

Integrated Optimization Framework for Composite Manufacturing: Minimizing Spring-In Effects and Improving Cure Cycles

RYAN ENOS, DIANYUN ZHANG* and QINGXUAN WEI

ABSTRACT

This study presents a structured methodology for the optimization of composite manufacturing processes, focusing specifically on autoclave techniques to mitigate issues such as the spring-in effect. The initial phase employs Sequential Quadratic Programming (SQP) in a weighted-sum approach to optimize the cure cycle, using Radford's equation for spring-in angle estimation and a multi-physics, multi-scale MATLAB model to investigate the cure and temperature-dependent laminate response. This phase underscores the efficacy of the selected optimization algorithm, demonstrating a significant reduction in spring-in while ensuring a high degree-of-cure. Subsequently, the study incorporates an integrated Finite Element Analysis (FEA) optimization framework linking ABAQUS and MATLAB. This framework utilizes the Non-dominated Sorting Genetic Algorithm II (NSGA-II) for multi-objective optimization with an integrated composite manufacturing processing model. This second phase illustrates the framework's robust capabilities in composite cure cycle optimization, providing a well-distributed set of optimal solutions in an efficient timeframe. The study highlights the potential of the approaches and frameworks investigated to improve the efficiency, performance, and quality of composite parts.

Ryan Enos, Purdue University, 610 Purdue Mall, West Lafayette, IN 47907, U.S.A.
Dianyun Zhang, Purdue University, 610 Purdue Mall, West Lafayette, IN 47907, U.S.A.
Qingxuan Wei, Purdue University, 610 Purdue Mall, West Lafayette, IN 47907, U.S.A.
*Corresponding Author: Dianyun Zhang—dianyun@purdue.edu

INTRODUCTION

Composite materials are increasingly becoming indispensable in the production of high-performance structures, largely due to their exceptional properties. They exhibit a high strength-to-weight ratio, offer design flexibility, and provide fatigue and corrosion resistance, among other attributes. However, manufacturing processes for these composites can be challenging, especially due to effects such as spring-in, which predominantly occurs as a result of combined thermal and chemical shrinkage during the curing process.

Despite the advancement in high fidelity curing-stress-deformation models, it has become increasingly clear that these are not sufficient in isolation. While such models are vital for the successful understanding and prediction of composite part behavior, we are often inundated with a multitude of factors that influence the composite's complex response during curing. Consequently, we can leverage curing simulation to investigate these effects, yet, turning this information into tangible improvements in the manufacturing process is still a considerable challenge.

To effectively harness the power of these sophisticated models, additional tools and methods are needed. This recognition leads us to integrate techniques such as statistical analysis, uncertainty quantification, and optimization into our framework. These approaches are not only complementary to the models but also crucial in facilitating an informed decision-making process in composite manufacturing.

The purpose of this study is to address these challenges through a two-phase effort in developing a versatile optimization approach for composite manufacturing processes. The benchmark for this study is an L-shaped composite part, which is a common feature in composite structures. With its inherent susceptibility to the spring-in effect, the L-shaped composite serves as a significant representative of the extent of distortion that more intricate components may experience during the manufacturing process.

In this study, our optimization approach is geared towards minimizing the spring-in effect, maintaining high degree of cure (DOC), and ensuring a reasonable processing time. We underscore that our ultimate objective is not to merely optimize cure cycles for these specific targets, but rather to establish a methodology that can be broadly applied across different facets of composite manufacturing.

EFFICIENT CURING-STRESS-DEFORMATION MODEL

This study aims to optimize composite cure cycles to minimize the spring-in effect, maintain high degrees of cure (DOC), and ensure reasonable processing time. To achieve this, the capability to model the curing of the composite, and the effect of curing on the stress-deformation response is required. A curing model typically integrates essential inputs including material properties, cure cycle parameters, heat transfer conditions, and geometric information, to generate outputs encompassing the degree of

cure, thermal history, chemical shrinkage, material properties, and potential residual stresses or distortions of the composite structure.

Resin properties are viscoelastic and change during curing. The idea of the cure hardening instantaneously linear elastic (CHILE) model is to simplify changes in properties and viscoelastic behavior by eliminating information that is not significant to the results of interest. Our CHILE model is physics based and the material input parameters are the fiber and matrix properties, including the cure kinetics parameters. In this study, we considered the thermoset resin system EPON 862/W, for which the cure kinetics can be expressed using the Kamal-Sourour autocatalytic model as [1]

$$\frac{d\phi}{dt} = \left(A_1 \exp\left(-\frac{\Delta E_1}{RT}\right) + A_2 \exp\left(-\frac{\Delta E_2}{RT}\right) \phi^m \right) (1-\phi)^n \quad (1)$$

where ϕ is the degree of cure (DOC). Based on the temperature evolution, this differential equation is solved numerically using a 4th order Runge-Kutta method to obtain the cure evolution, and this exothermic reaction in turn generates heat.

As the resin cures, it transitions from a liquid to the solid phase, the glass transition temperature simultaneously grows monotonically with DOC, which can be described using the DiBenedetto Equation [2]

$$\frac{T_g(\phi) - T_g^0}{T_g^\infty - T_g^0} = \frac{\lambda\phi}{1 - (1-\lambda)\phi} \quad (2)$$

where T_g is the glass transition temperature, T_g^0 and T_g^∞ are the glass transition temperatures of the un-cured and fully-cure resin, respectively. Detailed discussion regarding experimental characterization of the cure-dependent glass transition temperature of EPON 862/W is given in [3]. When the resin temperature is above or below T_g , it is said to be in the rubbery or glassy state, respectively.

