# MOCU for block-copolymer experimental design

Anthony DeGennaro

April 2019

## General Problem Description

- We have a "real-life" physical system that maps feature inputs to outputs: $y = f_r(x)$. Because of either measurement noise and/or stochasticity in $f_r$ (the latter of which could arise e.g. from physical processes in $f_r$ that involve states/dynamics that we have not captured with $x$), the output of $f_r$ given $x$ may not be completely deterministic, and so we characterize the output of the real system with the conditional probability distribution $\rho_r(y|x)$.
- Because it is expensive to query this system, we have a computationally cheaper model that approximates its behavior: $\hat{y} = f_m(x, \theta)$, with $\theta \sim \rho(\theta)$. As before, if we wish to make this model non-deterministic, we may do so by constructing the probability distribution $\rho_m(y|x, \theta)$.
- The parameters $\theta$ capture any uncertainties in the model structure. If the model $f_m$ is a physical model, then $\theta$ would consist of uncertain parameters appearing in the model dynamics. If instead the model is data-derived (e.g., POD-Galerkin, DMD/Koopman, spectral methods), then $\theta$ could simply capture statistical uncertainty in the weights/coefficients. As an example, if we had $f_m(\theta, x) = \sum_j \theta_j \phi_j(x)$ for some basis functions defined on $x$, then $\theta$ would simply represent the random coefficients of that expansion.
- We can think of $\theta$ as representing our ignorance in how the real system is related to the model system. That is, we assume that the real system *should* be described accurately by one of the candidate models represented by variations of $\theta$, but we do not know which specific values of $\theta$ produce that agreement.
- Our goal w.r.t. operator design is to build a function $\psi_{IBR}(x, \theta) : X \times \Theta \mapsto Y$ from a family of functions $\Psi$ that does the "best" job approximating the model $f_m$ on average over the uncertainty $\theta$. For example, we could use neural networks and think of $\Psi$ as the space of neural networks with a certain structure and number of weights. We wish to find the optimally-robust mapping:

  $\psi_{IBR}(x) = \text{argmin}_{\psi \in \Psi} \mathbb{E}_\theta[C(\theta, \psi)]$

  where $C(\theta, \psi)$ is a cost function that quantifies the discrepancy between predictions made by $\psi$ and $f_m$
- At the same time, we have model uncertainty *vis-a-vis* $\rho(\theta)$ that we wish to reduce by sampling the real system and updating our prior belief about $\theta$ to produce the posterior $\rho(\theta|D)$ through Bayesian inference. We don't want to select just any experiment though; we want to select that experiment $D = (x^*, y^*)$ that also reduces our cost. This leads (after "some algebra") to the MOCU framework for experiment selection:

  $x^* = \text{argmin}_{x \in X} \mathbb{E}_y[\mathbb{E}_{\theta|y}[C(\theta, \psi_{IBR}^{\Theta|x,y})]]$
- The nice thing about this framework is we are (1) designing experiments that respect an objective, (2) tuning a low-dimensional model to more accurately represent reality over the span of those objective-driven experiments, and (3) constructing a function that best represents the input/output mapping on average over all uncertainty, all in one shot.

