

Article

Bayesian Optimal Experimental Design for Race Tracking in Resin Transfer Moulding

Nicholas Wright ¹, Piaras Kelly ¹, Oliver Maclaren ¹, Ruanui Nicholson ¹ and Suresh Advani ^{2,*}

¹ Department of Engineering Science, University of Auckland, Auckland 1010, New Zealand; nwri968@aucklanduni.ac.nz (N.W.); pa.kelly@auckland.ac.nz (P.K.); oliver.maclaren@auckland.ac.nz (O.M.); ruanui.nicholson@auckland.ac.nz (R.N.)

² Department of Mechanical Engineering and Centre for Composite Materials, University of Delaware, Newark, DE 19716, USA

* Correspondence: advani@udel.edu

Featured Application: Predict race-tracking strength in an RTM process using minimal pressure sensor measurements and position the sensors optimally throughout the preform in order to do so.

Abstract: A Bayesian inference formulation is applied to the Resin Transfer Moulding process to estimate bulk permeability and race-tracking effects using measured values of pressure at discrete sensor locations throughout a preform. The algorithm quantifies uncertainty in both the permeability and race-tracking effects, which decreases when more sensors are used or the preform geometry is less complex. We show that this approach becomes less reliable with a smaller resin exit vent. Numerical experiments show that the formulation can accurately predict race-tracking effects with few measurements. A Bayesian A-optimality formulation is used to develop a method for producing optimal sensor locations that reduce the uncertainty in the permeability and race-tracking estimates the most. This method is applied to two numerical examples which show that optimal designs reduce uncertainty by up to an order of magnitude compared to a random design.

Keywords: permeability; computational modelling; Bayesian inference; resin transfer moulding (RTM)



Citation: Wright, N.; Kelly, P.; Maclaren, O.; Nicholson, R.; Advani, S. Bayesian Optimal Experimental Design for Race Tracking in Resin Transfer Moulding. *Appl. Sci.* **2023**, *13*, 11606. <https://doi.org/10.3390/app132011606>

Academic Editor: Alberto Corigliano

Received: 20 August 2023

Revised: 25 September 2023

Accepted: 16 October 2023

Published: 23 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Resin Transfer Moulding (RTM) is a common advanced processing method for producing near-net-shaped composite parts. It consists of five key stages:

1. **Preform Manufacturing:** The reinforcing fabric is manufactured, typically by stacking layers of fabric upon one another.
2. **Lay-up and Draping:** The fabric preform is layed up or draped in the mould.
3. **Mould Closure:** The fabric layers are compressed to the final thickness of the part.
4. **Resin Injection:** Resin is injected into the mould through the injection gate(s) until the preform is fully saturated and the resin exits through a vent.
5. **Cure and Demoulding:** A part is removed from the mould after the resin cures.

RTM is often used within the industry because of the relatively short production times and simple experimental setup to construct near-net-shaped parts. However, these advantages come with caveats.

An important stage in RTM is resin injection, where the flow of the resin must be carefully controlled to avoid incomplete fillings, in which air can be trapped inside the mould, forming macroscopic dry spots. Even very small regions without resin can cause catastrophic failure of the composite part [1–4]. Accurate predictions of resin flow allow for strategic design of gates, vents, and injection schemes to optimally fill the composite part, with minimal chance of dry spot formation [1].

The flow of the resin through the fabric preform depends on the permeability. Deviations in the permeability from what is expected often come from the lay-up and draping

stage of manufacturing, when the preform is placed into the mould. Occasionally, there can be areas of very high permeability at the boundaries of the preform where the fabric density is lower. These are 'race tracks' for the resin as it fills the mould. The strength and existence of race tracking are influenced by a variety of factors, including fabric type, the preform manufacturing method, and the placement of the preform into the mould. It can vary from one part to the next in the same production run and is usually not repeatable [5].

The unpredictable nature of race tracks, in terms of their strength, can lead to significant variations in resin propagation, significantly changing the flow front movement pattern [6–8]. This can lead to dry spots in the composite parts, as the flow moves in an unpredictable way, trapping air. It is of great interest to be able to predict the effects of race tracking and mitigate the chance of dry spot formation, even in cases of high race tracking. When race tracking is predicted, one can use it as a method for advantageous resin distribution, rather than race tracking being an unpredictable problem.

Research around the effects of race tracking is not mainly focused on the prevention of race tracking, which is inevitable to some extent, but on understanding and predicting the race-tracking effect. Techniques from areas of statistics and machine learning have been incorporated into prediction, as well as methods that incorporate existing physical models of RTM [8–12].

One significant example, [5], addresses the problem of predicting race tracking online, as the resin is injected. This is an important problem as it allows manufacturers to accurately predict the flow of the resin, taking into account the specific race-tracking scenario for each part. Online automation tools could then be used to change the resin flow to ensure that there are no dry spots by introducing control actions during filling, such as introducing new injection locations, opening and closing exit vents, and changing injection pressures [13]. The method is based on discretising potential race-tracking effects for each susceptible region and creating a database of resin flow for each permutation of potential effects for each region. As a part is filled, sensors collect data that are then compared to the closest scenario in the database. This method is efficient because all the extensive computation can be completed offline. However, by considering only discrete scenarios and approximating flow based on the closest scenario, the reality could differ (possibly significantly) from what is predicted.

To allow for the systematic incorporation of uncertainties with race-tracking prediction, we consider this inverse problem in the Bayesian framework [14–17]. In this framework, the solution to the underlying statistical inverse problem is given by the posterior probability density. For nonlinear inverse problems with expensive forward models and high-dimensional parameters (as is the case for RTM inverse problems), fully characterising the posterior is typically not tractable. Consequently, we compute the Laplace approximation of the posterior, which requires only the maximum a posteriori (MAP) estimate, i.e., permeability and race-tracking strength, which maximise the posterior density and the approximate posterior covariance.

The idea of incorporating Bayesian inference into this problem has been explored previously [18,19]. These investigations focus on the more general problem of predicting a complete permeability field throughout the mould. The purpose of this would be to use pressure and flow front data created during filling to predict defects in the final part after it is produced but before any structural testing. They showed that a permeability and porosity field can be predicted using Bayesian inference, including predicting potential defects. However, extracting enough data with multiple pressure sensors and linear flow front sensors could be considered impractical and tedious in real manufacturing cases.

The current work uses a novel, lower-dimensional parameterisation compared to that formulation [18,19], where permeability throughout the domain is characterised by a homogeneous region, along with homogeneous race-tracking regions with a greater permeability. The purpose of this work is to estimate the race tracking that occurs during mould filling. The parameterisation presented here allows for the efficient estimation of race-tracking strength while neglecting other smaller spatially varying permeability effects.

The simplification to a lower-dimensional parameterisation also allows for much faster computation, which opens the potential for online calculation in the future. In this paper, we use offline calculations to compute the optimal locations for pressure sensors, a problem that has not yet been considered to the best of the knowledge of the authors. These locations could be used to predict race tracking online in the future. It is the hope that an optimally designed experimental setup will allow practitioners to extract maximum insight into the permeability, and therefore flow pattern, as the mould is being filled. From there, control actions can be taken, as has been investigated before [5,13,20,21], to ensure that the resin flow is controlled and there are minimal dry spots.

2. Computational Model

2.1. Constitutive Equations

This paper is limited to considering the 2D case, which is sufficient to model the vast majority of net-shaped parts for which RTM is used. However, the computational model and all Bayesian formulations in this model can be applied to higher-dimensional formulations.

For the slow-moving thermoset resins that are typically used in RTM, Darcy's law adequately describes the 2D flow through a porous medium [1,5,13,20,22–24]:

$$\mathbf{q} = -\frac{\mathbf{K}}{\mu} \nabla p, \quad (1)$$

where $\mathbf{q} \in \mathbb{R}^2$ is the flux vector of resin, p is the pressure, μ is the fluid viscosity, and $\mathbf{K} \in \mathbb{R}^{2 \times 2}$ is the permeability tensor, which is symmetric and positive definite. Combining Darcy's law with conservation of mass gives:

$$-\nabla \cdot \left(\frac{\mathbf{K}}{\mu} \nabla p \right) = 0. \quad (2)$$

We base the following notation on the notation used in [19] to describe this problem. The domain of the mould D can be partitioned at any point in time into $F(t)$ and $U(t)$, which represent the filled and unfilled regions of the domain, respectively. The boundary of D is $\delta D = \delta D_I \cup \delta D_N \cup \delta D_O$, where δD_I represents the inlet, δD_N represents the impermeable walls of the mould, and δD_O is the outlet vent. We consider the case where resin is injected from a gate at a fixed constant pressure (p_{in}) and exits from a vent. At this vent, resin pressure is fixed at atmospheric pressure, taken to be 0 without loss of generality. The walls of the mould have zero normal flow. All unfilled regions are also fixed at zero pressure, neglecting the scenario in which air is sealed and compressed in the mould, which we are trying to prevent. In summary, the problem has the following boundary conditions, where $\mathbf{n}(\mathbf{x})$ is the normal vector to the domain boundary at \mathbf{x} :

$$p(\mathbf{x}, t) = p_{\text{in}}, \quad \mathbf{x} \in \delta D_I, \quad (3)$$

$$p(\mathbf{x}, t) = 0, \quad \mathbf{x} \in \delta D_O, \quad (4)$$

$$\nabla p(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) = 0, \quad \mathbf{x} \in \delta D_N, \quad (5)$$

$$p(\mathbf{x}, t) = 0, \quad \mathbf{x} \in U, \quad (6)$$

and the following initial conditions:

$$U(0) = D - \delta D_I, \quad (F(0) = \delta D_I). \quad (7)$$

2.2. Numerical RTM Model

The RTM simulation developed here uses a standard multigrid method [25–27]. This uses control volumes (CVs) that keep track of resin saturation, offset from the triangular finite elements that are used to calculate the pressure through Equation (2). These are

illustrated in Figure 1. This paper will refer to these as ‘control volumes’ (CVs) and ‘elements’, respectively, to avoid confusion.

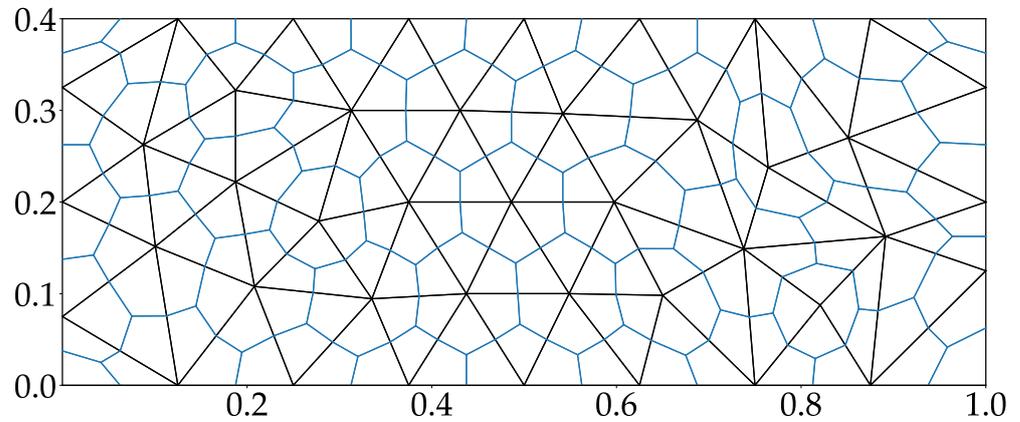


Figure 1. Control volumes (blue) offset from the triangular finite elements (black) for a basic mesh (a more complex mesh is used in the numerical simulations).

