# Reducing and Calibrating for Input Model Bias in Computer Simulation

Lucy E. Morgan, Luke Rhodes-Leader

Lancaster University Management School and STOR-i Centre for Doctoral Training

Lancaster University, Lancaster, LA1 4YR, UK

Russell R. Barton

Department of Supply Chain and Information Systems,

The Pennsylvania State University

University Park, PA 16802, USA

Input model bias is the bias found in the output performance measures of a simulation model caused by estimating the input distributions/ processes used to drive it. To be specific, when input models are estimated from a finite amount of real-world data they contain error and this error propagates through the simulation to the outputs under study. When the simulation response is a non-linear function of its inputs, as is usually the case when simulating complex systems, input modelling bias is one of the errors to arise. In this paper we introduce a method that re-calibrates the input parameters of parametric input models to reduce the bias in the simulation output. The method is shown to be successful in reducing input modelling bias and the total mean squared error caused by input modelling.

## 1 Introduction

In stochastic simulation the simulation model is driven by input distributions or processes. These inputs are often constructed using real-world observations from the system under study. As we can never collect an infinite number of observations the input distributions are always an approximation of the true unknown input distributions. The error in the input distributions and processes propagates through the simulation model to error in the output/s of the simulation model, often measured as mean squared error. Mean squared error can be broken down into i) variance from input sample to input sample, often called input uncertainty (IU); and ii) bias caused by input modelling. Until recently bias caused by input modelling was assumed negligible and ignored. Typically, confidence intervals for simulation output means that take into account input model uncertainty have been symmetric, i.e., do not take into account bias. Lam (2016) noted that the focus of research on variance rather than mean squared error was due to the fact that bias caused by input model error typically decreases at a faster rate than variance. However, Morgan et al. (2019) argued that in

the cases where input uncertainty is of most interest, i.e. when the amount of real-world data available is small, there is no way to say whether bias is irrelevant. Bias can still be large for finite sample size $m$ even though it decreases faster than input uncertainty as $m \to \infty$.

For a non-linear function $\eta(\cdot)$ and a mean-unbiased estimator $\hat{\theta}$ of a parameter $\theta$, the composite estimator $\eta(\hat{\theta})$ will likely not be a mean-unbiased estimator of $\eta(\theta)$. For example, by Jensen's inequality, if $\eta(\cdot)$ is a convex function, uncertainty in $\hat{\theta}$ will introduce positive bias in $\eta(\hat{\theta})$, while if $\eta(\cdot)$ is concave, uncertainty in $\hat{\theta}$ will result in negative bias for $\eta(\hat{\theta})$. The bias depends both on the sampling distribution of the estimator and on the transform, and can be quite involved to calculate in general. In the context of simulation the output is typically a nonlinear function of the input parameters, and intrinsic simulation uncertainty further complicates the bias estimation. Approaches to estimating bias caused by input modelling are summarised by Morgan et al. (2019).

In this paper we seek ways to bias the input parameter estimates to reduce bias in the simulation output. Our concern is not with the error in the parameters themselves but the impact on simulation output error. Given an estimate of the bias caused by input modelling, this paper provides a method to attain an expected value of simulation output equal to the bias-corrected value. The method adjusts the input parameter vector used to drive the simulation by the smallest possible amount (Euclidean metric) to achieve this result. Correction of the input parameters leads naturally to a sequential approach to reducing the bias in the simulation output. This could be thought of as calibrating the input parameters to account for input modelling error. We consider only parametric input distributions. While we assume that the initial parameter estimates are constructed via maximum likelihood, other estimators could be accommodated by the method.

Our overall objective is to reduce the mean squared error (MSE) of one or more simulation outputs, typically expected values of system performance measures. Of course, the method used to adjust input parameter bias must minimize additional variance, or the resulting MSE will not be decreased. By constraining the re-calibrated input parameters to be as close as possible to the original estimates, we hope that any potential addition to the input uncertainty will be small, and that we will observe an overall decrease in MSE. Our computational study shows this to be the case.