We consider accumulation of stress-deformation after resin gelation occurs and the material is therefore no longer able to fully relax. Since the resin shows a thermo-viscoelastic response during curing, the composite cure-dependent constitutive relation becomes

$$\sigma(t) = \int_0^t Cr(t-s) \frac{d}{ds} (\varepsilon - \varepsilon^{\text{th}} - \varepsilon^{\text{ch}}) ds \quad (3)$$

where Cr is the composite relaxation stiffness matrix. The CHILE model assumes the stress analysis can be simplified by assuming that the composite relaxation modulus can be decomposed into instantaneous and time-dependent parts as [4]

$$E(t) = E_a + E_m e^{-t/\tau} \quad (4)$$

When the resin is in the rubbery phase, the relaxation time is small, and the relaxation modulus can be approximated as $E_R = E_a$. When the resin is in the glassy phase, the relaxation time is large, and the relaxation modulus can be approximated as

$E_G = E_a + E_m$. As a result, the cure-dependent composite constitutive relation can be simplified as

$$\sigma(t) = \int_0^t C_i(t) \frac{\partial}{\partial t} (\varepsilon - \varepsilon^{th} - \varepsilon^{ch}) dt \quad (5)$$

where $i = R, G$ represent the composite properties at the rubbery and glassy phases, respectively. We implement Equation (5) in incremental form as

$$\Delta\sigma = C_i(\Delta\varepsilon - \Delta\varepsilon^{th} - \Delta\varepsilon^{ch}) \quad (6)$$

where $\Delta\sigma$ is the stress increment, and $\Delta\varepsilon^{th}$ and $\Delta\varepsilon^{ch}$ are the free thermal and chemical strain increments.

Multi-scale modeling

In this study, we represent the composite laminate as a homogeneous orthotropic solid. Our curing model takes fiber and matrix properties as input material properties, and the resin properties are cure and temperature dependent, therefore, we homogenize the effective composite thermo-mechanical properties, as shown conceptually in Figure 1. The Extended Concentric Cylinder Assemblage (ECCA) micromechanics models [5] is adopted to compute the effective lamina properties, and the laminate properties are homogenized using an extended Classical Laminated Plate Theory (CLPT)-based approach to obtain the full 3D properties, including effective elastic properties, and coefficients of thermal and chemical expansion.

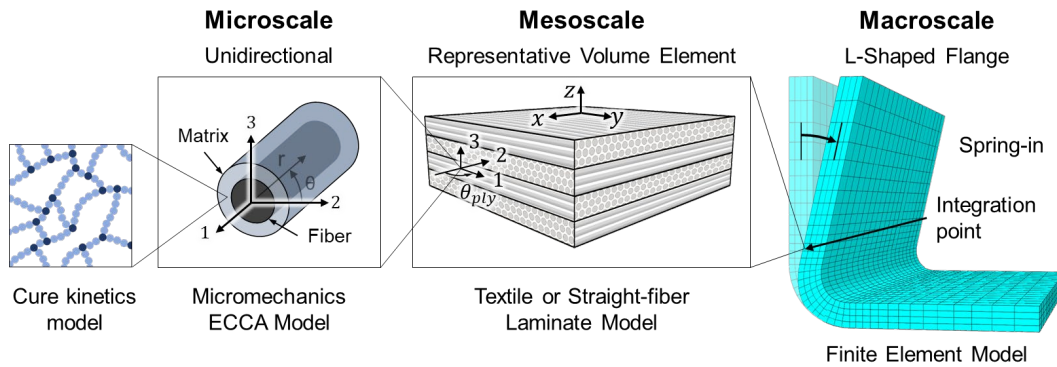


Figure 1. Conceptual diagram of multi-scale curing-stress-deformation model used in this study.

During the composite manufacturing process, thermal and chemical strain increments, used in Equation (6), accumulate due to changes in temperature and degree of cure as,

$$\begin{aligned} \Delta\varepsilon^{th} &= \alpha\Delta T \\ \Delta\varepsilon^{ch} &= \beta\Delta\phi \end{aligned} \quad (7)$$

where α and β are the homogenized coefficients of thermal and chemical expansion vectors, respectively, and ΔT and $\Delta \phi$ are temperature and cure increments, respectively. It is noted that β is a negative number as the chemical effects are manifested as cure shrinkage. In this study, we consider two implementations of the efficient curing model, the MATLAB and FEA (ABAQUS) implementations, namely, to predict the processing induced spring-in angle.

MATLAB implementation

In the MATLAB implementation, we define the temperature history directly by discretizing the cure-cycle. The temperature-cure-dependent properties are homogenized at each increment, and the thermal and chemical strain increments are obtained by Equation (7). Then, the spring-in is computed using Radford's equation as

$$\frac{\Delta \theta}{\theta} = \frac{(\alpha_{\theta} - \alpha_r)\Delta T}{1 + \alpha_r \Delta T} + \frac{(\beta_{\theta} - \beta_r)\Delta \phi}{1 + \beta_r \Delta \phi} \quad (8)$$

where the subscripts θ and r designate the azimuthal and radial directions, respectively. This allows us to obtain an estimated spring-in response without discretizing geometry or structural analysis. The advantages of this implementation are high efficiency and straightforward application to optimization, as the model is contained directly in a function, which is the conventional way to pass the objective function to an optimization algorithm.

FEA implementation

The FEA implementation utilizes a user-defined material subroutine, known as UMAT, within the ABAQUS commercial software, and offers significant flexibility by enabling tool-part interaction and heat transfer analysis through discretization. While the current study excludes heat transfer analysis for the sake of simplicity and operational efficiency, it remains an intriguing prospect for future investigation. By incorporating heat transfer analysis into the optimization framework, a more complete and accurate influence of the cure cycle can be achieved, and additional objectives defined, such as minimization of thermal gradients.

Discretization plays a crucial role in the FEA implementation as it divides the geometry into finite elements, allowing for detailed tool-part interactions to be captured. This enables a more accurate representation of the manufacturing process, considering factors such as contact, friction, and thermal effects. To measure the spring-in angle, the change of slope of nodes along the flange of the L-shaped geometry is computed following demolding. To measure the final DOC, the integration point volume average is taken.