The algorithm can be divided into a series of steps:

1. Solve finite element equations for pressures at the nodes.
2. Calculate the flow between each control volume.
3. Calculate the time to fill the next control volume.
4. Step forward in time and propagate the resin front forward.

These steps are repeated until all control volumes are full. This algorithm is available in LIMS [28,29], which can address complex geometries and 3D flows. However, for this work, the goal was to explore the A-optimality formulation in combination with the Bayesian framework. Once demonstrated, this can be implemented in other RTM simulations. As such, our numerical simulations in this paper use an implementation of this algorithm written in Julia code. Each of the steps of the algorithm is discussed in more detail in the following.

2.2.1. Solve for Pressure

Equation (2) can be written in a weak form using the Galerkin finite element method, for a given mesh. The derivation of this is given in Appendix A. This gives the stiffness equation to solve:

$$E(\mathbf{K})\mathbf{p} = \mathbf{f}, \tag{8}$$

where $E(\mathbf{K})$ and \mathbf{f} are the finite element stiffness matrix and the forcing vector, respectively, as defined in Appendix A. All nodes with respective CVs in $U(t)$ are fixed at $p = 0$, as discussed in Section 2.1. Applying boundary conditions based on these time-varying sets alters $E(\mathbf{K})$ to $A(\mathbf{K}, t)$, which is defined as follows, using MATLAB notation:

$$A(i,:) = \begin{cases} E(i,:) & CV_i \in \mathcal{F}, \\ e_i^T & CV_i \in U(t) \text{ or } n_i \cup \delta D_I \cup \delta D_O, \end{cases} \tag{9}$$

where CV_i represents the i th CV, n_i represents the i th node, and e_i^T is the standard unit basis vector. The vector \mathbf{f} does not change as U and \mathcal{F} change and is defined as follows:

$$\mathbf{f}(i) = \begin{cases} p_{in} & n_i \in \delta D_I, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Equation (8) is never actually solved, but, instead, the following is solved every timestep:

$$A(K, t)p(t) = f. \tag{11}$$

It is worth noting that $A(K, t)$ is generally very sparse.

2.2.2. Calculate Flow

Given the finite element pressure distribution, the flow between CVs is calculated. The flow between filled CVs is not important, as the net flux will add to zero. CV_i has a saturation Ψ_i where

- $\Psi_i = 0$ represents an empty CV.
- $\Psi_i = 1$ represents a filled CV.
- $0 < \Psi_i < 1$ is a partially filled CV.

CVs that increase in saturation during the current timestep must be considered. To this end, we define the set X that contains all the values for i such that CV_i has an adjacent CV that is filled and $\Psi_i < 1$.

For a given finite element, we can express Darcy’s law for the flow rate q —Equation (2)—in terms of the nodal pressures:

$$q = -\frac{1}{\mu}K \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix}, \quad \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix} = J^{-1} \begin{bmatrix} \sum_{i=1}^3 p_i \frac{\partial \phi_i}{\partial \xi_1} \\ \sum_{i=1}^3 p_i \frac{\partial \phi_i}{\partial \xi_2} \end{bmatrix}. \tag{12}$$

Here, J is the finite element Jacobian, defined as follows, based on the three coordinates of the element (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) :

$$J = \begin{bmatrix} x_1 - x_3 & y_1 - y_3 \\ x_2 - x_3 & y_2 - y_3 \end{bmatrix}. \tag{13}$$

Note the lack of dependence of q on the element coordinates—the linear elements for pressure give a constant flow rate throughout an element. Through a boundary C , the volume flow rate Q_C can be obtained by integrating $q \cdot n$ across the boundary. Within one element, there is no spatial dependence, so we have the following for Q_C :

$$Q_C = (q \cdot n)|C|, \tag{14}$$

where $|C|$ is the length of the boundary C . By summing Q_C for all boundaries for the i th CV, the total flow Q_i is calculated.

2.2.3. Time to Fill Next CV

Once the total flow Q_i is found, the time to fill the CV can be calculated based on the current saturation. For all $i \in X$:

$$t_{\text{fill}}^{(i)} = \frac{\phi(1 - \Psi_i)A_i}{Q_i}, \tag{15}$$

where ϕ is the porosity of the domain and A_i is the area of CV_i . We then know the timestep to propagate to the next iteration:

$$\Delta t = \min_{i \in X} t_{\text{fill}}^{(i)}, \tag{16}$$

$$i' = \operatorname{argmin}_{i \in X} t_{\text{fill}}^{(i)}. \tag{17}$$

2.2.4. Timestep

The time dependence in this problem comes only from the moving boundary, causing a change in $U(t)$ and $F(t)$. At the end of the iteration of this method, the following change takes place:

$$U(t + \delta t) = U(t) - CV_i, \quad (18)$$

$$F(t + \delta t) = F(t) + CV_i. \quad (19)$$

If there are still unfilled CVs, we return to solving for pressure (Section 2.2.1). This begins with a change from $A(\mathbf{K}, t)$ to $A(\mathbf{K}, t + \delta t)$ (Equation (9)).

2.2.5. Interpolation

For this investigation, it is of interest to be able to output the measurements that pressure sensors at specific locations would give at specified measurement times. The finite element solution defines the pressure at every location in the domain. However, a disadvantage of the algorithm described here is that the timesteps are dependent on the filling times. Therefore, extracting pressure measurements at specific times requires linear interpolation.

3. Bayesian Formulation

3.1. Permeability Vector

The unknown model parameters that represent the permeability are described by $\mathbf{k} = [c_1, c_2, c_3, \alpha_1, \alpha_2, \dots, \alpha_n]$. We use c_1 , c_2 , and c_3 to represent the log-Cholesky factorisation of the bulk permeability tensor, \mathbf{K}_{bulk} —see (20). \mathbf{K}_{bulk} represents the permeability tensor for the majority of the domain, aside from the region experiencing race tracking. Specifically, we take:

$$\mathbf{C}^T \mathbf{C} = \mathbf{K}_{\text{bulk}}, \quad \mathbf{C} = \begin{bmatrix} e^{c_1} & c_2 \\ 0 & e^{c_3} \end{bmatrix}. \quad (20)$$

Parameterising the bulk permeability in this way ensures that the bulk permeability tensor is symmetric and positive definite, regardless of the values of c_1 , c_2 , and c_3 [30]. The remaining parameters $\alpha_1, \alpha_2, \dots, \alpha_n$ represent race tracking in various areas. For given values of these parameters, the permeability tensor of each area can be constructed as follows.

$$\mathbf{K}_j = \exp(\alpha_j) \mathbf{K}_{\text{bulk}}, \quad j = [1, 2, \dots, n], \quad (21)$$

where \mathbf{K}_j is the permeability tensor in a specific race-tracking region. The use of the exponential function is convenient, as it allows the parameters α_j to be normally distributed, but gives an exponential distribution in the race-tracking multipliers. This exponential distribution reflects typical race-tracking multipliers, which are often 1–5 times larger than the bulk permeability but can be 100 times higher in some instances [1,5,13,20]. The benefit of using normally distributed random parameters will be discussed in the following sections.

It should be noted that we have limited the stochastic variables in our model to only those parameterised by k . This includes the bulk permeability and the race-tracking strength of each region. We fix the porosity and race-tracking width parameters. The reasons for excluding each are different. Stochastic porosity has been considered in a previous study [19] but will be neglected here due to its typical low variation and because the focus of this research is specifically on race tracking. We neglect the race-tracking width, although stochastic in actuality, because the dual effect of the race-tracking width and strength can be replicated with just strength. As the interest in race tracking is in its effect rather than describing the exact source, this is an appropriate assumption.

3.2. Bayesian Inference

The objective of this investigation is to estimate the permeability vector k , which exists and produces a vector of measurements d , as the composite part is produced. For the RTM problem, the length of this vector is the number of measurement times multiplied by the number of pressure sensors. We assume that these data contain some error or noise e , so we have:

$$d = \mathcal{G}(k) + e, \quad e \sim \mathcal{N}(\mathbf{0}, \Gamma_e), \tag{22}$$

where Γ_e is a covariance matrix that describes the distribution of this error, and \mathcal{G} represents the ‘forward model’. That is, $\mathcal{G}(k)$ produces noise-free measurements that would be taken when completing an RTM resin injection with permeability vector k . The computational model described in Section 2 is used every time $\mathcal{G}(k)$ is calculated.

This estimation of k can be achieved by Bayesian inference. We treat the unknown k as a random variable and determine the posterior probability distribution $\pi(k|d)$. This has the added benefit of automatically quantifying the uncertainty in the parameters. The posterior can be found using Bayes’ rule:

$$\begin{aligned} \pi(k|d) &= \frac{\pi(d|k)\pi(k)}{\pi(d)} \\ &\propto \pi(d|k)\pi(k). \end{aligned} \tag{23}$$

Here, $\pi(d|k)$ is the likelihood function, which is the probability of receiving the data for a given permeability vector. If we assume that the noise e and the parameters k are independent, and given Equation (22), this likelihood is [14]:

$$\begin{aligned} \pi(d|k) &\propto \exp\left(-\frac{1}{2}(\mathcal{G}(k) - d)^T \Gamma_e^{-1}(\mathcal{G}(k) - d)\right) \\ &= \exp\left(-\frac{1}{2}\|L_e(\mathcal{G}(k) - d)\|^2\right). \end{aligned} \tag{24}$$

where $\Gamma_e^{-1} = L_e^T L_e$.

On the other hand, $\pi(k)$ is the prior that encodes the prior beliefs of k . Given the parameterisation described in Section 3.1, it is now sensible to describe the prior using a multivariate Gaussian distribution with the mean μ_k and covariance Γ_k . This gives the following equation for the prior probability density:

$$\begin{aligned} \pi(k) &\propto \exp\left(-\frac{1}{2}(k - \mu_k)^T \Gamma_k^{-1}(k - \mu_k)\right), \\ &= \exp\left(-\frac{1}{2}\|L_k(k - \mu_k)\|^2\right), \end{aligned} \tag{25}$$

where $\Gamma_k^{-1} = L_k^T L_k$. Combining Equations (23)–(25) gives the following equation for the posterior:

$$\pi(k|d) \propto \exp\left(-\frac{1}{2}\|L_e(\mathcal{G}(k) - d)\|^2 - \frac{1}{2}\|L_k(k - \mu_k)\|^2\right) \tag{26}$$

3.3. Exploring the Posterior

Although Equation (26) is the posterior, we need some way of interpreting this distribution. The posterior could be analysed using Markov Chain Monte Carlo (MCMC) methods [8], but this would involve evaluating the posterior hundreds of thousands of times. Calculating the posterior is a computationally expensive operation because it requires calculating $\mathcal{G}(k)$ and running the RTM model. The RTM model involves solving the system of equations for pressure at each timestep, where the number of timesteps scales with the number of nodes. Other research in this area focuses on the development of less computationally expensive methods [18,19].