Adjustment of input parameters to achieve reduced MSE for simulation output is important in two settings. First, when the simulation model is a submodel of some larger simulation under study, then the re-calibrated input parameters can be used to drive the submodel in studies on the larger system. This will be useful where input parameters and/or structural aspects of other parts of the simulation are being designed or studied. Second, the magnitude of the bias adjustment will serve as an indicator of the most bias sensitive input distribution components. This can focus efforts on input model refinement and additional

data collection.

The remainder of the paper is organized as follows. Prior work in bias identification and reduction in the simulation setting is reviewed in Section 2. This is followed by the description of the proposed bias reduction methodology in Section 3. In Section 4 the performance of the method is then explored in two settings: a simple M/M/1/K queue, and a stochastic activity network. The final section discusses the performance results and potential uses of this bias-adjustment strategy.

## 2 Background

To date, estimating bias caused by input modelling has received little attention in the field of stochastic simulation. This is in part due to the fact that bias caused by input modelling is known to reduce quicker than input uncertainty (IU) as the amount of input data increases. But note that the exact cases where input modelling error is of most concern are the cases where there is a limited amount of input data, and in such cases little can be said about the relative size of bias caused by input modelling compared to IU. Morgan et al. (2019) present a delta method approach to estimating the bias caused by input modelling alongside a test with controlled power for detecting when bias caused by input modelling is of relevant size. They recognise that when bias is small it can be hard to estimate accurately, but relatively small effort is required to conclude whether it is significantly different to zero.

Reichert and Schuwirth (2012) discuss bias in the output of a model caused by error in its input models. They present a Bayesian approach to quantifying bias and calibrating input parameters using multi-objective model calibration techniques, but only consider deterministic models, without the presence of simulation noise. Kennedy and O'Hagan (2001) also present a Bayesian method for calibrating the input parameters of deterministic models that accounts for the possible uncertainties in the parameter estimates. In addition both papers consider bias in the model output caused by using a simplified model of a real-world system, otherwise known as model error. We do not discuss model error in this paper and instead assume that any model under consideration is of high enough fidelity to accurately represent the system of interest.

Other standard methods for bias estimation are the jackknife and the bootstrap (Efron, 1982). Most uses of these estimators have assumed the population parameter of interest can be observed without noise. This is most often not the case in simulation experiments. Withers and Nadarajah (2014) compare the computational efficiency of the delta method, bootstrap and jackknife bias estimates in a scenario without simulation noise. With the addition of simulation noise efficiency of the method of choice is even more important.

In this paper our contribution is not to provide a new estimator of bias caused by input modelling, but to

reduce this form of bias and re-calibrate the parameters of the input models used to drive the simulation. We will therefore focus on a single estimator of bias caused by input modelling throughout the paper. That is, within our evaluation, when the true simulation response $\eta(\cdot)$ is unknown, we use the bootstrap estimator of bias caused by input modelling. This estimator is intuitive and easy to calculate. In Section 3 the bootstrap estimator of bias is introduced.

Estimation of bias caused by input modelling falls within the larger field of input modelling error quantification. Within this field the majority of publications focus on estimating the IU variance. See Barton (2012), Song et al. (2014) and references therein for a discussion of methods on quantifying IU variance in systems with stationary input models.

The bias correction method introduced in this paper uses sequential quadratic programming (SQP) to simualtaneously correct bias caused by input modelling and the input parameters driving the simulation model. See Nocedal and Wright (2006b) for an introduction to sequential quadratic programming.

