac{1}{36} \sum_{i=1}^{6} \sum_{j=1}^{6} (T_i(t_f) - T_{i,j}(t_f))^2 \right)^{1/2}$$

The optimal design variables calculated by the traditional method is $d_T^* = [166^\circ, 4.5\text{ mm}]$. Then, the controller is obtained by solving the control design problem (20).

Some performance indexes are set up for an easy comparison as follows:

- Spatio-temporal error:
  $$e_{i,j}(t) = T_i(t) - T_{i,j}(t)$$

- Root of mean squared error:
  $$\text{RMSE} = \left( \frac{1}{36 \times 50} \sum_{i=1}^{6} \sum_{j=1}^{6} e_{i,j}(t)^2 \right)^{1/2}$$

- Spatial normalized absolute error:
  $$\text{SNAE}(t) = \frac{1}{36} \sum_{i=1}^{6} \sum_{j=1}^{6} |e_{i,j}(t)|$$

Fig. 9. Robust tracking performance and uniform temperature distribution.

![Fig. 8. Iterative process of PSO.](image)
The difference of time normalized absolute error is defined as follows:

\[ TNAE(i, j) = \frac{1}{50} \sum_{i=1}^{50} |e_i,j(t)| \]

• Difference of time normalized absolute error:

\[ DTNAE(i, j) = TNAE_t(i, j) - TNAE_p(i, j) \]

where \( TNAE_t \) and \( TNAE_p \) are the time normalized absolute error obtained by the traditional sequential method and the proposed integration method respectively.

The SNAE and DTNAE are shown in Fig. 10. From 10(a), it shows that the SNAE of the integration method is much smaller than the traditional sequential method when the process works in the dynamic environment (from 0 s to 20 s). This comparison shows that the integration method can improve the dynamic performance and keep a better steady-state performance than the traditional sequential method.

From 10(b), since all DTNAE are larger than zero, the integration method has a smaller TNAE than the traditional sequential method. This comparison shows that the integration method can obtain a better performance than the traditional sequential method.

Finally, RMSE are compared for both methods, with 4.5851 for the proposed integration method and 5.2076 for the traditional sequential method. It is clear that the proposed integration method has a better RMSE than the traditional method.

Thus, from the above comparisons, it is clear that the proposed integration method has a better performance than the traditional sequential method.

6. Conclusion

A novel intelligent integration of design and control is proposed for one kind of nonlinear curing process. In the proposed integration method, fuzzy-modeling method can very well approximate the non-convex and non-differential integration problem. Since the PSO-based design is integrated with the fuzzy modeling/control, the proposed method has combined the merits of both fuzzy modeling/control and PSO. Thus, it has ability to deal with the complex nonlinear problem with uncertainty in a large operating region, and can solve the non-convex and non-differential integration problem. Its solution can achieve the desirable steady-state and dynamic performances.

The proposed integrated method is applied to design and control a nonlinear curing process. The results show that this proposed method can obtain the desirable steady-state and dynamic performances. Furthermore, the comparisons between the proposed method and the traditional sequential method are studied on controlling the temperature profile of this nonlinear curing process. These comparisons demonstrate that the proposed integration method has a better dynamic performance as well as the steady-state performance than the traditional sequential method.

Acknowledgements

The project is partially supported by a project from RGC of Hong Kong (CityU: 117208), a project from City University of Hong Kong (7002450), a project from NSF China (50775224), and the National Basic Research Program (973) of China under Grants 2006CB705404 and 2005CB724102.

Appendix A.

The matrices \( B(d) \) and \( C \) are described as

\[
B(d) = \begin{bmatrix}
\frac{F_{1,1}(d)S_{1,1}}{m_{1,1}c} & \ldots & \frac{F_{n,p}(d)S_{n,p}}{m_{n,p}c}
\end{bmatrix}^T,
\]

\[
C = \text{diag} \left( \begin{bmatrix}
\frac{\alpha S_{1,1}}{m_{1,1}c} & \ldots & \frac{\alpha S_{n,p}}{m_{n,p}c}
\end{bmatrix}^T \right)
\] (A.1)

Since the view factor \( F_{i,j}(d) \) is the function of design parameters vector \( d \) as presented in [29], the matrix \( B(d) \) is also the function of design parameters vector \( d \).

The element of \( A \) is defined as follows, with the identical meaning of \((i, j)\) as Fig. 2.