We use a Laplace approximation to the posterior, which requires several orders of magnitude less computational time. The Laplace approximation works by approximating the posterior about the maximum a posteriori (MAP) estimate. We then take a Gaussian approximation of this estimate. However, the calculation of the MAP estimate is not trivial and requires maximising the posterior. This can be simplified by noting the following.

$$\begin{aligned}
 k_{\text{MAP}} &= \underset{k}{\operatorname{argmax}} (\pi(k|\mathbf{d})) \\
 &= \underset{k}{\operatorname{argmin}} (-2 \ln \pi(k|\mathbf{d})) \\
 &= \underset{k}{\operatorname{argmin}} (\|L_e(\mathcal{G}(k) - \mathbf{d})\|^2 + \|L_k(k - \mu_k)\|^2).
 \end{aligned}
 \tag{27}$$

Once the MAP estimate has been found, the Laplace approximation to the posterior can be found by considering the Taylor expansion of $\mathcal{G}(k)$ about k_{MAP} :

$$\mathcal{G}(k) \approx \mathcal{G}(k_{\text{MAP}}) + \mathcal{J}(k - k_{\text{MAP}}),
 \tag{28}$$

where \mathcal{J} is the Jacobian of the RTM model evaluated at the MAP estimate. This Jacobian considers the derivative of all outputs of $\mathcal{G}(k)$, with respect to each of the components of k :

$$\mathcal{J} = \begin{bmatrix} \frac{\partial \mathcal{G}_1}{\partial k_1} & \frac{\partial \mathcal{G}_1}{\partial k_2} & \cdots & \frac{\partial \mathcal{G}_1}{\partial k_n} \\ \frac{\partial \mathcal{G}_2}{\partial k_1} & \frac{\partial \mathcal{G}_2}{\partial k_2} & \cdots & \frac{\partial \mathcal{G}_2}{\partial k_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{G}_m}{\partial k_1} & \frac{\partial \mathcal{G}_m}{\partial k_2} & \cdots & \frac{\partial \mathcal{G}_m}{\partial k_n} \end{bmatrix},
 \tag{29}$$

where \mathcal{G}_i is the i th component of the output vector from \mathcal{G} . Substituting this into Equation (26) gives an equation for the Laplace approximation of $\pi(k|\mathbf{d})$, $\pi_{\text{Laplace}}(k|\mathbf{d})$:

$$\pi_{\text{Laplace}}(k|\mathbf{d}) \propto \exp\left(-\frac{1}{2}\|L_e(\mathcal{J}k - z)\|^2 - \frac{1}{2}\|L_k(k - \mu_k)\|^2\right),
 \tag{30}$$

where $z = \mathbf{d} + \mathcal{J}k_{\text{MAP}} - \mathcal{G}(k_{\text{MAP}})$, which does not depend on k . This probability density function describes a Gaussian distribution (that is, $\pi_{\text{Laplace}} \sim \mathcal{N}(k_{\text{MAP}}, \Gamma_{\text{post}})$), the covariance of which can be calculated as follows:

$$\Gamma_{\text{post}} = (\mathcal{J}^T \Gamma_e^{-1} \mathcal{J} + \Gamma_k^{-1})^{-1}.
 \tag{31}$$

4. Numerical Examples

We now apply the outlined methodology to two example problems. For these, the data are generated by running the RTM model with a true permeability, which we denote by k_{true} , which we are trying to estimate. Then, noise is added to these simulated pressures. It is assumed that the permeability in the bulk of the domain (K_{bulk}) is homogeneous and that the permeability in the race-tracking regions is a scalar multiple of the bulk region.

We assume the data come in the form of pressure measurements taken at discrete sensors at predefined times. The pressure sensors are placed at optimal locations using the scheme described in the later section on Bayesian Optimal Experimental Design (Section 6). All pressure measurements are then normalised by dividing by the inlet pressure.

The programming language Julia was used to both execute the RTM simulations and to complete the Bayesian calculations in general. This allows for faster code execution compared to other interpreted languages, and we use the *ForwardDiff* package for automatic derivative calculation [31].

A Gauss–Newton algorithm [32] with random restarts is used to carry out the optimisation required to find the MAP estimate. We can reliably produce solutions with smaller

residuals (i.e., a lower cost function described by Equation (27) than k_{true} , with less than 10 restarts, for the small-scale models considered in this paper.

4.1. Model 1

The first investigated example is illustrated in Figure 2. Resin is injected into the mould from a 5 cm injection gate in the lower left corner of the domain at a constant pressure of $p_{\text{in}} = 400$ kPa. The only exit vent is at the top right corner of the domain, which is again fixed at 5 cm. There are three areas of potential race tracking, with different permeabilities at the top and bottom of the domain, as well as around a central block. In reality, there will be potential for race tracking along the vertical walls, but this is ignored here for simplicity.

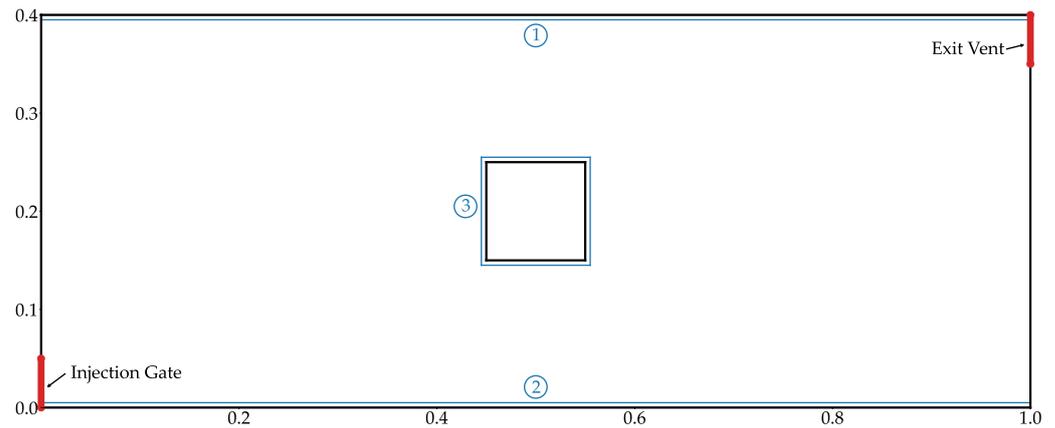


Figure 2. Domain for Model 1, representing a rectangular part with a square block at the centre. Race-tracking regions are indicated in blue and are numbered.

As there are three race-tracking regions, the permeability vector is

$$k = [c_1, c_2, c_3, \alpha_1, \alpha_2, \alpha_3]^T, \tag{32}$$

where:

$$K_i = \exp(\alpha_i) K_{\text{bulk}}, \tag{33}$$

and K_i represents the permeability tensor for region $i = 1, 2, 3$, as shown in Figure 2.

Here, k_{true} was arbitrarily defined as follows:

$$k_{\text{true}} = [0, 0, 0, \ln(10.0), \ln(1.0), \ln(50.0)]^T. \tag{34}$$

This represents a scenario where there is no race tracking along the bottom, 10 times higher permeability on the top, and 50 times higher permeability around the centre of the domain. All permeability values here have units of 10^{-11} m^2 .

Error was added to the output of this model with the distribution $\mathcal{N}(\mathbf{0}, \Gamma_e)$. For this experiment, we set

$$\Gamma_e = \sigma_e^2 I, \quad \sigma_e = 0.01. \tag{35}$$

As with the error distribution, the prior for this experiment is arbitrary and can be adjusted to the beliefs of the implementer but should, in general, be created broadly so as not to introduce bias into the prediction. This experiment used the following prior, which roughly approximated our prior beliefs in the race-tracking and permeability values.

$$\mu_k = [0, 0, -0.144, 2, 2, 2]^T, \tag{36}$$

$$\Gamma_k = \text{diag}([0.25, 1, 0.25, 2, 2, 2]). \tag{37}$$

The prior mean represents a permeability field that it is slightly more permeable in the x direction than in the y direction. Specifically, it represents the following tensors:

$$K_{\text{bulk}} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{3}{4} \end{bmatrix}, \quad K_1, K_2, K_3 = e^2 K_{\text{bulk}} \tag{38}$$

4.2. Model 2

This example considers a more complicated scenario in which there are ten race-tracking regions. This time, we let resin exit through the entire right-hand boundary. The reasons for this are discussed in Section 7.1. There are now two central blocks, each with a different race-tracking strength along each of its four edges. The domain is illustrated in Figure 3.

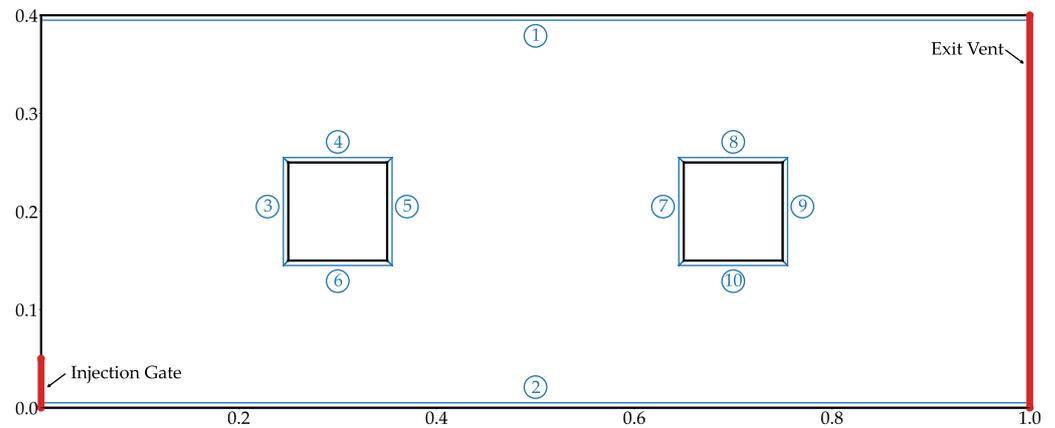


Figure 3. Domain for Model 2, representing a rectangular part with two square blocks. Each race-tracking region is labelled in blue and numbered.

The associated permeability vector is

$$k = [c_1, c_2, c_3, \alpha_1, \alpha_2, \dots, \alpha_{10}]^T, \tag{39}$$

where $\alpha_1, \alpha_2, \dots, \alpha_{10}$ are defined in Equation (21) and their respective regions are labelled in Figure 3; k_{true} is taken as a random sample from the prior, which is defined as similar to Model 1, specifically:

$$\mu_k = [0, 0, -0.144, 2, 2, \dots, 2]^T, \tag{40}$$

$$\Gamma_k = \text{diag}([0.25, 1, 0.25, 2, 2, \dots, 2]). \tag{41}$$

We use the same error distribution as in Model 1, given in Equation (35).