# 3 Methodology

The general set up in stochastic simulation is for a simulation model to take a set of input distributions, $\boldsymbol{F}$, and system parameters, $\mathbf{x}$, and to output a performance measure/ measures of interest $\boldsymbol{Y}$. In this paper we assume that there are $L$ known parametric input distributions, $\{F_1, F_2, \ldots, F_L\}$, with $P \geq L$ true, but unknown, input parameters $\boldsymbol{\theta}^c = \{\theta_1^c, \theta_2^c, \ldots, \theta_P^c\}$. In this paper we assume we have observations collected from the system of interest with which to estimate the input parameters of the $L$ input distributions. Let $m_l$ denote the number of observations collected from the $l^{th}$ input distribution, we will use maximum likelihood estimation to estimate the parameters of the model. Repeating this for input models $l = 1, 2, \ldots, L$ we gain $\boldsymbol{\theta}^{mle} = \{\boldsymbol{\theta}_1^{mle}, \boldsymbol{\theta}_2^{mle}, \ldots, \boldsymbol{\theta}_L^{mle}\}$. From hereon we will assume that an equal number of observations $m$ is collected from each input distribution $m = m_1 = m_2 = \cdots = m_L$, but note that this is for notational convenience and the following methods generalise to unequal numbers of observations. Since the observations are collected by observing a real-world system there will always be constraints that limit the number of observations that can be collected, for example a study may be limited by time or the cost of data collection. The number of observations $m$ will therefore always be finite, and the estimate input parameters, $\boldsymbol{\theta}^{mle}$ will be approximate. By using $\boldsymbol{\theta}^{mle}$ to drive a simulation experiment the error arising from their estimation passes through the simulation to the output performance measures of interest. This form of error is known as input modelling error and can be broken down into the variance in the simulation response caused by input modelling, IU, and bias caused by input modelling, which is the subject of this paper.

To define bias caused by input modelling we first introduce some standard notation for the simulation output. We consider the simulation output to be the expected value of some performance measure i.e. an aggregate measure of long-run system performance over a large number of replications, $n$. Let us denote the simulation output in replication $j$ at a chosen set of input parameters $\boldsymbol{\theta}$ by

$$Y_j(\boldsymbol{\theta};\mathbf{x}) = \eta(\boldsymbol{\theta};\boldsymbol{x}) + \varepsilon_j \tag{1}$$

where $\eta(\cdot)$ is the expected value of the simulation response, $\mathbf{x}$ are system parameters (e.g. number of servers or queueing discipline), and $\varepsilon_j$ is a mean 0 random variable with finite variance, $\sigma^2$, that represents the simulation noise. Here we assume that $\varepsilon_j$ is not a function of the parameters $\boldsymbol{\theta}$ and $\mathbf{x}$. Note that in reality this is unlikely to be correct, but within our method we make large deviations from the original parameter estimates, $\boldsymbol{\theta}^{mle}$, unattractive, and would therefore expect the simulation variance to vary minimally. For ease of exposition we now drop the system parameters $\mathbf{x}$ from our notation; we will return to discuss them further in Section 6. After $n$ replications of the simulation given input parameters $\boldsymbol{\theta}$, we denote the average output by $\bar{Y}_{\boldsymbol{\theta}}$.

In the context of simulation within Equation (1) there are usually two unknowns: $\boldsymbol{\theta}^c$ the true but unknown input distribution parameters, and $\eta(\cdot)$ the expected simulation response. In this paper we are interested in the bias caused by input modelling which describes how far, on average, our simulation response is from the real-world performance given the error that arises from estimating the input models, (Morgan et al., 2019). Bias caused by input modelling is defined by

$$\beta = \mathbb{E}[\eta(\boldsymbol{\theta}^{mle})] - \eta(\boldsymbol{\theta}^c), \tag{2}$$

where the expectation is with respect to the distribution of $\boldsymbol{\theta}^{mle}$. Unfortunately when $\eta(\cdot)$ and/ or $\boldsymbol{\theta}^c$ are unknown we cannot directly observe $\beta$, and in practice work with estimates, $\hat{\beta}$, calculated from the output of the simulation. When the simulation response is unknown the bias, $\hat{\beta}$, is estimated in the presence of simulation noise.

There are several bias estimators that can be used to estimate the bias caused by input modelling. In Section 2 we introduced the idea of the bootstrap, the jackknife and the delta estimates of bias. Note that our method for the reduction of bias caused by input modelling by recalibrating the simulation input parameters can be used with any estimate of bias caused by input modelling. In Section 4 when we come to evaluate our method we work with the bootstrap estimate of bias and now provide details on the method.