As a first step in the optimization study, a trend insight of the cure-cycle effects on spring-in angle was pursued by designing eight cure cycles for the EPON862/W resin system. The curing process was simulated for each case and the spring-in angle was measured from each result, along with the degree of cure and processing stresses. The results of each cure-cycle were compared to identify the optimum. The results are shown

in Figure 2. Based on these preliminary results, we concluded that spring-in is highly dependent on the cure cycle, and tool-part interaction introduces added complexity.

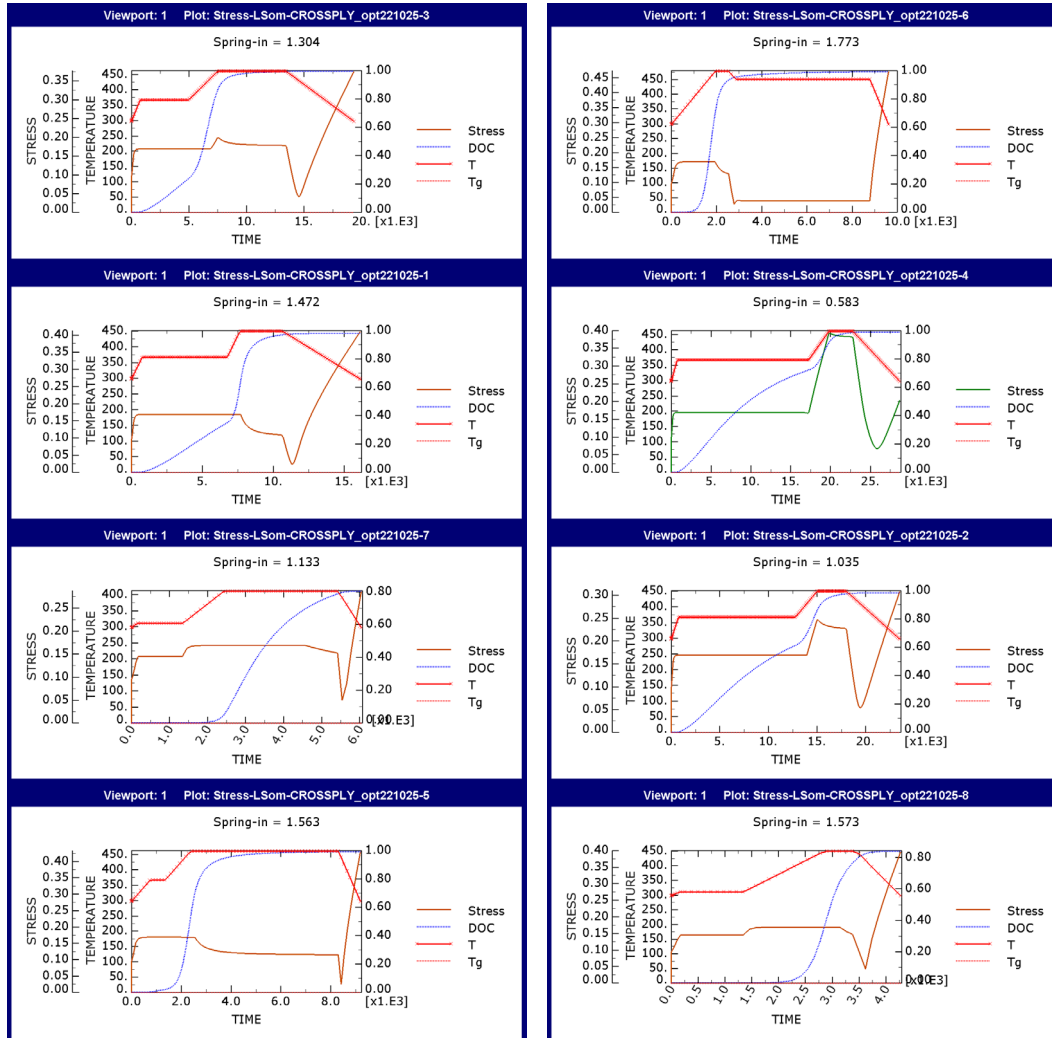


Figure 2. FEA “trend-insight” curing results including tool-part-interaction, with cure-cycle as “T”, spring-in, DOC, and internal stress history.

Experimental validation

For the purpose of experimentally validating the FEA processing model, manufacturing experimentation was conducted using an aluminum female mold and isothermal cure cycle. The experiment yielded an average spring-in angle of 1.95°. Subsequently, a corresponding simulation using the FEA model was executed with identical tooling and cure cycle as employed in the experiment. The output from the simulation closely mirrored the experimental result, yielding a spring-in angle of 1.90°. This close match supports the accuracy and applicability of the FEA model in reproducing realistic scenarios. The experimental specimen and corresponding simulation are visually represented in Figure 3 (a) and (b) respectively.

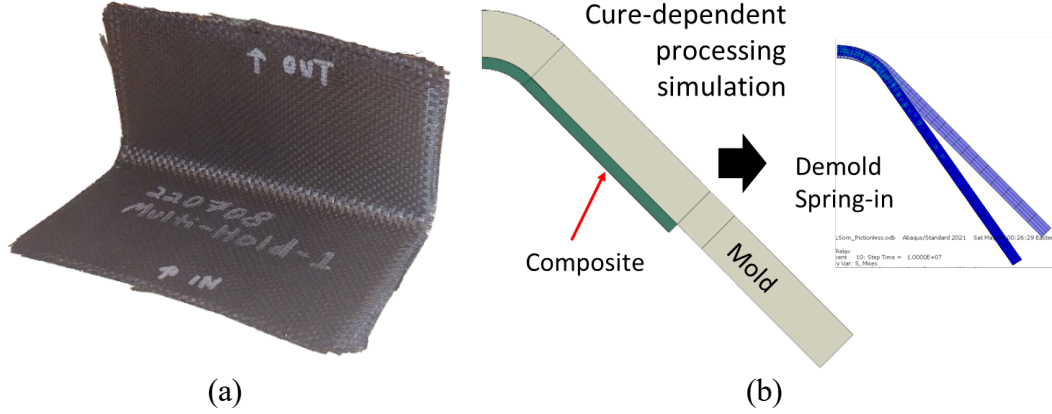


Figure 3. Composite L-shape manufactured using female tooling; (a) experiment and (b) simulation.