5. Results

5.1. Model 1

The posterior and prior for Model 1 are compared in Figure 4 when only two pressure sensors are used. We take measurements every 1000 s, for up to 15,000 s. When k_{true} is used, this gives a fill time of 10,500 s, which means that we measure both during filling and after the resin has reached the vent—a steady state pressure distribution.

The posterior encompasses the truth for all parameters as expected. The posteriors are generally broad due to the low amount of data used to develop the posterior, particularly in parameters such as α_1, α_2 , and α_3 , which are perhaps less influential on the pressure measurements themselves. It is important to note that although Figures 4–6 show each parameter’s marginal posterior individually, there is a multivariate correlation between the parameters developed by the posterior that is not visualised.

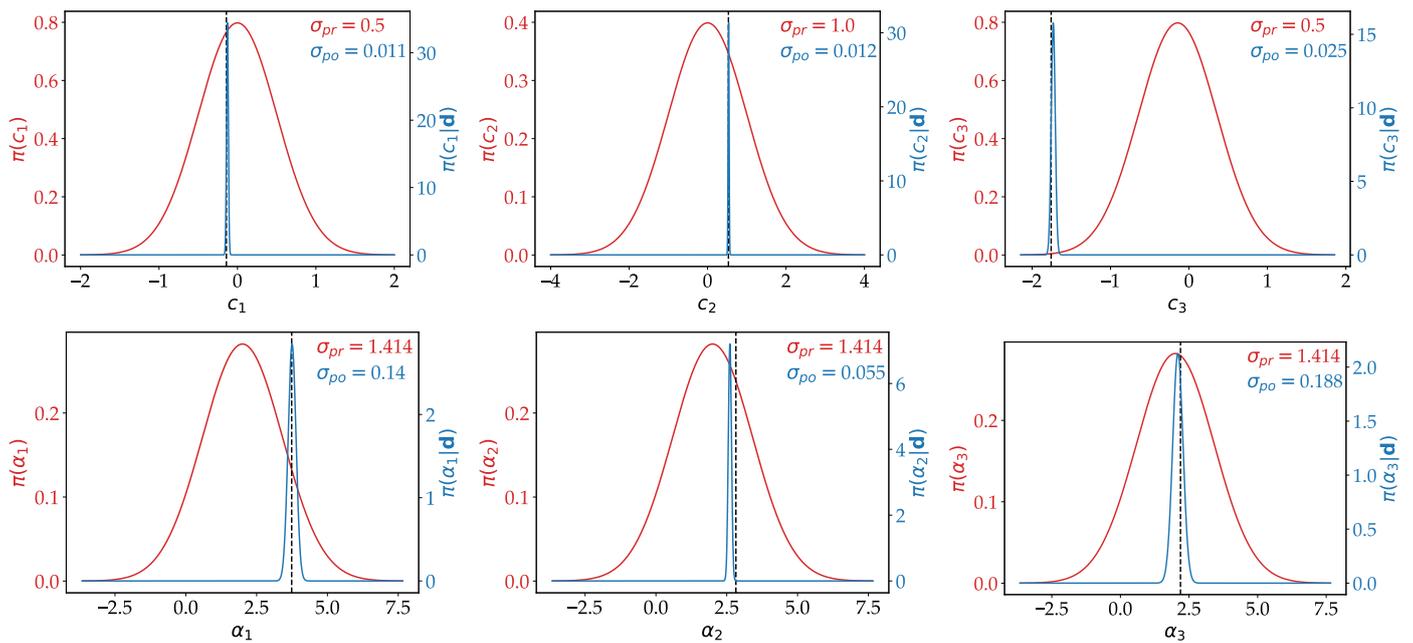


Figure 4. Marginal prior (red) and posterior (blue) distributions for Model 1 with 2 sensors. Truth is shown with a black dashed line. Note that these distributions are presented on separate y-axes, for ease of comparison.

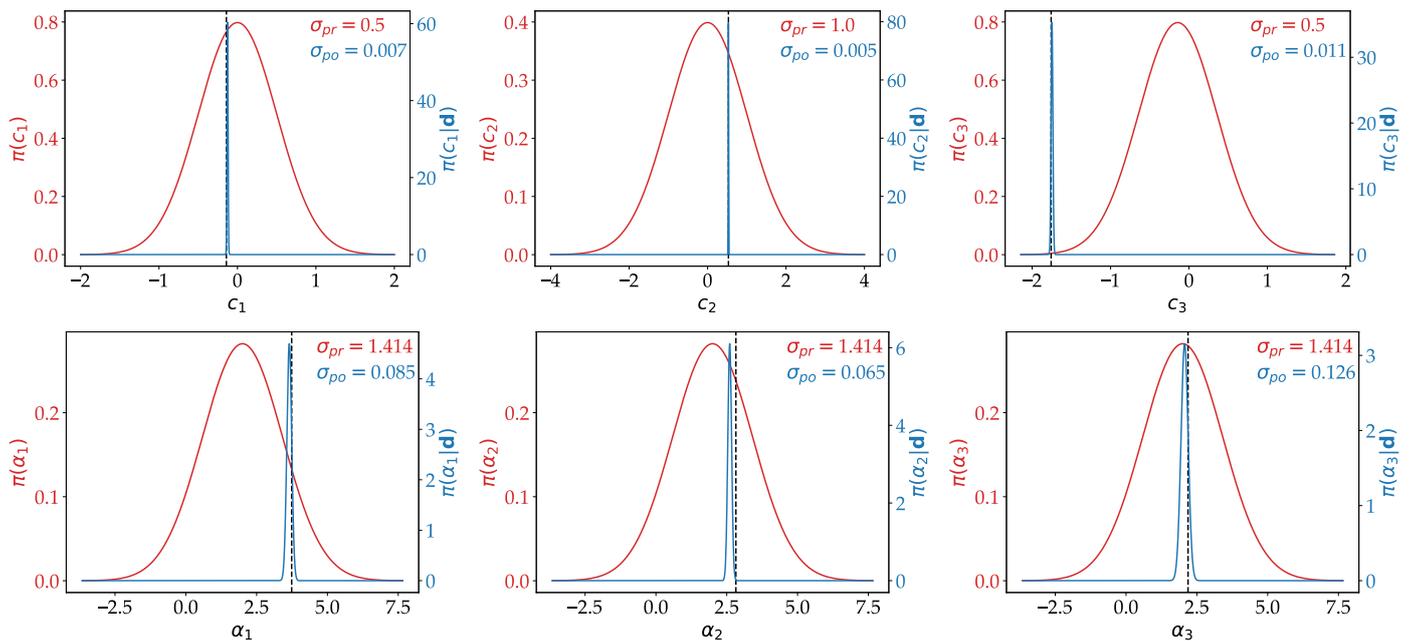


Figure 5. Marginal prior (red) and posterior (blue) distributions for Model 1 with 10 sensors. Truth is shown with a black dashed line. Note that these distributions are presented on separate y-axes, for ease of comparison.

Figure 5 shows how the posterior varies when 10 pressure sensors are used. With the increase in information provided by the 10 sensors, the marginal posteriors have less uncertainty. However, the caveats of using the Laplace approximation are evident here, particularly for the posterior for α_2 , where the truth is not encompassed by the posterior. Although we can be sure that the unapproximated posterior would cover these truths, the Laplace approximation can sometimes be inaccurate. In addition, the posterior with 10 sensors has greater uncertainty than with 2 sensors for this parameter. One key reason for these inaccuracies in the Laplace approximation is discussed in Section 7.1.

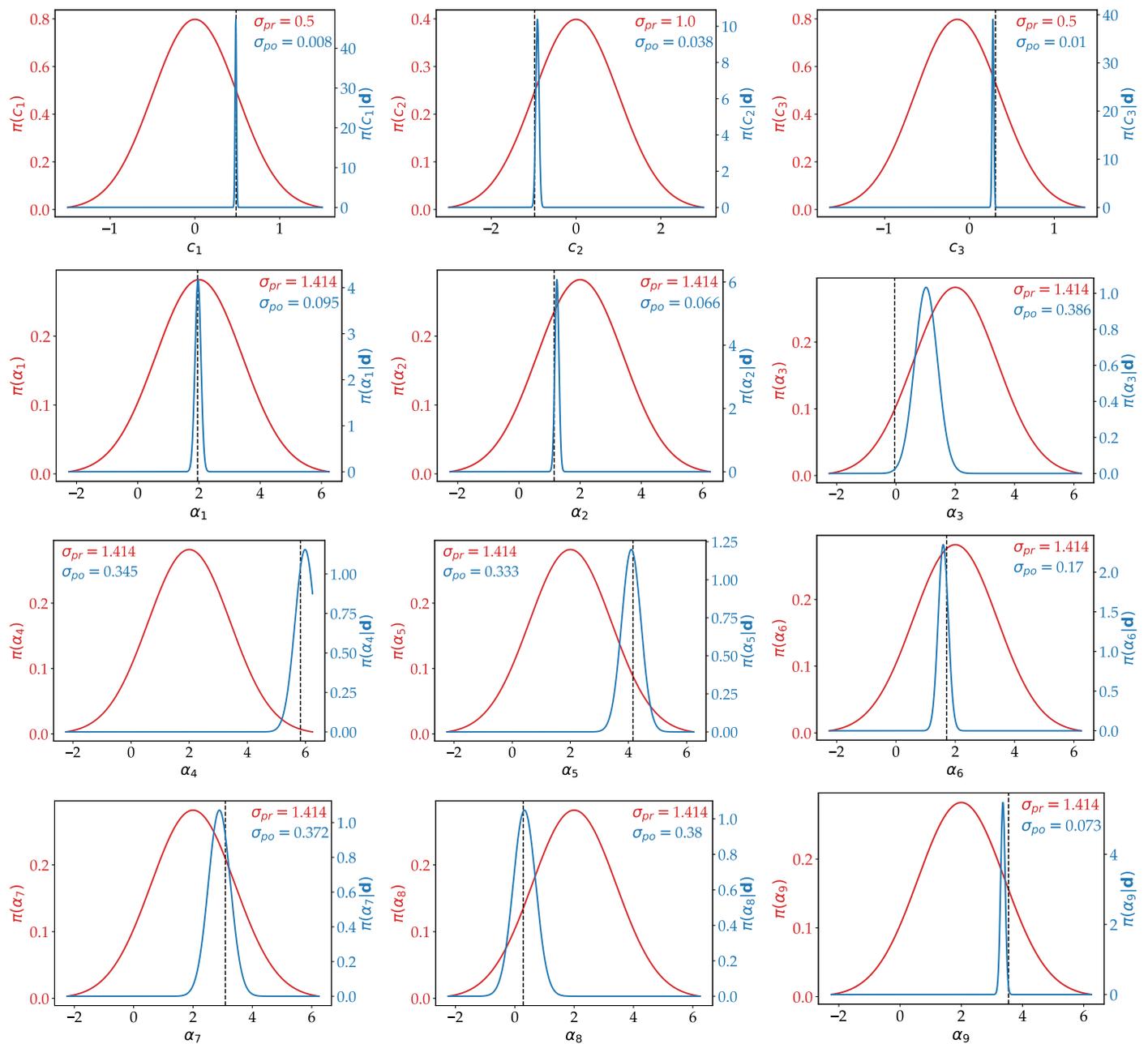


Figure 6. Marginal prior (red) and posterior (blue) distributions for Model 2 with 10 sensors. Truth is shown with a black dashed line. Note that these distributions are associated with separate y-axes, for ease of comparison. For presentation, α_{10} is also omitted.