The bootstrap estimate of bias is easy to use in practice and intuitive. To calculate it we take $B$ samples

of size $m$, with replacement, from our original observed data. Given the $B$ data sets we re-estimate the input parameters giving bootstrap estimates denoted here by $\boldsymbol{\theta}_b^\star$, for $b = 1, 2, \ldots, B$. For each set of bootstrap input parameters $\boldsymbol{\theta}_b^\star$ we then complete $n$ runs of the simulation to evaluate $\bar{Y}_{\boldsymbol{\theta}_b^\star}$, and $\bar{\bar{Y}}$ where

$$\bar{Y}_{\boldsymbol{\theta}_b^\star} = \frac{1}{n}\sum_{j=1}^{n} Y_j(\boldsymbol{\theta}_b^\star)$$
$$\bar{\bar{Y}} = \frac{1}{B}\sum_{i=1}^{B} \bar{Y}_{\boldsymbol{\theta}_b^\star}.$$

The bootstrap estimate of bias is then

$$\hat{\beta} = \bar{\bar{Y}} - \bar{Y}_{\boldsymbol{\theta}^{mle}}.$$

All bootstrap estimators work from the idea that the relationship between the true input parameters, $\boldsymbol{\theta}^c$, and the MLEs, $\boldsymbol{\theta}^{mle}$, is approximately equivalent to the relationship between the MLEs, $\boldsymbol{\theta}^{mle}$, and the bootstrap estimates, $\boldsymbol{\theta}^\star$. In the case of bias we are assuming that if we had infinite simulation effort available so that we could effectively observe $\eta(\cdot)$ without any simulation noise, that the difference between the simulation output at $\boldsymbol{\theta}^c$ compared to $\boldsymbol{\theta}^{mle}$ would be approximately equal to the difference between the simulation output at $\boldsymbol{\theta}^{mle}$ compared to $\boldsymbol{\theta}^\star$. That is

$$\mathbb{E}[\eta(\boldsymbol{\theta}^{mle})] - \eta(\boldsymbol{\theta}^c) = \mathbb{E}[\eta(\boldsymbol{\theta}^{mle})|\boldsymbol{\theta}^c] - \eta(\boldsymbol{\theta}^c) \approx \mathbb{E}[\eta(\boldsymbol{\theta}^\star)|\boldsymbol{\theta}^{mle}] - \mathbb{E}[\eta(\boldsymbol{\theta}^{mle})|\boldsymbol{\theta}^c].$$

We now describe the formulation of our method for input bias reduction by input parameter re-calibration. Note that this formulation is not dependent on the type of bias estimate used.

### 3.1 Problem Formulation

We aim to adjust the input parameters of the simulation model, $\boldsymbol{\theta}$, to correct for the input modelling bias, whilst causing minimal deviation from $\boldsymbol{\theta}^{mle}$. Recall that $\boldsymbol{\theta}^{mle}$ are data-driven estimates, and often have statistically advantageous properties as $m$ gets large such as unbiasedness and consistency with respect to $\boldsymbol{\theta}^c$, we therefore would like our bias-corrected parameters to stay as close as possible to $\boldsymbol{\theta}^{mle}$. To take this into account we solve the following optimisation model:

$$\min_{\boldsymbol{\theta}} ||\boldsymbol{\theta} - \boldsymbol{\theta}^{mle}||^2 \tag{3}$$

$$\text{subject to} \qquad \bar{Y}_{\boldsymbol{\theta}} = \bar{Y}_{\boldsymbol{\theta}^{mle}} - \widehat{\beta} = 2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}}. \tag{4}$$

where $\bar{Y}_{\boldsymbol{\theta}^{mle}} - \hat{\beta} = 2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}}$ is the bias-corrected simulation output. The optimal solution of this quadratic program is the set of input parameters that correct for the estimated bias whilst minimising the necessary change to the data-based estimates (MLEs). Whilst the objective function is quadratic, constraint (4) is likely to be a non-linear function. Thus, we propose using a sequential quadratic programming solution approach.

### 3.2 Formulation of the Sequential Quadratic Program

Sequential Quadratic Programming (SQP) uses a sequence of sub-problems to produce a solution by approximating an optimisation problem with a quadratic program. The objective function is approximated locally by a quadratic function about the current iterate, whilst the constraints are linearised. The solution to the sub-problem is used as the next incumbent solution. In solving the problem we set the initial solution to be $\boldsymbol{\theta}^1 = \boldsymbol{\theta}^{mle}$.