PHASE 1: OPTIMIZATION OF CURE CYCLE USING SQP (NON-FEA)

Problem Description

In this study, the cure cycles are optimized. The design variables are three temperature ramping rates, a_1 , a_2 , and a_3 , two temperature holds, T_1 and T_2 , together with hold durations, t_1 and t_2 , which are illustrated in Figure 4.

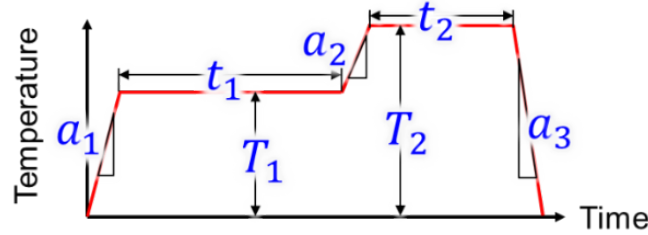


Figure 4. Illustration of cure cycles and design variables.

The objective of this study is to minimize the spring-in angle while keeping a short duration and DOC, ϕ , close to 1. They are considered as three objective functions.

$$f = r_1 f_1 + r_2 f_2 + r_3 f_3 \quad (9)$$

where

$$\begin{aligned} f_1 &= \text{abs}(\Delta\theta) \\ f_2 &= 1 - \phi \\ f_3 &= t/10000 \end{aligned}$$

where r_1 , r_2 , and r_3 are the weights of the three objective functions, and t is the total duration of cure cycles. The three functions are fully independent. When t is increased, the spring-in effect can be decreased, and the curing can be more completed. However, if t is excessive, the further decrease of the f_1 and f_2 does not worth the time cost.

Sequential Quadratic Programming (SQP) Algorithm

An in-house SQP algorithm was written in MATLAB to conduct the optimization in this study [6, 7]. The algorithm starts with finding a search direction, s , from the current point, $\mathbf{x}$. The problem itself is transformed into quadratic form as:

Minimize

$$Q(\mathbf{s}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{B} \mathbf{s} \quad (10)$$

Subject to

$$\nabla g_j(\mathbf{x})^T \mathbf{s} + \delta_j g_j(\mathbf{x}) \leq 0 \quad j = 1 \dots 8 \quad (11)$$

Here, a matrix $\mathbf{B}$ is used to approximate the Hessian matrix using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method. Initially, $\mathbf{B}$ is an identity matrix. If g_j is satisfied or linear, then $\delta_j = 1$. Otherwise, $\delta_j = 0.9$. MATLAB function “quadprog” is adopted to solve for $\mathbf{s}$. Next, the step size α is found by making use of an exterior penalty approach to approximate Lagrangian function:

$$\Phi(\alpha) = f(\mathbf{x} + \alpha \mathbf{s}) + \sum_{j=1}^8 u_j \{ \max[0, g_j(\mathbf{x} + \alpha \mathbf{s})] \} \quad (12)$$

where u_j are updated based on Lagrange Multipliers, λ_j , from the search direction results.

$$u_j = \begin{cases} |\lambda_j| & \text{for the first iteration} \\ \max \left[|\lambda_j|, \frac{1}{2} (u'_j + |\lambda_j|) \right] & \text{for subsequent iterations and} \\ u'_j & \text{are the } u_j \text{ in the previous step} \end{cases} \quad (13)$$

The MATLAB function “fminbnd” is used to compute the optimal α at the current point. The next point would be:

$$\mathbf{x}^q = \mathbf{x}^{q-1} + \alpha \mathbf{s} \quad (14)$$

Then, it is necessary to update the approximation of Hessian using BFGS method

$$\mathbf{B}^* = \mathbf{B} - \frac{\mathbf{B} \mathbf{p} \mathbf{p}^T \mathbf{B}}{\mathbf{p}^T \mathbf{B} \mathbf{p}} + \frac{\boldsymbol{\eta} \boldsymbol{\eta}^T}{\mathbf{p}^T \boldsymbol{\eta}} \quad (15)$$

where $\mathbf{p} = \mathbf{x}^q - \mathbf{x}^{q-1}$ and $\boldsymbol{\eta} = \theta \mathbf{y} + (1 - \theta) \mathbf{B} \mathbf{p}$. The vector $\mathbf{y}$ is the change of the gradient of the Lagrangian function. The scalar θ is chosen as

$$\theta = \begin{cases} 1 & \mathbf{p}^T \mathbf{y} \geq 0.2 \mathbf{p}^T \mathbf{B} \mathbf{p} \\ \frac{0.8 \mathbf{p}^T \mathbf{B} \mathbf{p}}{\mathbf{p}^T \mathbf{B} \mathbf{p} - \mathbf{p}^T \mathbf{y}} & \text{otherwise} \end{cases} \quad (16)$$

Optimal Design

The in-house SQP algorithm was used to optimize the cure cycle in the composite manufacturing process of a curved composite section. Three objectives were used, namely, the spring-in angle, the process time (total duration of the cure cycle), and the DOC. A weighted-sum approach was used to combine the multiple objectives in a single objective function, and the weights were assigned to the three terms such that they were on the same order of magnitude, and the order of importance was as follows; 1. Spring-in angle, 2. Process time, and 3. DOC. The absolute value of the spring-in angle was used, as in our case it is desired to achieve a spring-in angle as close to zero as possible. The DOC ranges from 0-1, and it is desired to be as close to 1 as possible, though values above 95% are acceptable, therefore, the weight of the DOC term was set to increase when DOC dips below 95%, similar to a penalty function. For mass production in the industry, processing time is extremely important to reduce as much as possible. The process-time term used in this study is proportional to the total duration of the cure cycle.