These results are quantitatively summarised in Table 1, which shows how the prior, posterior, and truth compare when a 99% confidence interval is developed from the prior and posterior.

5.2. Model 2

The marginal priors and posteriors for the more complex Model 2 are compared in Figure 6. Because there are more parameters to estimate, we now take more data. We consider only the 10-sensor setup and take measurements every 100 s for a total of 10,000 s. We take measurements in a shorter time period because the mould takes only 7800 s to fill with k_{true} .

The uncertainty in influential parameters such as the bulk permeability and the top and bottom race-tracking regions is very low because these parameters have a greater

impact on domain pressure. The uncertainty in the less significant regions around the central blocks is much larger. The truth is encompassed by the posterior for all parameters, as expected.

Table 1. Figures 4 and 5 summarised in a table. Confidence intervals are 99% based on 3 standard deviations.

	c_1	c_2	c_3
Truth	−0.141	0.531	−1.758
Prior	0.000 ± 1.500	0.000 ± 2.121	−0.144 ± 1.500
Posterior (2 Sensor)	−0.128 ± 0.034	0.537 ± 0.036	−1.731 ± 0.076
Posterior (10 Sensor)	−0.127 ± 0.020	0.533 ± 0.015	−1.747 ± 0.034
	α_1	α_2	α_3
Truth	3.737	2.824	2.189
Prior	2.000 ± 4.243	2.000 ± 4.243	2.000 ± 4.243
Posterior (2 Sensor)	3.751 ± 0.421	2.612 ± 0.166	2.093 ± 0.563
Posterior (10 Sensor)	3.653 ± 0.254	2.605 ± 0.196	2.067 ± 0.379

6. Bayesian Optimal Experimental Design

This paper has so far shown that a lot of information about race tracking and permeability in the preform could be determined with even two sensors, taking 15 discrete measurements each. This section investigates where to optimally place sensors in order to bring posterior uncertainty to a minimum and how this varies when different amounts of sensors are used. The goal here is to allow practitioners to extract the maximum amount of information using minimal measuring equipment.

6.1. Method

We have $N \in \mathbb{N}$ sensors; we wish to place them within the mould so that when considering all potential permeability scenarios, the variance in parameters is minimised. One way to achieve this is to determine the covariance matrix for a variety of permeability scenarios with a sensor setup S . Then, we aim to minimise the sum of these covariance traces, which is known as A-optimality [33–40]. In essence, this is minimising the following cost function:

$$\min_S \sum_{i=1}^M \text{tr}(\Gamma_{\text{post}}^S(k_i)), \quad (42)$$

where M is the number of permeability scenarios considered and $\Gamma_{\text{post}}^S(k_i)$ is the posterior covariance for the permeability scenario k_i and sensor design S . A-optimality is appealing for this problem, where we could be interested in using the MAP permeability estimate to predict future resin flow and want minimal uncertainty in each individual estimate. Common alternatives, such as D-optimality or E-optimality [33], would consider the off-diagonal elements of the posterior covariance, which are not as relevant for this situation.

To simplify this problem, potential sensor locations will be limited to discrete locations spaced 0.025 m apart throughout the domain, resulting in 697 potential locations. When considering that the possible number of locations for N sensors is 697^N , we will limit the optimisation to follow a greedy decision process [36,39]. That is, the optimal location for the first sensor will be found, and then the second sensor will be optimally placed, given that the first is fixed. The process will be repeated for all N sensors. Sensors are also prohibited from occupying the same position. An advantage of the greedy algorithm is that the optimal design for N sensors gives us the optimal design for all $n < N$ sensors.

Despite reducing the computation time significantly with a greedy algorithm, the computation for solving this problem is still extremely intensive. Determining the MAP estimate by optimisation took 1–2 h on a standard desktop computer (12th Gen Intel Core i7-12700

CPU, no GPU acceleration). For more complex geometries and finer meshes, this time can significantly increase. Solving this inverse problem M times for each sensor location will take $697M$ times longer for every sensor. Although this could be parallelised, this would still take too long (potentially years) to receive a solution in a reasonable amount of time. As a computationally feasible approximation [39], we use:

$$\Gamma'_{\text{post}} = (\mathcal{J}_{\text{true}}^T \Gamma_e^{-1} \mathcal{J}_{\text{true}} + \Gamma_k^{-1})^{-1}, \tag{43}$$

where $\mathcal{J}_{\text{true}}^T$ is the Jacobian of the forward model evaluated at k_{true} . This is in contrast to Equation (31), where now we use the Jacobian evaluated at k_{true} for the current sample, instead of evaluating it at the MAP estimate. Although this is an approximation, the optimal sensor locations using this approximation should not vary significantly from the true optima [39] and bring the problem to a solvable time frame (hours). Note that although this approximation does not make sense to use in a standard Bayesian inference scenario, it is sufficient for the purposes of judging a given sensor design using Equation (42).

Computation can be further shortened by noting that we can precompute $\mathcal{J}_{\text{true}}$ for all potential sensor locations. Then, for a given sensor design S , we can simply extract the rows of the Jacobian that correspond to the relevant outputs, where the sensors have been placed. This can be computed for all M permeability scenarios.

6.2. Results

The optimal sensor locations for Model 1 are shown in Figure 7, calibrated using $M = 200$ different permeability scenarios. The choice of M was decided by investigating the robustness of the optimal locations trained with various amounts of data. This is omitted here for clarity but is contained in Appendix C, along with a brief discussion on the robustness of the design. The sensors tend to be distributed around the race-tracking areas. For the top and bottom race-tracking areas, it seems to be beneficial to have sensors at the start and end of these regions. There are some sensors (most notably sensor 2) that are positioned to gain pressure measurements at the inlet, where the large pressures make the influence of noise less confounding.

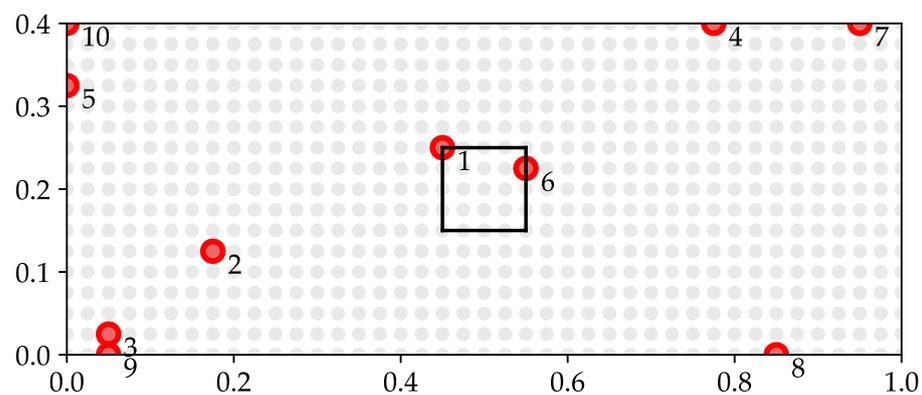


Figure 7. The positions within the domain of the optimal sensor locations, as determined by the greedy algorithm for Model 1. Sensors are indicated in red and numbered in the order they appear in the greedy algorithm—i.e., the optimal locations of four sensors are locations 1, 2, 3, and 4. All candidate locations are shown in grey.

Figure 8 illustrates how the objective function in Equation (42) changes with an increasing number of sensors, for Model 1. The results for (the more complex) Model 2 are given in Appendix B. The figure shows a monotonic decrease in objective function as more sensors are added, regardless of location, as expected. For the random design, there are points where adding a sensor gives no change in the objective. This is because the sensor is placed in a location with no information—for example, inside the central block. The optimal sensor locations lead to a significant reduction in uncertainty compared to

using random sensor locations, with roughly five times lower variance in the permeability estimates. However, this difference would be expected to converge to zero as the number of sensors increases to the maximum [33–36,39,40]. It is also clear in Figure 8 that the largest variances come from the race-tracking parameters $\alpha_1, \alpha_2,$ and α_3 rather than the bulk parameters $c_1, c_2,$ and c_3 . This is in agreement with the posterior variances shown in Section 5. The eighth sensor appears to have greatly reduced the uncertainty in $c_1, c_2,$ and c_3 . However, the logarithmic scale of the graph emphasises this drop, and the sum of variances for all parameters still decreases, as seen in the upper graph (red line).

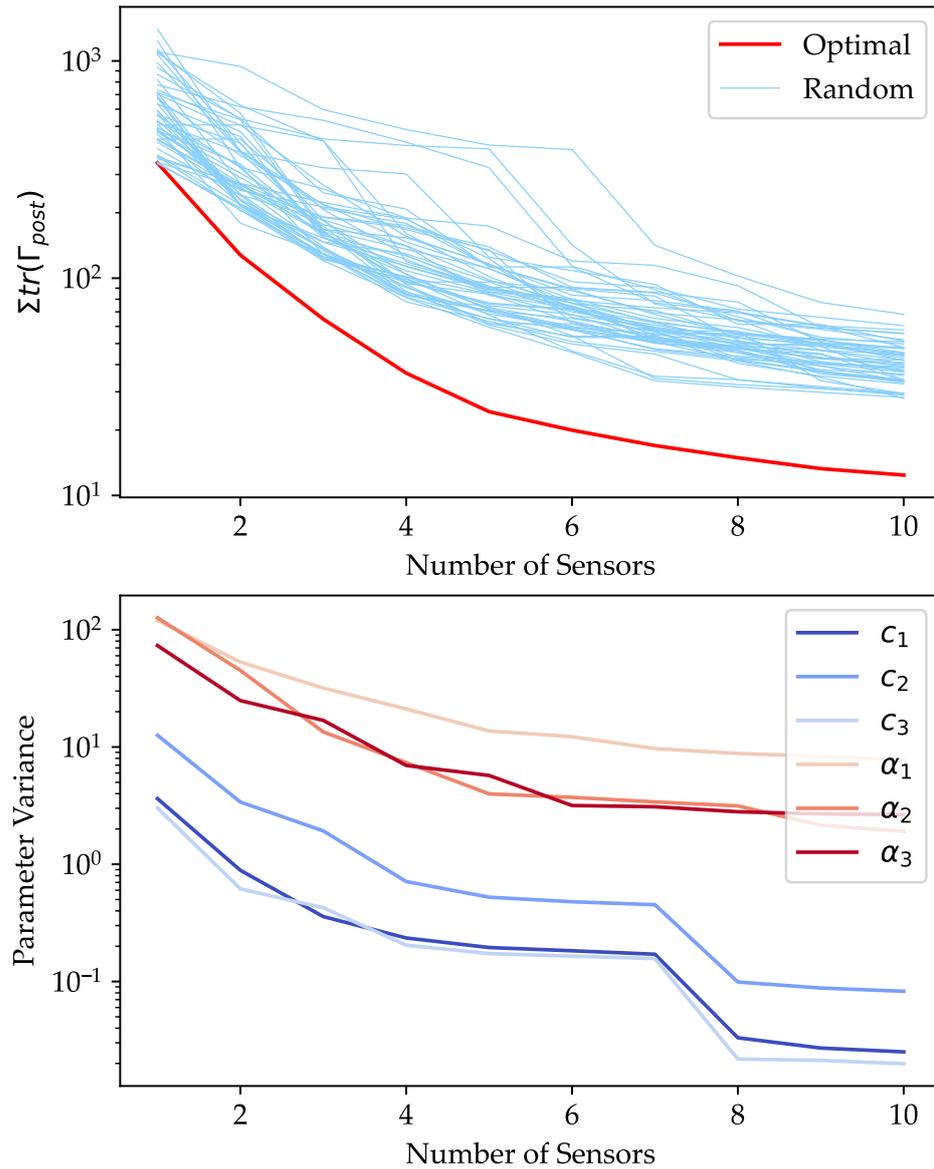


Figure 8. (Top:) Sum of the trace of the posterior covariance over all simulations versus the number of sensors used. The optimal solution is shown in red, and random solutions are shown in blue. (Bottom:) Individual variances of each posterior parameter (log scale), for the optimal design.