If \( i = 1, j = 1 \):

\[
A_{1,1} = \frac{kS_x}{m_{1,j}c} \Delta x - \frac{kS_y}{m_{1,1}c} \Delta y,
A_{1,2} = \frac{kS_y}{m_{1,j}c} \Delta y,
A_{1,1+n} = \frac{kS_x}{m_{1,j}c} \Delta x
\]

If \( i = 1, 2 \leq j \leq p - 1 \):

\[
A_{j,j} = \frac{kS_x}{m_{1,j}c} \Delta x - \frac{kS_y}{m_{1,j}c} \Delta y,
A_{j,j-1} = A_{j,j+1} = \frac{kS_y}{m_{1,j}c} \Delta y,
A_{j,j+p} = \frac{kS_y}{m_{1,j}c} \Delta x
\]
If \( i = 1, j = p \):

\[
A_{j} = -\frac{kS_x}{m_{ij,c}} \Delta x - \frac{kS_y}{m_{ij,c}} \Delta y,
A_{j+1} = \frac{kS_y}{m_{ij,c}} \Delta y
\]

If \( 2 \leq i \leq n - 1, j = 1 \):

\[
A_{(i-1)n+j, (i-1)n+j-1} = -\frac{kS_x}{m_{ij,c}} \Delta x - \frac{kS_y}{m_{ij,c}} \Delta y
\]

\[
A_{(i-1)n+j, (i-1)n+j} = \frac{kS_y}{m_{ij,c}} \Delta y
\]

\[
A_{(i-1)n+j, (i-1)n+j+1} = -\frac{kS_x}{m_{ij,c}} \Delta x
\]

\[
A_{(i-1)n+j, (i-1)n+j+p} = A_{(i-1)n+j, (i-1)n+j-1} - \frac{kS_x}{m_{ij,c}} \Delta x - \frac{kS_y}{m_{ij,c}} \Delta y
\]

Otherwise, all elements of \( A \) are equal to zero.

**Appendix B.**

Assume that the inequalities of (17) holds and \( P > 0 \). From the inequality (17) and the equality (15), we have

\[
P(A - C_v - (d)K_w) + (A - C_v - (d)K_w)^T P
+ \gamma^2 P(\nu - C_v)(A - C_v)^T P + 2\gamma^{-2} P \gamma^2 Q < 0
\]

Pre- and post-multiplying \( X^T \) and \( X \) with \( P = X^{-1} \):

\[
(A - C_v - (d)K_w)X + X^T(A - C_v - (d)K_w)^T
+ \gamma^{-2}(A - C_v)(A - C_v)^T + 2\gamma^{-2} I + X^T QX < 0
\]

Let \( Y_w = K_wX \). Then the inequality (B.2) can be rewritten as

\[
(A - C_v)X + X^T(A - C_v)^T - B(d)Y_w - Y_w^T B^T(d)
+ \gamma^{-2}(A - C_v)(A - C_v)^T + 2\gamma^{-2} I + X^T QX < 0
\]

The inequality (B.3) may be transformed into LMI form:

\[
\begin{bmatrix}
(A - C_v)X + X^T(A - C_v)^T - B(d)Y_w & Y_w^T B^T(d) \\
Y_w & \gamma^{-2}(A - C_v)(A - C_v)^T + 2\gamma^{-2} I + X^T QX
\end{bmatrix} < 0
\]

Thus, the stable condition (17) is transformed into (19) (the same with (B.4)), which can be solved by LMI.

Thus, if (19) is satisfied, the inequality (18) will hold. Integrating (18) from \( t = 0 \) to \( y \) yields:

\[
V(t) - V(t_0) \leq -\int_{t_0}^{t_f} e^T(t)Qe(t)dt + \gamma^2 \int_{t_0}^{t_f} \tilde{w}^T(t)\tilde{w}(t)dt
\]

Since the Lyapunov function \( V(t) > 0 \), the inequality (B.5) is rewritten as

\[
\int_{t_0}^{t_f} e^T(t)Qe(t)dt \leq V(t_0) + \gamma^2 \int_{t_0}^{t_f} \tilde{w}^T(t)\tilde{w}(t)dt
\]

Thus, we can get the \( H_\infty \) tracking performance (10) from (B.6).

In addition, when \( \tilde{w}(t) = 0 \), the inequality (18) may be written as

\[
\tilde{V}(t) \leq -e^T(t)Qe(t) \leq -\lambda_{\min}(Q)e^T(t)e(t) \leq -\lambda_{\min}(Q)/\lambda_{\max}(P) V(t)
\]

From the above inequality (B.7), we have

\[
V(t) \leq V(0)e^{-(\lambda_{\min}(Q)/\lambda_{\max}(P))t}
\]

so that \( \|e(t)\| \leq \sqrt{\lambda_{\max}(P)/\lambda_{\min}(Q)e^{-(\lambda_{\min}(Q)/\lambda_{\max}(P))t}\|e(0)\|} \) for all trajectories. Therefore, the close-loop fuzzy system (12) with \( \tilde{w}(t) = 0 \) is exponentially stable.

**References**