The manufacturer provides what is called the manufacturer recommended cure cycle (MRCC), which serves as a simple guideline for general processing, but does not account for specific design requirements or part geometry. TABLE 1 presents the design variables corresponding to the MRCC, and the resulting optimal design when using the MRCC as the initial design. The simulated manufacturing process results are presented for the initial (MRCC) design in Figure 5. (a), and the optimal design in Figure 5. (b). Here, the red curve shows the applied temperature-time history, which is dictated by the seven design variables, the dotted black curve shows the degree-of-cure development, and the blue curve shows the spring-in response. The title of each plot displays the final spring-in angle and final degree-of-cure, and the total duration of the cure cycle is the limit of the x axis, in seconds. The MRCC results for spring-in, processing-time, and DOC are 2.1°, 10800 s, and 98.8%, respectively. The optimal cure cycle results are 1.2°, 8960 s, and 97.9%, respectively, which represents a significant improvement.

TABLE 1. INITIAL AND FINAL DESIGN STARTING WITH THE MANUFACTURER RECOMMENDED CURE CYCLE.

Design	a_1 (K/min)	a_2 (K/min)	a_3 (K/min)	T_1 (K)	T_2 (K)	t_1 (min)	t_2 (min)
x_0	9.6667	1	3.2222	443.15	443.15	0	120
x^*	20	20	9.0234	403.1414	465	94.0692	29.0852

Initially, the problem was formulated as single-objective to optimize spring-in only, which resulted in many local minima or flat regions in the design space, which was not suitable for our optimization algorithm, and resulted in sensitivity to the initial design. By incorporating the process-time and DOC as additional objectives, rather than constraints, additional slope is superimposed throughout the design space, which improved the optimization performance and allowed us to produce more optimal design. After improving the formulation of our problem as weighted-sum multi-objective, the results were no longer highly sensitive to the initial design. To demonstrate this, we

randomized the initial design to produce a sequence of optimization results and find that while the initial designs vary greatly, the optimized designs are consistent, with spring-in, process-time, and DOC typically around 1.5° , 7000 s, and 98%, respectively.

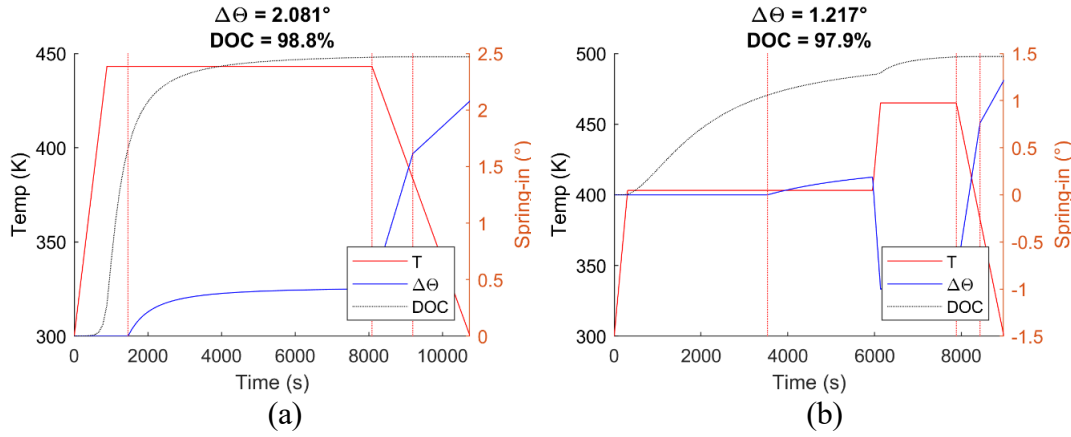


Figure 5. Manufacturing process simulation results using, (a) manufacturer recommended cure cycle as initial design, and (b) the corresponding optimized cure cycle.

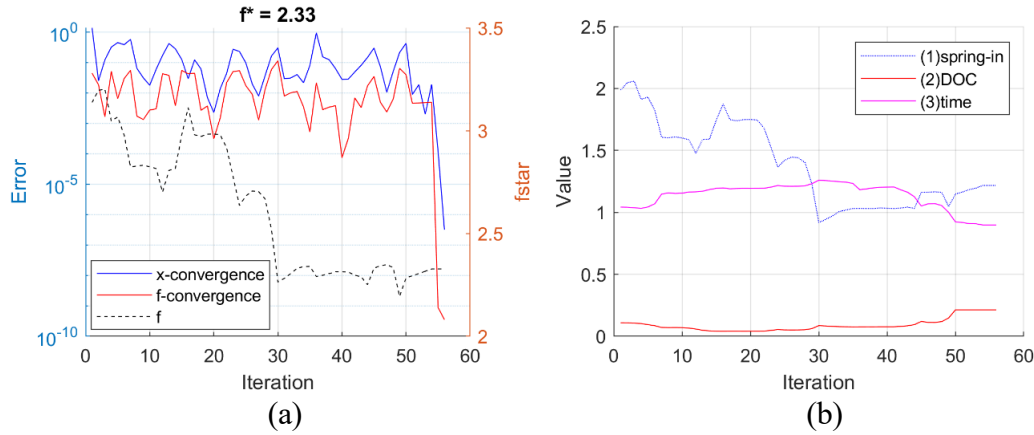


Figure 6. Performance of the optimization process; (a) convergence of the design and fitness overlaid with the objective function value, and (b) values of the individual terms in the weighted-sum multi-objective function.