The MAP estimate for each parameter is shown in Figure 9, along with 99% confidence intervals. These intervals are based on the Laplace approximation to the posterior about the MAP estimate (Equation (31)), rather than the approximation defined in Equation (43). The MAP estimate tends to converge to the truth as more sensors are added, as expected. The variance tends to decrease as more sensors are added, as in Figure 8, but not monotonically. When calculating the Laplace approximation about the MAP estimate, there is

no guarantee of strictly decreasing variances because the MAP estimate changes. There are some instances where the truth is not contained within the 99% confidence interval. This is due to a discrepancy between our Laplace approximation and the unapproximated posterior. One key reason for this discrepancy in this scenario is discussed in Section 7.1.

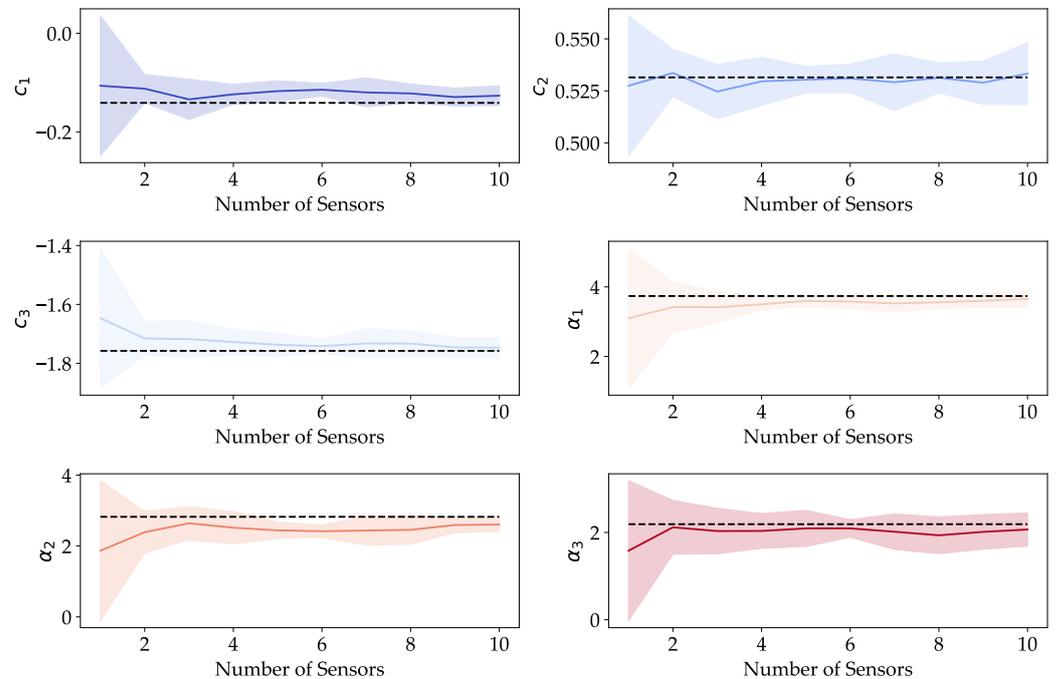


Figure 9. MAP estimate (solid line) and 99% confidence intervals (shaded region) for each parameter versus the number of sensors used, in the optimal locations. The true value of each parameter is indicated with a dashed line.

7. Discussion and Conclusions

We end this paper by discussing the limitations of this methodology and potential future work.

7.1. Non-Differentiability

The Laplace approximation of the posterior, along with all gradient-based optimisation techniques that are used to find the MAP estimate, relies on the RTM process being differentiable. In this section, we discuss how this assumption only holds under specific experiments.

When pressure measurements are extracted at fixed, equal time intervals within a pre-defined period, the pressure data from a given sensor can be split into three distinct phases:

1. Before the resin reaches the sensor, the sensor gives 0 pressure (with noise).
2. Once the resin reaches the sensor, the pressure increases from 0 in a decaying, quasi-logarithmic fashion.
3. When the resin has almost filled the mould, the limited size of the exit vent means that the boundary condition on the resin changes from a Dirichlet boundary to a zero-flow Neumann boundary for the large majority of the boundary. This rapid change in the boundary condition causes a large and sudden increase in pressure throughout the mould. This change is sometimes so quick that it is ‘almost discontinuous’.

This is illustrated in Figure 10. The three phases are clearly visible here. The ‘almost discontinuity’ that appears in the posterior comes from the following: if, in the example shown in Figure 10, there was a measurement taken at $t = 23,000$ s, then a slight increase in any of the components in the permeability vector \mathbf{k} from either the bulk permeability or from the race tracking will cause the pressure measured here to rapidly change from roughly

0.2 to roughly 0.35. This effect is more prominent in models with a small exit vent, where there is a large change in boundary conditions as the resin reaches the end. The scenario in Figure 10 is one with a very small but realistic (5 mm wide) exit vent—smaller than what is used in the numerical examples in this paper. However, the effect still exists, to a lesser extent, for larger vents.

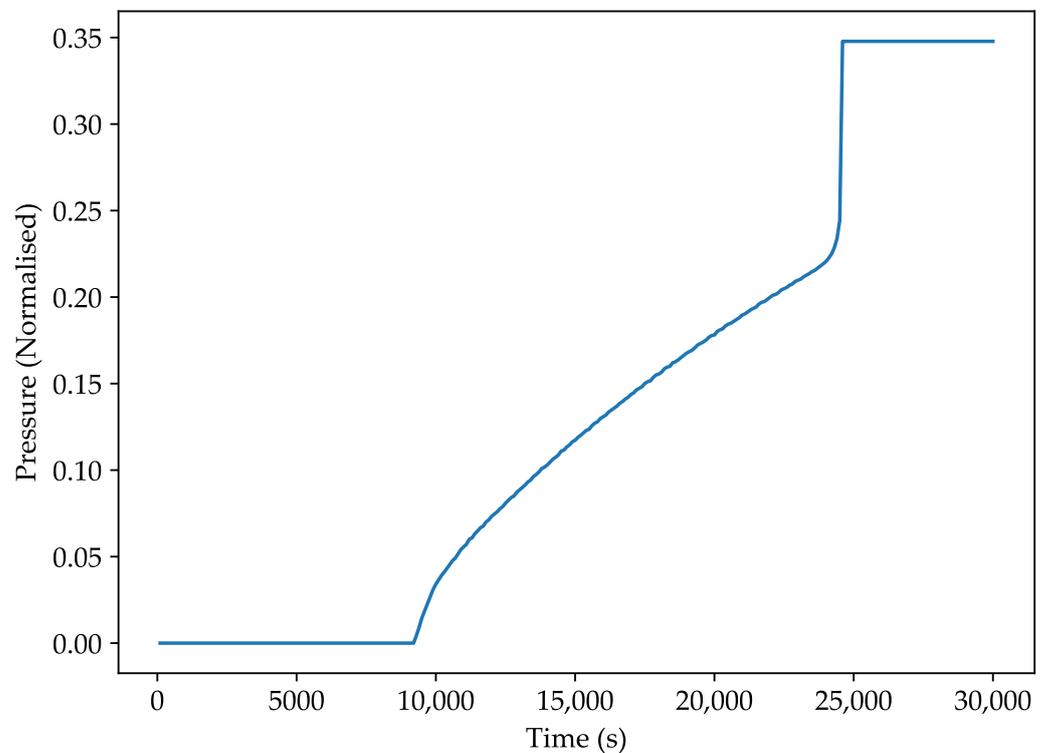


Figure 10. An example of pressure measurements over time from a sensor located at (0.45 m, 0.175 m) in Model 1 (blue). This simulation uses a small 5 mm vent.

Figure 11 shows the true posterior (without using the Laplace approximation) for a much simpler formulation of the problem, where the only parameter to predict is $K = K_{xx} = K_{yy}$ for a homogeneous domain, with $K_{xy} = 0$. Again, a central block is included, as in Section 4.1. The true value for this parameter is one. The inverse problem is solved with four different outlet boundary conditions: a single node, the top 0.1 m of the right boundary, the top 0.25 m of the right boundary, and the entire right boundary. As the outlet vent becomes smaller, a nondifferentiable point develops around $K = 1.02$. $K = 1.02$ is a point that exhibits the behaviour discussed above, where increasing this value a small amount causes one measurement to be taken after the resin has reached the end of the mould, instead of before. This point still exists even when the outlet is as large as 0.25 m. The roughness of the posterior shape aside from this discontinuity should be ignored, it is the product of using a coarse mesh in order to develop the true posterior within an acceptable time frame.

The non-differentiability causes two key limitations to our results:

- It increases the discrepancy between the Laplace approximation and the true posterior, as seen in Figure 11 and in the results for Model 1, where the true values were outside of the Laplace approximation.
- Finding the MAP estimate through optimisation techniques becomes much more difficult, as the function becomes nonsmooth.