To set up the SQP sub-problem, constraint (4) must be linearised. For this, we use a first-order Taylor series expansion of the simulation response. At the $k^{\text{th}}$ iteration, this is

$$m(\boldsymbol{\theta};\boldsymbol{\theta}^k) = m(\boldsymbol{\theta}^k) + \nabla m(\boldsymbol{\theta}^k)^T(\boldsymbol{\theta} - \boldsymbol{\theta}^k). \tag{5}$$

When the simulation response, $\eta(\cdot)$, is unknown, $m(\boldsymbol{\theta}^k) = \bar{Y}_{\boldsymbol{\theta}^k}$, and is calculated by running $n$ replications of the simulation with $\boldsymbol{\theta}^k$. To estimate the gradient term, $\nabla m(\boldsymbol{\theta}^k)$, we use the internal estimator proposed by Wieland and Schmeiser (2006) and used for input uncertainty quantification by Lin et al. (2015). This method for gradient estimation requires no further simulation effort than the replications required to calculate $\bar{Y}_{\boldsymbol{\theta}^k}$. Consider a simulation model with a single stationary Poisson input process, i.e. a system where inter-arrival times can be assumed to be exponentially distributed with some unknown rate $\lambda^c$. Using observations from the real-world system we can approximate $\lambda^c$ using the MLE, $\lambda^{mle}$. Within the simulation random inter-arrival times are generated from $A \sim \text{Exp}(\lambda^{mle})$ and these random observations can then be used to re-estimate the MLE of the input model that generated them. Let us denote this estimate by $\bar{\lambda}$. Note that $\mathbb{E}(\bar{\lambda}) = \lambda^{mle}$ as $\lambda^{mle}$ was used to generate the observations. After $n$ replications of the simulation this process results in $n$ paired observations, $(Y_j, \bar{\lambda}_j)$ for $j = 1, 2, \ldots, n$, which can be assumed to be dependent as the simulation output, $Y$, is likely to depend on the input models that drove it. If in addition we assume the joint distribution of the paired observations to be bivariate normal then

$$\mathbb{E}[Y_j(\lambda^{mle})|\bar{\lambda}_j] = \eta(\lambda^{mle}) + \Sigma_{Y\bar{\lambda}}\Sigma_{\bar{\lambda}\bar{\lambda}}^{-1}(\bar{\lambda}_j - \lambda^{mle}) = \psi_0 + \psi_1\bar{\lambda}_j$$

where $\Sigma_{Y\bar{\lambda}}$ is the covariance between $Y_j$ and $\bar{\lambda}_j$ and $\Sigma_{\bar{\lambda}\bar{\lambda}}$ is the variance of $\bar{\lambda}_j$. Using this model the gradient estimated at $\lambda^{mle}$, is equal to $\psi_1 = \Sigma_{Y\bar{\lambda}}\Sigma^{-1}_{\bar{\lambda}\bar{\lambda}}$, which can easily be estimated using least squares. For simulation models with multiple input distributions this method can be extended using the assumption that the joint distribution of the simulation output and input model parameters is multivariate normal. In this case the gradient, $\nabla m(\boldsymbol{\theta}^k)$, can again be estimated easily using least squares.

The SQP is a sequential algorithm where the solution in step $k$, the step, $\mathbf{p}^k$, is found by solving the following quadratic program:

$$\min_{\mathbf{p}} ||(\boldsymbol{\theta}^k + \mathbf{p}) - \boldsymbol{\theta}^{mle}||^2 \tag{6}$$

$$\text{subject to} \qquad m(\boldsymbol{\theta}^k) + \nabla m(\boldsymbol{\theta}^k)^T \mathbf{p} = 2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}}. \tag{7}$$

**THEOREM 3.1.** *The optimal solution to the SQP is*

$$\mathbf{p}^k = \left( \frac{(2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}} - m(\boldsymbol{\theta}^k)) + \nabla m(\boldsymbol{\theta}^k)^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle})}{||\nabla m(\boldsymbol{\theta}^k)||^2} \right) \nabla m(\boldsymbol{\theta}^k) - (\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle}), \tag{8}$$

*Proof.* Proof of Theorem A.1 The proof that (17) satisfies the first order Karush-Kuhn-Tucker (KKT) conditions is presented in the accompanying online supplement. □