PHASE 2: INTRODUCTION OF INTEGRATED FEA OPTIMIZATION FRAMEWORK USING NSGA-II

In the second phase, an integrated Finite Element Analysis (FEA) optimization framework linking ABAQUS with MATLAB is introduced. The framework features a closed-loop process involving an optimizer (MATLAB) and an evaluator (ABAQUS /Python) that interact through text files, enabling a seamless iterative optimization process. The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is employed for multi-objective optimization.

A flowchart demonstrating the FEA optimization framework is shown in Figure 7. Here, it is shown that the input to the optimizer program is the “current_fitness.csv” file,

containing the fitness of (objective values associated with) the current design. The optimizer output is the “current_design.csv” file, containing the design variables of the new generation. These design variables are the input for the evaluator program, which subsequently evaluates the fitness and stores them as output to the “current_fitness.csv” file, and the cycle repeats.

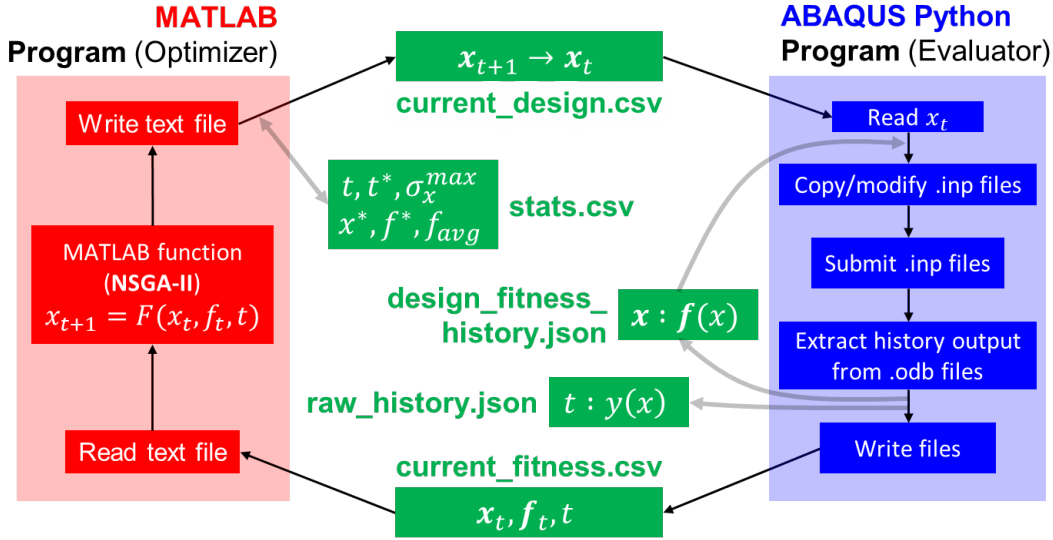


Figure 7. Flowchart of FEA optimization framework.

Optimizer program

In the context of our optimization problem, the FEA optimization framework is grounded in the principles of genetic algorithms. In our application, the cure cycle parameters, fundamental to the composite manufacturing process, are represented as evolving genes within the genetic algorithm. Each unique arrangement of these parameters forms an individual 'design' within the population. The 'fitness' of each design is evaluated based on the value of the associated objective function, thus quantifying the design's performance. Following the survival-of-the-fittest principle, the 'winning' designs, i.e., those with the highest fitness, are allowed to reproduce. The offspring produced are a random amalgamation of the 'genes', or design variables, of their parent designs, thus ensuring diversity and exploration within the design space. To further explore this design space and avoid premature convergence to a suboptimal solution, 'mutation' is incorporated, introducing random changes to the design variables, and providing an essential mechanism for the evolution and improvement of designs in our optimization framework.

NSGA-II is a multi-objective optimization algorithm that uses a genetic algorithm approach to find a set of solutions known as the Pareto front, which enables true multi-objective optimization. It balances exploration and exploitation to efficiently identify diverse and optimal solutions by employing concepts such as nondominated sorting, selection based on dominance, and genetic operators like crossover and mutation. NSGA-II, as compared to a standard GA, provides a better exploration and exploitation of the solution space due to its elitism and diversity preservation mechanisms, though it

can be computationally more demanding due to the need for additional sorting and ranking steps. The additional steps use existing fitness values to enhance solution diversity and robustness, aiming for a well-distributed set of optimal solutions without necessitating extra fitness evaluations, making it well-suited for FEA and composites manufacturing process optimization.

In the optimization algorithm, the current fitness values of the population, represented as $\mathbf{f}_t$, form the input data, which the algorithm uses to generate the next generation of individual designs, denoted as $\mathbf{x}_{t+1}$. In this context, the subscript t denotes the generation. The population comprises of $2N$ individuals, with each having n design variables, aiming to optimize M objectives. Thus, $\mathbf{x}$ is of size $2N \times n$ and $\mathbf{f}$ is of size $2N \times M$. The reason for the population size being $2N$ is tied to the sorting, ranking, and Pareto front procedures; these methods necessitate the inclusion of N parent individuals alongside their N offspring.

To initialize the optimization procedure, the algorithm generates designs randomly within the upper and lower bounds of the design variables, which make up the population and is the optimizer output. The population is written to the current-design file as shown at the top of Figure 7, containing the matrix $\mathbf{x}$ and the base-name of the optimization problem, and this file constitutes the input to the evaluator program. The optimizer then calls the evaluator and waits for it to finish before recursing. An alternative approach verified in this study is for the optimizer and evaluator to both run in the background and directly call one another.