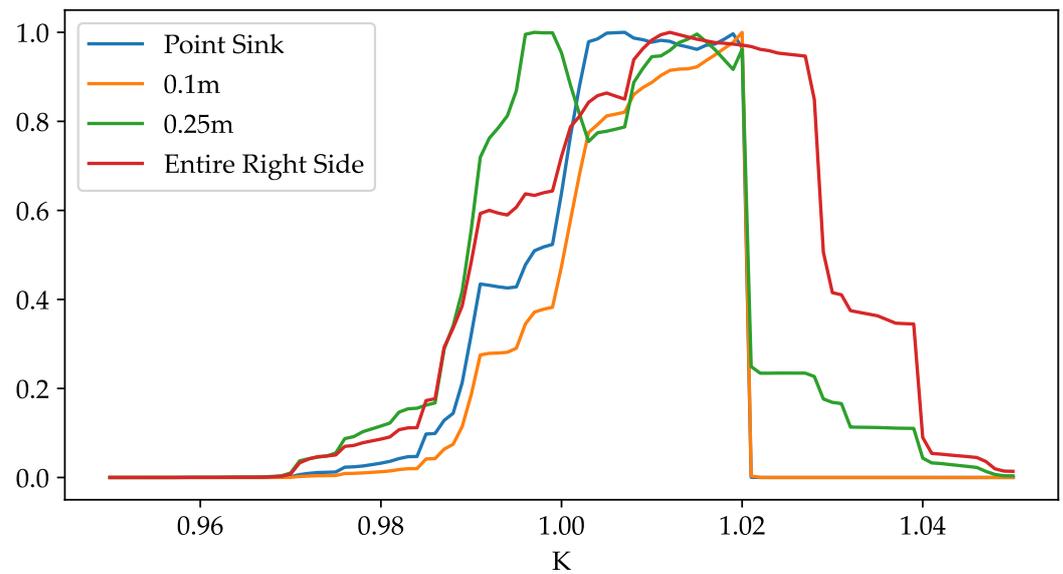


Figure 11. The true posteriors visualised for a one-parameter formulation.

For Model 1, the nondifferentiable points are far away from the MAP estimate, so there is no noticeable effect on the posterior. However, the optimisation to find the MAP estimate is long and difficult due to these nondifferentiable regions. For Model 2, we use a large exit vent to reduce the effects of this problem, speeding up the optimisation, and ensuring that the posterior is not affected.

For the numerical examples in this paper, we can minimise this issue by measuring only while the mould is filling. In a potential future case where we wish to estimate race tracking during the Resin Injection stage, it would not be useful to estimate after the mould has already been filled. However, when we consider more complicated geometries, there could be scenarios in which the resin reaches the end of a channel but the mould is not yet filled. So, the nondifferentiable issues mainly arise at the end of filling for our examples; this is not the case in general for all geometries.

Even in cases where our exact methodology is not used, the non-differentiability should be noted in all research where a Jacobian or gradient of the RTM process is considered, as differentiability is often assumed.

7.2. Conclusions

By representing the permeability throughout a preform as a homogeneous but anisotropic property, except for regions of race tracking, we are able to predict the race-tracking effect as well as the bulk permeability to a high level of certainty. A Bayesian inversion algorithm was used to achieve this and was demonstrated with two example scenarios that showed that the precision of the estimates increased with fewer race-tracking regions and when more sensors were used. Although these models are simple and only two-dimensional, the approach could be extended to three-dimensional, with many race-tracking regions.

In this paper, we only estimated the permeability once the mould had been filled and measurements had been collected throughout the filling process. However, in the future, this methodology could be applied through different stages of mould filling. This would allow researchers to estimate race tracking and then predict the flow front propagation pattern for the remainder of the filling. It would then be possible to take some control actions if we predict that the resin will propagate in an unfavourable way.

The uncertainty in the computed posterior distributions developed here considers only measurement error from sensors, where the measurement from the sensor is made noisy by the addition of randomly and independently sampled error. However, there are approaches that deal with another significant type of error—the approximation errors that arise due to the use of an approximate model, which differs from the true RTM process. There are

approaches to deal with this so-called Bayesian approximation error [41–43] that could be investigated and integrated into the current framework in the future. In particular, this paper made several simplifying assumptions to the model, such as a piecewise homogeneous permeability and fixed porosity, which add to this Bayesian approximation error.

We have also used the Bayesian formulation to develop a methodology for optimising sensor placement within a mould to extract maximum insight. This investigation also revealed the decaying value of adding more sensors. The optimal sensor configuration showed significant decreases in uncertainty compared to random locations. In the future, this framework could be combined with an online permeability estimation tool for increased predictive power. The results show that using a large number of sensors with a tedious, disruptive, and potentially expensive setup is not necessary—very similar accuracy can be achieved with between five and seven sensors, provided that they are in optimal locations. The investigation has also shown that even two sensors can provide reasonably accurate estimates for race tracking for simple scenarios.

Author Contributions: Conceptualization, N.W. and S.A.; methodology, N.W., P.K., O.M. and R.N.; software, N.W.; validation, N.W.; formal analysis, N.W.; resources, P.K.; writing—original draft preparation, N.W.; writing—review and editing, N.W., R.N., P.K. and S.A.; visualization, N.W.; supervision, P.K., O.M., R.N. and S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Finite Element Derivation

Applying the Galerkin finite element method to Equation (2), where R are the residuals from using an approximate p in the equation:

$$\int_{\Omega} R w d\Omega = 0, \quad (\text{A1})$$

$$\int_{\Omega} w \nabla \cdot (\mathbf{K} \nabla p) d\Omega = 0, \quad (\text{A2})$$

where Ω represents the total domain. Applying the Green–Gauss theorem, we have:

$$\int_{\Omega} \nabla w \cdot (\mathbf{K} \nabla p) d\Omega = \int_{\Gamma} \mathbf{K} \frac{\partial p}{\partial n} w d\Gamma, \quad (\text{A3})$$

where Γ is the complete domain boundary. We use a linear interpolation scheme with triangular elements, which is defined as follows:

$$p = \sum_n p_n \varphi_n, \quad w = \sum_m \varphi_m, \quad (\text{A4})$$

$$\nabla p = \sum_n p_n \nabla \varphi_n, \quad \nabla w = \sum_m \nabla \varphi_m, \quad (\text{A5})$$

where φ_n is the linear Lagrange basis function for the n th node in the domain. This gives:

$$\mathbf{E} p = f, \quad (\text{A6})$$

where \mathbf{p} is the vector of nodal pressures and \mathbf{E} and \mathbf{f} are defined element-wise as follows:

$$\mathbf{E}_{mn} = \int_{\Omega} \nabla \varphi_m \cdot \mathbf{K} \nabla \varphi_n d\Omega, \tag{A7}$$

$$\mathbf{f}_m = \int_{\Gamma} \mathbf{K} \frac{\partial p}{\partial n} \varphi_m d\Gamma. \tag{A8}$$

Transforming to the local coordinates (ξ_1, ξ_2) :

$$\mathbf{E}_{mn} = \int_0^1 \int_0^1 \mathbf{J}^{-1} \nabla_{\xi} \varphi_m \cdot \mathbf{K} \mathbf{J}^{-1} \nabla_{\xi} \varphi_n |\mathbf{J}| d\xi_1 d\xi_2, \tag{A9}$$

where ∇_{ξ} is the gradient in terms of the local coordinates ξ_1 and ξ_2 , and:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi_1} & \frac{\partial y}{\partial \xi_1} \\ \frac{\partial x}{\partial \xi_2} & \frac{\partial y}{\partial \xi_2} \end{bmatrix}. \tag{A10}$$

Appendix B. Model 2 Optimal Experimental Design

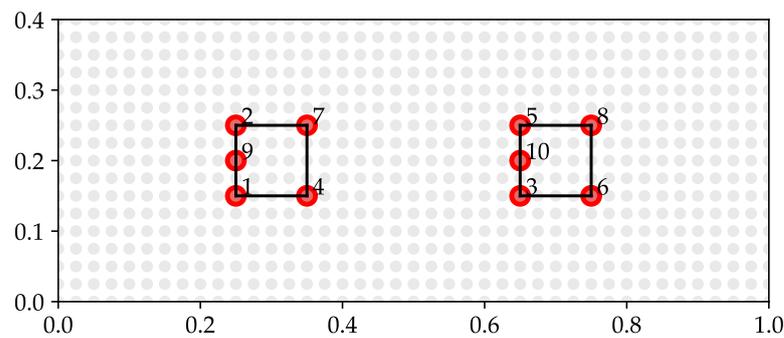


Figure A1. Numbered optimal sensor locations, as determined by the greedy algorithm. Sensors are numbered in the order they appear in the greedy algorithm—i.e., the optimal locations of 4 sensors are locations 1, 2, 3, and 4. All candidate locations are shown in grey.

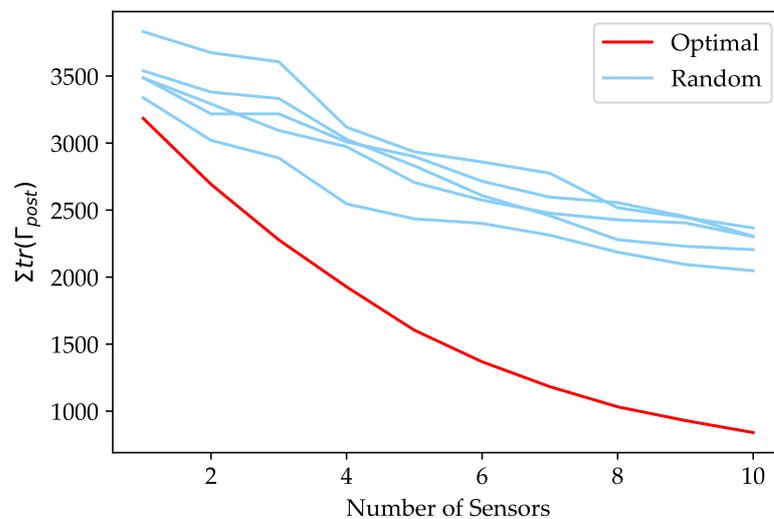


Figure A2. Sum of the covariance trace over all simulations versus the number of sensors used. The optimal solution is in red, and random solutions are in blue.

Appendix C. Number of Training Samples

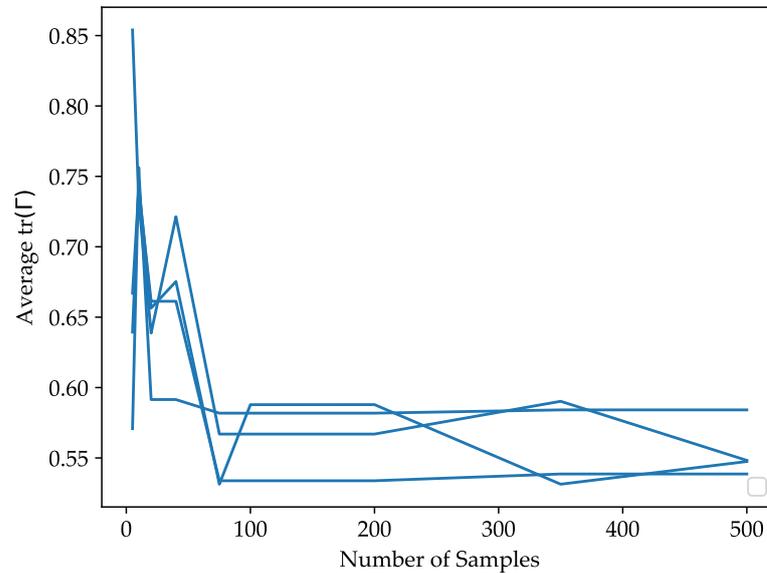


Figure A3. Average covariance trace in the training set vs. the test set, when an increasing number of training samples are considered.