This analytical solution removes the need for a complex optimisation algorithm at each step. On solving (17) the set of input parameters is:

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \mathbf{p}^k. \tag{9}$$

At each step in the algorithm, a new experiment is performed to estimate the simulation output and its gradient at the new position, $\boldsymbol{\theta}^{k+1}$. This process continues iteratively until some stopping criterion is met. We denote the total number of iterations until the stopping rule is satisfied by $K$. One choice of stopping criterion is to continue iterations of the SQP until the simulation output in iterate $k$, $\bar{Y}_{\boldsymbol{\theta}^k}$, falls within an approximate 95% confidence interval of the bias corrected target, $2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}}$. Conditional on the observed data $\bar{Y}_{\boldsymbol{\theta}^{mle}}$ is constant thus the confidence interval is calculated using the output values of the $B$ bootstrap simulation experiments $\{\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_B\}$. Under the assumption that $\bar{\bar{Y}}$ is approximately normally distributed we have

$$(2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}}) \pm t_{B-1,0.975} \frac{s}{\sqrt{B}} \tag{10}$$

where $s^2 = 1/(B-1) \sum_{i=1}^{B} (\bar{Y}_i - \bar{\bar{Y}})$. In using this stopping rule, when the output of the nominal experiment $\bar{Y}_{\boldsymbol{\theta}^{mle}}$ falls within a 95% confidence interval of $2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}}$ no iterations of the SQP will be completed. In other

words, we do not trigger the SQP if we would accept the hypothesis that $\bar{Y}_{\boldsymbol{\theta}^{mle}}$ comes from the approximate distribution of the true outputs i.e. when it is not required.

### 3.3 Feasibility and the input parameter space

The method presented in Section 3.2 assumes that all input parameters $\boldsymbol{\theta}^c$ are unbounded. For certain parametric distributions this is known not to be true and in reality this is necessary to match the process physics of many systems. For example rate parameters, i.e. arrival and service rates cannot be negative as entities cannot enter a system at a negative rate. There is also no general rule of thumb for input parameters, and for a single system there maybe a range of necessary bound constraints.

For many input distributions, adding individual bound constraints to the SQP formulation (7) is sufficient:

$$\min_{\mathbf{p}} ||(\boldsymbol{\theta}^k + \mathbf{p}) - \boldsymbol{\theta}^{mle}||^2 \tag{11}$$

$$\text{subject to} \qquad m(\boldsymbol{\theta}^k) + \nabla m(\boldsymbol{\theta}^k)^T \mathbf{p} = 2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}} \tag{12}$$

$$\theta_i^k + p_i \geq L_i \qquad \forall i \in \mathscr{I}_L \tag{13}$$

$$\theta_i^k + p_i \leq U_i \qquad \forall i \in \mathscr{I}_U. \tag{14}$$

But this addition means the analytical solution (17) may not be feasible. Retaining feasibility is important as it is often impossible to run a simulation to obtain the bias estimate with infeasible parameters (one cannot sample from an exponential distribution with a negative rate). Thus, we provide an alternative step for iterations where the input parameters are suggested by the traditional SQP are infeasible.

It is also important to note that the bound constraints will only become active if an input parameter falls close to the boundary of the feasible space. For example we may have problems in a system where arrivals occur at a very low rate, but for rates far from zero we would not expect any steps of the SQP to pass into an infeasible part of the parameter space.

In practice to deal with this problem we suggest that in all steps of the SQP the analytical step (17) is attempted first. If it is feasible this step is quick to calculate and is known to be optimal. If this step fails i.e. the step takes us outside of the feasible input parameter space we suggest an alternative projected gradient step, see Algorithm 3. In essence, the violation of the bound constraints from step (17) is corrected for by $\mathbf{e}$, giving an intermediate step, $\boldsymbol{\theta}^{k'}$, with all well behaved parameters unchanged and the badly behaved parameters equal to the boundary. We now search along the boundary in the direction of the projected

gradient, $\tilde{\nabla} m(\boldsymbol{\theta}^k)$, for the point at which (7) holds. If this leads to infeasibility in another bound constraint, the process is repeated. This approach is illustrated in our evaluation in Section 4.2 on a Stochastic Activity Network (SAN) example. In most cases where Algorithm 3 was used, only one iteration was needed, though occasionally multiple iterations were required.