Evaluator program

The evaluator reads the matrix $\mathbf{x}$ from the current-design file, along with the base-name, which it uses to identify the optimization problem. In our framework, the optimization problem is defined as a Python module, containing the members:

- $inp = \text{makeJob}(\mathbf{x})$: Function that makes an ABAQUS input file inp based on the individual's design, $\mathbf{x}_i$. Submission of this input file runs the corresponding simulation, producing the corresponding ABAQUS output database, odb .
- $\mathbf{y} = \text{getOutput}(odb)$: Function that extracts raw output $\mathbf{y}$ of interest from the ABAQUS simulation output odb . $\mathbf{y}$ is a dictionary that maps the names of the quantities of interest to their respective values.
- $\mathbf{f} = \text{Phi}(\mathbf{y})$: Pseudo-objective function that defines the value(s) $\mathbf{f}$ to be minimized, based on the extracted raw output $\mathbf{y}$. This method can be used to define weighted-sum multi-objective functions via scalar output or true multi-objective functions with vector output. Additionally, constraints can be implemented here as penalty functions.
- SUBROUTINE: (optional) String defining the name of a user-defined FORTRAN subroutine required by the job.

Here, each method is designed to handle a single individual from the population and define all the information needed by the evaluator program to determine the fitness of the population. The name of the module contains the problem base-name, allowing it to be imported dynamically by the evaluator based on the problem at hand. Thus, the evaluator acts as an interface between the optimization process and the ABAQUS

simulation, translating design decisions into meaningful simulations and retrieving the necessary output data to assess performance.

The input files corresponding to current-design are generated using `makeJob`, which in this case contains a base input file as a f-string and computes the input parameters from the design variables $\mathbf{x}_i$. The program writes a batch file to submit the jobs in parallel, and while the ABAQUS simulations are running, the Python evaluator program is checking for completed jobs in the background by monitoring the log files. This way, the individual outputs for completed simulations can be processed immediately while the other simulations are still running in parallel, improving the efficiency of the overall framework. When a simulation completes successfully, the evaluator program passes the output database to the `getOutput` to obtain the raw output values, which in this study include the final spring-in angle, volume average final degree of cure, and total processing time. The raw output is passed to the `Phi` pseudo-objective function to obtain the job's fitness.

The fitness data is assembled for the entire population and written to the current-fitness file along with the generation number, which is the output of the evaluator program, and the input for the optimizer program. The optimizer program now reads the fitness values for the population it had generated, iterates the generation number, and computes the next generation based on the fitnesses as described earlier, and the cycle is complete, as seen in Figure 7.

Features

Figure 7 shows three additional files utilized by the framework not mentioned thus far, namely, the statistics file “stats.csv”, design-fitness-history file “design_fitness_history.json”, and raw history “raw_history.json” file, which are not necessary to form the closed loop, but improve efficiency and usability.

The statistics file is updated each generation with the current generation number, t , the number of generations that the best design has not changed, t^* , homogeneity of current designs, σ_x^{max} , current best design, $\mathbf{x}^*$, the corresponding current best fitness, f^* , and the average fitness of the current population, f_{avg} . These optimization statistics are used to plot the optimization performance in real time, and compute the stopping criteria. Stopping criteria include maximum number of generations, average fitness approaches best fitness ($f_{avg} \rightarrow f_{best}$), designs are homogeneous across population ($\sigma_x^{max} < \text{threshold value}$), best design does not change for several generations ($t^* > \text{threshold value}$).

The purpose of the design-fitness-history is to prevent a single design from being evaluated more than once. This file maps every unique design to its corresponding fitness since the first generation. If the objective function is unchanged, the file can be re-used from previous optimizations. After reading the current-design, the evaluator checks the design-fitness-history for any of the current designs. Since NSGA-II includes parents as half of the population, at least half of the population will be in the design-fitness-history each generation. Additionally, the evaluator will check for any duplicate designs within the current generation, to ensure that it is only evaluated once.

The raw-history saves the raw output of the best design each generation in JSON format, and the purpose is to provide the user with meaningful data from the jobs, since the fitnesses may lack physical interpretation due to their role as an optimization tool.

Cure model optimization results

The multi-scale curing FEA model was integrated in the FEA optimization framework. Figure 4 illustrates the design variables, and Figure 8 (a) shows the finite element model response, respectively, for the cure-cycle optimization problem, where it is desired to minimize the spring-in angle and processing time, while maximizing the degree-of-cure. Figure 9 shows the solution history for this example, which is a real-time visualization of the information contained in the statistics file, where the dashed blue curve is the average fitness, the solid black curve is the best fitness, the red curve quantifies the variance of the designs, and the red X marks the number of generations that the best design has not changed. In this example, the stopping criteria that was met was the maximum allowable number generations that the best design does not change, which was five. In this study, the optimization framework ran 190 evaluations of the curing simulation, arriving at the optimal design in 8.8 minutes. An advantage of using a genetic algorithm in the FEA optimization framework is that the entire population of designs can be simulated (depending on computational and licensing resources) in parallel each generation, which allows for many more fitness evaluations in the same wall-clock time than could be achieved with other optimization algorithms such as SQP, which evaluates a single design at a time. The optimal results are shown in Figure 8 (b).

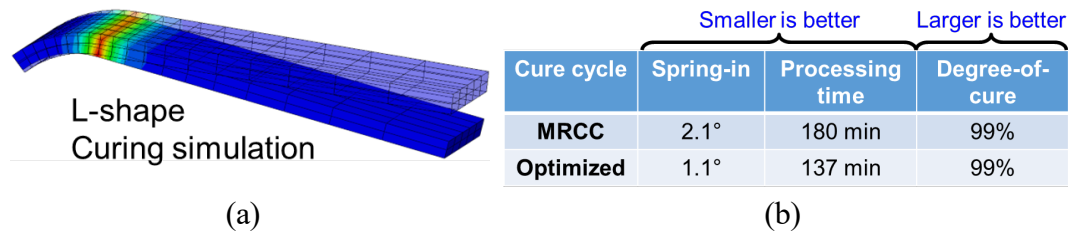


Figure 8. FEA cure-cycle optimization problem (a) stress-deformation output and (b) optimal cure-cycle compared to MRCC.