In our investigation, we considered 200 different permeability scenarios in which to calculate the trace of the posterior covariance matrix for a given sensor layout. There is a trade-off to this number. If this number is increased, it allows the ensemble of scenarios to better represent all the potential scenarios that could possibly occur, making the sensor design far more robust. Clearly, this is beneficial for the design of locations that work well regardless of the exact permeability scenario. However, increasing this number also increases the computation time significantly.

To determine what this number should be, an experiment was conducted using Model 1. We considered solving the optimal sensor location problem, using a different amount of permeability scenarios (values for M). We then consider the average trace of the posterior covariance matrix for 100 different scenarios. These are testing scenarios, which are fixed as we increase M , and the optimal sensor locations potentially change. We compare the setup against scenarios other than what the setup was ‘trained’ on in order to determine whether the sensor locations are still optimal in conditions outside of what it was directly optimised on. This is the objective, as it shows that the optimal sensor design is robust, regardless of permeability. This is calculated as follows (a variation of Equation (42)):

$$\frac{1}{100} \sum_{i=1}^{100} \text{tr}(\Gamma_{\text{post}}^S(k_i)) \quad (\text{A11})$$

The results of conducting this investigation four times from $M = 10$ to $M = 500$ are shown in blue in Figure A3. With small numbers for M , the location of the sensors changes greatly because the configuration is not yet robust, and adding more training samples causes the optimal locations to change. This causes rapid changes in the testing metric. The sensor locations developed using these low numbers of samples perform worse in the test set compared to those trained on a larger amount of samples. When the optimal locations are based on only a few samples, they will be over-fitted to those samples and perform poorly in the test set. The testing metric appears to remain relatively constant (i.e., the optimal sensor locations are changing minimally) after around 75–100 samples. In this paper, we use $M = 200$ to be safe while not creating any excessive run time.

References

1. Advani, S.G.; Sozer, E.M. *Process Modeling in Composites Manufacturing*; CRC Press: Boca Raton, FL, USA, 2003. [CrossRef]
2. Zhu, H.; Li, D.; Zhang, D.; Wu, B.; Chen, Y. Influence of voids on interlaminar shear strength of carbon/epoxy fabric laminates. *Trans. Nonferrous Met. Soc. China* **2009**, *19*, s470–s475. [CrossRef]
3. Varna, J.; Joffe, R.; Berglund, L.; Lundström, T. Effect of voids on failure mechanisms in RTM laminates. *Compos. Sci. Technol.* **1995**, *53*, 241–249. [CrossRef]
4. Mehdikhani, M.; Gorbatikh, L.; Verpoest, I.; Lomov, S.V. Voids in fiber-reinforced polymer composites: A review on their formation, characteristics, and effects on mechanical performance. *J. Compos. Mater.* **2019**, *53*, 1579–1669. [CrossRef]
5. Mathieu Devillard, Kuang-Ting Hsiao, A.G.; Advani, S.G. On-line characterization of bulk permeability and race-tracking during the filling stage in resin transfer molding process. *J. Compos. Mater.* **2003**, *37*, 1525–1541. [CrossRef]
6. Bickerton, S.; Advani, S.; Mohan, R.V.; Shires, D. Experimental analysis and numerical modeling of flow channel effects in resin transfer molding. *Polym. Compos.* **2000**, *21*, 134–153. [CrossRef]
7. Lawrence, J.M.; Barr, J.; Karmakar, R.; Advani, S.G. Characterization of preform permeability in the presence of race tracking. *Compos. Part A Appl. Sci. Manuf.* **2004**, *35*, 1393–1405. [CrossRef]
8. Agogué, R.; Shakoob, M.; Beauchêne, P.; Park, C.H. Analysis and minimization of race tracking in the resin-transfer-molding process by Monte Carlo simulation. *Materials* **2023**, *16*, 4438. [CrossRef]
9. Caglar, B.; Salvatori, D.; Sozer, E.M.; Michaud, V. In-plane permeability distribution mapping of isotropic mats using flow front detection. *Compos. Part A Appl. Sci. Manuf.* **2018**, *113*, 275–286. [CrossRef]
10. Fernández-León, J.; Keramati, K.; Garoz, D.; Baumela, L.; Miguel, C.; González, C. A machine learning strategy for race-tracking detection during manufacturing of composites by liquid moulding. *Integr. Mater. Manuf. Innov.* **2022**, *11*, 296–311. [CrossRef]
11. Koutsonas, S. Race-Track Modelling and Variability in RTM for Advanced Composites Structures. Ph.D. Thesis, The University of Nottingham, Nottingham, UK, 2015.
12. Siddig, N.; Binetruy, C.; Syerko, E.; Simacek, P.; Advani, S. A new methodology for race-tracking detection and criticality in resin transfer molding (RTM) process using pressure sensors. *J. Compos. Mater.* **2018**, *52*, 4087–4103. [CrossRef]
13. Devillard, M.; Hsiao, K.T.; Advani, S.G. Flow sensing and control strategies to address race-tracking disturbances in resin transfer molding—Part II: Automation and validation. *Compos. Part A Appl. Sci. Manuf.* **2005**, *36*, 1581–1589. [CrossRef]
14. Tarantola, A. *Inverse Problem Theory and Methods for Model Parameter Estimation*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2005. [CrossRef]
15. Jari, P.; Kaipio, E.S. *Statistical and Computational Inverse Problems*; Springer: New York, NY, USA, 2005. [CrossRef]
16. Garnett, R. *Bayesian Optimization*; Cambridge University Press: Cambridge, UK, 2023. [CrossRef]
17. Yuen, K.V.; Kuok, S.C. Bayesian Methods for Updating Dynamic Models. *Appl. Mech. Rev.* **2011**, *64*, 010802. [CrossRef]
18. Iglesias, M.; Park, M.; Tretyakov, M.V. Bayesian inversion in resin transfer molding. *Inverse Probl.* **2018**, *34*, 105002. [CrossRef]
19. Matveev, M.; Endruweit, A.; Long, A.; Iglesias, M.; Tretyakov, M. Bayesian inversion algorithm for estimating local variations in permeability and porosity of reinforcements using experimental data. *Compos. Part A Appl. Sci. Manuf.* **2021**, *143*, 106323. [CrossRef]
20. Hsiao, K.T.; Advani, S.G. Flow sensing and control strategies to address race-tracking disturbances in resin transfer molding. Part I: Design and algorithm development. *Compos. Part A Appl. Sci. Manuf.* **2004**, *35*, 1149–1159. [CrossRef]
21. Sozer, E.M.; Bickerton, S.; Advani, S.G. On-line strategic control of liquid composite mould filling process. *Compos. Part A Appl. Sci. Manuf.* **2000**, *31*, 1383–1394. [CrossRef]
22. Shojaei, A.; Ghaffarian, S.R.; Karimian, S.M.H. Modeling and simulation approaches in the resin transfer molding process: A review. *Polym. Compos.* **2003**, *24*, 525–544. [CrossRef]
23. Gauvin, R.; Chibani, M. The Modelling of Mold Filling in Resin Transfer Molding. *Int. Polym. Process.* **1986**, *1*, 42–46. [CrossRef]
24. Song, X. *Vacuum Assisted Resin Transfer Molding (VARTM): Model Development and Verification*; Virginia Polytechnic Institute and State University: Blacksburg, VA, USA, 2003. Available online: <https://www.proquest.com/docview/305301229> (accessed on 24 September 2023).
25. Bruschke, M.V.; Advani, S.G. A finite element/control volume approach to mold filling in anisotropic porous media. *Polym. Compos.* **1990**, *11*, 398–405. [CrossRef]
26. Kang, M.; Jung, J.; Lee, W.I. Analysis of resin transfer moulding process with controlled multiple gates resin injection. *Compos. Part A Appl. Sci. Manuf.* **2000**, *31*, 407–422. [CrossRef]
27. Lee, L.; Young, W.; Lin, R. Mold filling and cure modeling of RTM and SRIM processes. *Compos. Struct.* **1994**, *27*, 109–120. [CrossRef]
28. Simacek, P.; Advani, S.; Binetruy, C. Liquid injection molding simulation (LIMS) a comprehensive tool to design, optimize and control the filling process in liquid composite molding. *JEC-Compos* **2004**, *8*, 58–61.
29. Šimáček, P.; Advani, S.G. Desirable features in mold filling simulations for liquid composite molding processes. *Polym. Compos.* **2004**, *25*, 355–367. [CrossRef]
30. Guan Koay, C. Least squares approaches to diffusion tensor estimation. In *Diffusion MRI: Theory, Methods, and Applications*; Oxford University Press: Oxford, UK, 2010; pp. 272–284. [CrossRef]
31. Revels, J.; Lubin, M.; Papamarkou, T. Forward-mode automatic differentiation in Julia. *arXiv* **2016**, arXiv:1607.07892.
32. Nocedal, J.; Wright, S.J. *Numerical Optimization*; Springer: New York, NY, USA, 2006. [CrossRef]

33. Ucinski, D. *Optimal Measurement Methods for Distributed Parameter System Identification*; CRC Press: Boca Raton, FL, USA, 2004.
34. Alexanderian, A. Optimal experimental design for infinite-dimensional Bayesian inverse problems governed by PDEs: A review. *arXiv* **2021**, arXiv:2005.12998.
35. Alexanderian, A.; Petra, N.; Stadler, G.; Ghattas, O. A fast and scalable method for A-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems. *SIAM J. Sci. Comput.* **2016**, *38*, A243–A272. [[CrossRef](#)]
36. Alexanderian, A.; Nicholson, R.; Petra, N. Optimal design of large-scale nonlinear Bayesian inverse problems under model uncertainty. *arXiv* **2022**, arXiv:2211.03952.
37. Koval, K.; Alexanderian, A.; Stadler, G. Optimal experimental design under irreducible uncertainty for linear inverse problems governed by PDEs. *Inverse Probl.* **2020**, *36*, 075007. [[CrossRef](#)]
38. Chaloner, K.; Verdinelli, I. Bayesian experimental design: A review. *Stat. Sci.* **1995**, *10*, 273–304. [[CrossRef](#)]
39. Wu, K.; Chen, P.; Ghattas, O. A fast and scalable computational framework for large-scale and high-dimensional Bayesian optimal experimental design. *arXiv* **2020**, arXiv:2010.15196.
40. Haber, E.; Horesh, L.; Tenorio, L. Numerical methods for experimental design of large-scale linear ill-posed inverse problems. *Inverse Probl.* **2008**, *24*, 055012. [[CrossRef](#)]
41. Kaipio, J.; Kolehmainen, V. Approximate marginalization over modeling errors and uncertainties in inverse problems. In *Bayesian Theory and Applications*; Oxford University Press: Oxford, UK, 2013; pp. 644–672.
42. Nicholson, R.; Petra, N.; Kaipio, J.P. Estimation of the Robin coefficient field in a Poisson problem with uncertain conductivity field. *Inverse Probl.* **2018**, *34*, 115005. [[CrossRef](#)]
43. Kennedy, M.C.; O’Hagan, A. Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2001**, *63*, 425–464. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.