**Algorithm 1** Projected gradient approach

1: Calculate the step $\mathbf{p}^k$ from Equation (17)
2: Let projected gradient be $\tilde{\nabla} m(\boldsymbol{\theta}^k) = \nabla m(\boldsymbol{\theta}^k)$
3: **while** $(\boldsymbol{\theta}^k + \mathbf{p}^k)$ has infeasible parameters **do**
4: Set components of projected gradient, $\tilde{\nabla} m(\boldsymbol{\theta}^k)$, corresponding to bound parameters to 0
5: Set the infeasible parameters to the boundary of the feasible region ($L_i$ or $U_i$), using a correction step $\mathbf{e}$, to give new position $\boldsymbol{\theta}^{k'}$
6: Calculate $m(\boldsymbol{\theta}^{k'})$ using Equation (5)
7: Add to the proposed step:

$$\mathbf{p}^k \leftarrow \mathbf{p}^k + \mathbf{e} + \frac{(2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}} - m(\boldsymbol{\theta}^{k'}))}{||\tilde{\nabla} m(\boldsymbol{\theta}^k)||^2} \tilde{\nabla} m(\boldsymbol{\theta}^k) \tag{15}$$

8: **end while**
9: **return** $(\boldsymbol{\theta}^k + \mathbf{p}^k)$ as the next solution.

**THEOREM 3.2.** *The solution to the SQP with bound constraints obtained through Algorithm 3 is optimal.*

*Proof.* Proof of Theorem B.1 A proof that the solution satisfies the first order KKT conditions is presented in the accompanying online supplement. □

The advantage of our suggested approach is that it negates the need to use a complex optimisation algorithm. Another option would be to use a quadratic programming solver with the addition of any necessary bound constraints. This would attain a feasible and optimal solution given the additional constraint, but would be more computationally expensive.

It is important to note that the parametrisation of the input distributions will affect the final result. This is because the objective function, Equation (3), treats all deviations from the MLE equally. That is, a change in a rate parameter of 1, is penalised as much as a change in the scale parameter of 1, despite this leading to different solutions. Further work could consider how to address this dependence on parametrisation whilst retaining the simplicity. Furthermore, for some distributions, more complex constraints are required, such as multinomial distributions. Such distributions are not within the scope of this paper.

### 3.4 Method Overview

The method for biasing the input parameters to reduce bias caused by input modelling is summarised in Algorithm 2. This algorithm demonstrates the process required to implement our method in practise.

**Algorithm 2** Calibrating Simulation Parameters to Reduce Input Bias

1: Calculate the MLE parameters, $\boldsymbol{\theta}^{mle}$ from the available data, set number of simulation replications $n$ and number of bootstrap samples $B$
2: Estimate $\bar{Y}_{\boldsymbol{\theta}^{mle}}$ using $n$ simulation replications with $\boldsymbol{\theta}^{mle}$ as the input
3: Estimate the bias due to input modelling, $\hat{\beta}$, e.g. through bootstrap estimator
4: Calculate tolerance interval with, $\varepsilon$, around $\bar{Y}_{\boldsymbol{\theta}^{mle}} - \hat{\beta}$, e.g. through (10)
5: Set $\boldsymbol{\theta}^1 = \boldsymbol{\theta}^{mle}$ and $k = 1$
6: **while** $|\bar{Y}_{\boldsymbol{\theta}^k} - (\bar{Y}_{\boldsymbol{\theta}^{mle}} - \hat{\beta})| > \varepsilon$ **do**
7: Estimate meta-model gradient, $\nabla m(\boldsymbol{\theta}^k)$ using internal gradient estimator
8: Solve the SQP sub-problem to obtain the next step, $\mathbf{p}^k$, using either (17) or Algorithm 3, and set new input parameters $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \mathbf{p}^k$
9: Estimate $\bar{Y}_{\boldsymbol{\theta}^{k+1}}$ using $n$ simulation replications with $\boldsymbol{\theta}^{k+1}$ as the input
10: $k \leftarrow k+1$
11: **end while**
12: **return** $\boldsymbol{\theta}^k$ as the bias corrected input parameters

In summary we have presented an algorithm for reducing bias caused by input modelling given an estimate of it, that simultaneously re-calibrates the input parameters that drive the simulation so that less bias is passed to the simulation output. Note that because our solution is explicit the computational effort required to use the method grows slowly as the number of parameters increases.