## Details Specific to Our Setting

1. **Ground Truth Source**
   - We should start with a computational model (Cahn-Hilliard) as our ground truth source. In the future, we will hopefully shift to considering actual experimental data for this purpose. However, for an initial proof-of-concept, we should start here.
   - The features $x \in X$ of Cahn-Hilliard are comprised of the material-specific parameters that appear in the dynamics. These parameters include the interface thickness parameter, the shape/form of the potential function, and other material constants. We will consult the materials-science literature in order to identify physically-meaningful ranges/distributions for these.
   - We may make the system non-deterministic by adding noise to the dynamics, i.e. $\dot{x} = \mathcal{C}(x) + \mathcal{N}$, where $N \sim \rho(\mathcal{N})$ is some noise profile and $\mathcal{C}(\cdot)$ simply denotes the (deterministic) CH dynamics.
   - We should begin by assuming known, fixed initial/boundary conditions, so as not to complicate things. If we want to consider a range of initial/boundary conditions, then probably we will have to incorporate these into the feature (experiment) space $X$ via some parameterization.
   - W.r.t. numerics, we should probably use Danial Faghihi-Shahrestani's (UT) code. If we cannot do that, I (Anthony DeGennaro, BNL) have a 2-D spectral solver, although that would be non-ideal for a variety of reasons.
2. **Low-Dimensional Model**
   - The cheap model should be fitted prior to MOCU-based sampling using some $k$ training data pairs $D_{train} = \{(x_1, \ldots, x_k), (y_1, \ldots, y_k)\}_{train}$, collected from Cahn-Hilliard. This model could be constructed in a variety of ways, depending on how we do things. POD/POD-Galerkin would be classical choices, and DMD/Koopman methods would be an interesting alternative. Karen Wilcox (UT) and Anthony DeGennaro (BNL) could investigate these and other approaches.
   - We should fit a "mean" model to the training data: $\hat{y} = f_m(x, \theta_{fit})$, where $\theta_{fit}$ represent some weights (or coefficients) associated with the model fit
   - To account for model imperfections etc., we can "fuzzify" the model with uncertainty and consider the parameterized class of models $\hat{y} = f_m(x, \theta + \mathcal{N})$ with $\theta \sim \rho(\theta)$ and $\mathcal{N} \sim \rho(\mathcal{N})$. $\theta$ accounts for uncertainty in the model structure; $\mathcal{N}$ is just non-deterministic noise that makes the system stochastic.
   - $\rho(\theta)$ should be based on our prior expectations. For example, the mean value should be at $\theta_{fit}$. If we are using a POD-based or spectral type method, then we might also expect exponential decay in the variance of coefficients for higher-order modes.
3. **Intrinsically Bayesian Robust Operator**
   - We should use some sort of regressor for $y = \psi(x, \theta)$, e.g. a fully-connected neural network
   - The difference between $f_m(x, \theta)$ and $\psi(x, \theta)$ is that the computational model is a low-dimensional model that has been trained to approximate the physics, whereas $\psi$ is just a function that maps $(x, \theta)$ to $y$. For example, if we use POD for $f_m$, then we have $\hat{y} = \sum_j \theta_j \phi_j(x) + \mathcal{N}$ and we will still have to drive the approximate system dynamics to steady-state to get $\hat{y}$, whereas $\psi$ just gives a direct mapping. Also note that in the MOCU machinery, we will need to compute $\psi(\Theta|(x, y))$, which is the optimal regressor that approximates $f_m$ given $(x, y)$, for all combinations of $(x, y) \in X \times Y$. This will result in a different robust operator for each pair of $(x, y)$
   - Obtaining $\psi$ could be done in the usual way, e.g. training a neural net on a set of data generated by the ROM. For example, to approximate $\psi(\Theta|(x, y))$, we would train a neural network on a subset of $k$ data points generated from the ROM using $(x, y; \theta_1 \ldots \theta_k)$
4. **MOCU Methodology**
   - Ed Dougherty and Guang Zhao (A&M) have recently done a derivation showing how the MOCU sampling formula reduces from the general form presented in these notes to something else by marginalizing over $\Theta$, under mild assumptions about $X, Y, \Theta, \Psi$. As far as I can tell, these assumptions are perfectly valid and I defer to their presentation/algorithm for specific details.

# MOCU Methodology

Anthony DeGennaro

April 2019

1. **Cost and Optimality**
   - Assume we have a cost function $C(\theta, \psi)$ that quantifies a cost related to our experimental design objective
   - Our experimental design, then, is to seek a $\psi^{\Theta}_{IBR} \in \Psi$ that is optimal on average over all $\Theta$ w.r.t. this cost:

     $$\mathbb{E}_\theta[C(\theta, \psi^{\Theta}_{IBR})] \leq \mathbb{E}_\theta[C(\theta, \psi)] \quad \forall \psi \in \Psi$$

   - Of course, if we had perfect knowledge of $\theta$, then we could design a classifier $\psi_\theta$ for that specific value that would almost certainly be better than $\psi^{\Theta}_{IBR}$. We can quantify this by computing the cost difference between the two choices, averaged over all of $\Theta$, called the MOCU:

     $$M_\Psi(\Theta) = \mathbb{E}_\theta[C(\theta, \psi^{\Theta}_{IBR}) - C(\theta, \psi_\theta)]$$

     where $\psi_\theta \in \Psi$ denotes the classifier that is optimal for the particular choice of $\theta$. Thus, we should choose experiments in a way that seeks to minimize this MOCU.