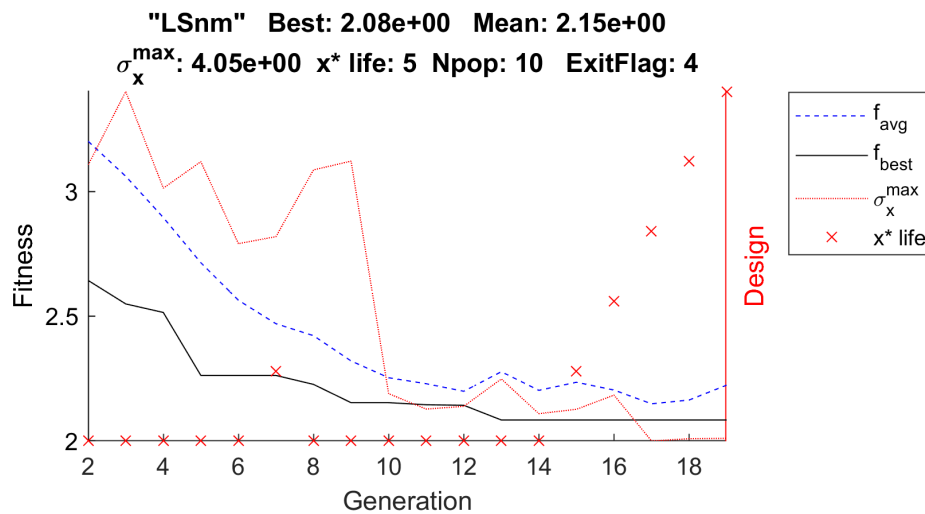


Figure 9. Optimization performance.

The MRCC results for spring-in, processing-time, and DOC are 2.1° , 10800 s, and 98.8%, respectively. The optimal cure cycle results are 1.1° , 8200 s, and 98.7%, respectively. The optimization in phase 2 further refines the manufacturing process with an additional reduction in spring-in, down to 1.1° from the initial 2.1° given by the MRCC, and a 24% decrease in processing time from the original 10800s to 8200s. While the fitness evaluation method differed from phase 1, the phase 2 result still achieved superior outcomes with a slightly lower spring-in angle and processing time, and slightly higher degree-of-cure.

FUTURE WORK

Future work will aim to expand the current optimization methodology in several directions:

1. **Experimental validation.** Fabricate and test composite samples using optimized cure cycles, comparing their spring-in angles and properties with control samples.
2. **API-based parametric model generation.** Integrate an alternative FEA job maker which relies on generating a complete ABAQUS model using a parametric Python script, rather than modifying parameters in an existing .inp file.
3. **Influence of tool design.** We find that tooling design can effect the resulting spring-in angle. With the optimization framework as it is, we can provide several tooling cases and optimize the cure cycle for each one. Alternatively, using the future API-based model generation, the geometric design of the mold could be defined parametrically and treated as design variables.
4. **Sequentially coupled analyses.** Include heat transfer by integrating the capability for sequentially coupled heat transfer and stress analyses in the optimization framework.
5. **Thermo-visco-elastic curing.** Optimize cure-cycle using visco-elastic curing model and compare with CHILE results and performance.
6. **HPC.** Integrate high-performance-computing capabilities in the optimization framework by enforcing Linux compatibility and conforming to SLURM procedures.

CONCLUSIONS

In conclusion, this study demonstrates the successful application of optimization algorithms to composite manufacturing and introduces an integrated FEA optimization framework to enhance the overall efficiency and performance of composite structures. The developed methodology and framework can be further extended to optimize other critical parameters in composite manufacturing, leading to high-quality composite parts with reduced manufacturing time.

Phase 1 of the investigation employed MATLAB and Sequential Quadratic Programming (SQP), despite their inherent limitations in capturing the complex nature of composite manufacturing. While SQP was not directly applicable to the optimization

of composite cure cycles, careful formulation of the pseudo-objective function enabled its effective use, circumventing the challenges posed by local minima in the design space. The optimization process resulted in a substantial 43% decrease in spring-in and a 17% reduction in processing time, thereby substantially bolstering manufacturing performance. Although a marginal reduction in the degree-of-cure was observed, it remained at a high level, preserving the requisite properties of the composite.

Phase 2 of the study marked a shift to a more comprehensive FEA optimization framework that effectively leveraged the capabilities of MATLAB and ABAQUS, deploying NSGA-II for multi-objective optimization tasks. With the integration of an efficient composite manufacturing processing model, the framework demonstrated the ability to efficiently optimize cure cycles within a span of just 8.8 minutes on the author's local desktop, achieving a further reduction in the spring-in angle and processing time while maintaining a high degree-of-cure. The advancements achieved in this phase highlight the potential of the FEA optimization framework as an instrumental resource for augmenting composite manufacturing processes in the future.

REFERENCES

1. Kamal MR, Sourour S (1973) Kinetics and thermal characterization of thermoset cure. *Polym Eng Sci*. <https://doi.org/10.1002/pen.760130110>
2. DiBenedetto AT (1987) Prediction of the glass transition temperature of polymers: A model based on the principle of corresponding states. *J Polym Sci B Polym Phys*. <https://doi.org/10.1002/polb.1987.090250914>
3. O'Brien DJ, White SR (2003) Cure Kinetics, Gelation, and Glass Transition of a Bisphenol F Epoxide. *Polym Eng Sci*. <https://doi.org/10.1002/pen.10071>
4. Gutierrez-Lemini D (2014) Engineering viscoelasticity. *Engineering Viscoelasticity*. <https://doi.org/10.1007/978-1-4614-8139-3>
5. Zhang D, Waas AM (2014) A micromechanics based multiscale model for nonlinear composites. *Acta Mech*. <https://doi.org/10.1007/s00707-013-1057-1>
6. Arora JS (2012) More on Numerical Methods for Constrained Optimum Design. *Introduction to Optimum Design*:533–573. <https://doi.org/10.1016/B978-0-12-381375-6.00013-9>
7. Boyd S, Vandenberghe L (2006) *Convex Optimization*. 3:1–26.