2. **Adaptive Experimental Selection**
   - Given a new piece of data $(x, y) \in (x, y)$, we can compute a new MOCU conditioned on that new piece of information. For ease of notation, let $\xi = (x, y)$, and hence $\mathbb{E}_\xi[\cdot]$ refers to the expectation over $\rho(y|x)$ (i.e., the probability that $y$ occurred, given x):

     $$M_\Psi(\Theta|\xi) = \mathbb{E}_{\theta|\xi}[C(\theta, \psi^{\Theta}_{IBR}) - C(\theta, \psi_\theta)]$$

   - Averaging this over many experiments gives the average conditional MOCU:

     $$D_\Psi(\Theta, \xi) = \mathbb{E}_\xi[M_\Psi(\Theta|\xi)]$$

   - The experiment $x^*$ that minimizes this quantity is said to be optimal:

     $$\begin{aligned} x^* &= \text{argmin}_{x \in X}\ D_\Psi(\Theta, \xi) \\ &= \text{argmin}_{x \in X}\ \mathbb{E}_\xi[\mathbb{E}_{\theta|\xi}[C(\theta, \psi^{\Theta}_{IBR}) - C(\theta, \psi_\theta)]] \end{aligned}$$

   - Because $x^*$ also minimizes the quantity $D_\Psi(\Theta, \xi) - M_\Psi(\Theta)$, one can show after some algebra that it also minimizes this quantity:

     $$x^* = \text{argmin}_{x \in X}\ \mathbb{E}_\xi[\mathbb{E}_{\theta|\xi}[C(\theta, \psi^{\Theta|\xi}_{IBR})]] - \mathbb{E}_\theta[C(\theta, \psi^{\Theta}_{IBR})]$$

   - And, because the cost $C(\theta, \psi)$ does not vary with $x$, we may eliminate it to obtain:

     $$x^* = \text{argmin}_{x \in X}\ \mathbb{E}_\xi[\mathbb{E}_{\theta|\xi}[C(\theta, \psi^{\Theta|\xi}_{IBR})]]$$

3. **MOCU-Specific Calculus**
   - The equation for $x^*$ involves the double-nested expectation $\mathbb{E}_\xi[\mathbb{E}_{\theta|\xi}[\cdot]]$
   - Outer loop: $\mathbb{E}_\xi[F] = \int_\xi F(\xi)\rho(\xi)d\xi$ where:

     $$\rho(\xi) = \mathbb{E}_\theta[\rho(\xi|\theta)] = \int_\theta \rho(\xi|\theta)\rho(\theta)d\theta$$

     Thus, we must know $\rho(\xi|\theta)$ (or be able to sample from it e.g. with a computer model) a priori (assumption)

- Inner loop: $\mathbb{E}_{\theta|\xi}[F] = \int_{\xi} F(\xi)\rho(\theta|\xi)d\xi$ where, by Bayes' law:

$$\rho(\theta|\xi) = \frac{\rho(\xi|\theta)\rho(\theta)}{\rho(\xi)}$$

  Thus, we must also know $\rho(\theta)$ (uncertain parameter distribution) a priori (assumption)

4. **Software Algorithmic Implementation**

**inputs** : set of $\Theta = \{\theta_1 \dots \theta_k\}$ with $\rho(\Theta)$ , computer model $\hat{y} = f_m(x, \theta)$, set of experimental choices $X = \{x_1 \dots x_n\}$ with possible outcomes $Y = \{y_1 \dots y_m\}$

Monte Carlo sample the computer model over $X \times Y \times \Theta$ and approximate $\rho(Y|X, \Theta)$

for all $x_i$ in $X$:

for all $y_j$ in $Y$:

Compute $\rho(\theta|x_i = y_j)$ via Bayes with priors $\rho(\theta)$, $\rho(y_j|x_i, \theta)$

Compute $\psi(\Theta|x_i = y_j) = \text{argmin}_{\psi}\ \mathbb{E}_{\theta|x_i=y_j}[C(\theta, \psi)]$

Compute $\omega(x_i = y_j) = \mathbb{E}_{\theta|x_i=y_j}[C(\theta, \psi(\Theta|x_i = y_j))]$

Compute $\mathbb{E}_{y|x_i}[\omega(x_i)]$ (i.e., averaged over the $m$ outcomes in $Y$)

Compute $x^* = \text{argmin}_X\ \mathbb{E}_{y|x}[\omega(x)]$ (i.e., minimization over the $n$ experiments in $X$)

**outputs** : $x^*$