## 4 Evaluation

In this section we seek to evaluate our method. We first illustrate its impact on bias caused by input modelling of the expected number in system for an M/M/1/K queueing model, and then implement it on a Stochastic Activity Network (SAN). For both experiments we use the bootstrap estimator of bias caused by input modelling, introduced in Section 2. Recall that our contribution is in the correction of the output and input parameters for bias caused by input modelling and not in the estimate of this bias.

### 4.1 A Queueing Example

To evaluate the performance of the bias correction method presented in Section 3 we consider a queueing model for which the simulation response as a function of the input distribution parameters is known. This allows us to evaluate a (simulation) noise free estimate of bias caused by input modelling. Our system of

choice is an M/M/1/K queueing model; a single server queueing model with a capacity on the number that can enter the system, K. In the M/M/1/K model arrivals are assumed to follow a stationary Poisson process with arrival rate $\lambda$, and the service times are assumed to follow an exponential distribution with rate $\mu$. There are therefore two parametric input models within this system and $\boldsymbol{\theta} = \{\lambda, \mu\}$. Let our performance measure of interest be the expected number in the system, which has the known form,

$$\eta(\lambda, \mu) = \frac{\lambda}{\mu - \lambda} - \frac{(K+1)\lambda^{K+1}}{\mu^{K+1} - \lambda^{K+1}}. \tag{16}$$

This output is known to be sensitive to small changes in traffic intensity, $\rho$, close to 1. It is therefore a system of interest to us, as for $\rho$ close to 1 it likely to be prone to error caused by input modelling. Recall that bias caused by input modelling occurs when the expected simulation response $\eta(\cdot)$ is a non-linear function of its inputs; for traffic intensities close to 1 Model (16) is highly non-linear.

We complete two experiments using this model. Both experiments are designed to test the assumptions of our bias correction method. The first experiment with the M/M/1/K queueing model investigates the sensitivity of the method to the linearisation of constraint (7) in the SQP; for this experiment Model (16) is used directly. The second experiment considers the relative size of the simulation noise and bias caused by input modelling and how this impacts the method; in this experiment a simulation model of the M/M/1/K model is used to evaluate number in system. In both experiments the cap on queue capacity is set to K=100. In Figure 1 we provide a visualisation of the bias reduction and parameter correction method for this system when a simulation model is used to evaluate the number in system. The contours are the difference between the target, $2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}}$, and the exact system performance given by Model (16). In this case, the algorithm does not stop after two iterations (despite being inside the tolerance region) due to simulation noise in the performance estimation. This discrepancy is shown by the second plot in Figure 1 of the bias estimated by the simulation, $\bar{Y}_{\boldsymbol{\theta}^k} - (2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}})$.

Within the first experiment we evaluate our method using two systems, one uncongested system with traffic intensity $\rho$ =0.6 ($\lambda = 0.6$, $\mu = 1$) and one congested system with traffic intensity $\rho = 0.9$ ($\lambda = 0.9$, $\mu = 1$). For both systems the simulation response is non-linear, but the non-linearity is more pronounced in the higher traffic intensity system. When $\rho = 0.9$ we would not expect the linearisation of Constraint (5) to hold well. This effects the quality of our gradient estimate which guides the direction of the step within the SQP. If the gradient estimate is poor we would expect the algorithm to take longer (more iterations/ larger $I$) to achieve the stopping criterion.

In this experiment we also consider different levels of input modelling bias by using different values

of $m$, the number of observations available from which to estimate $\boldsymbol{\theta}^{mle}$. Although $\boldsymbol{\theta}^{c}$ is known in this controlled experiment we treat it as unknown and assume we have $m$ observations from the arrival and service distributions to calculate $\boldsymbol{\theta}^{mle}$. Different values of $m$ will clearly effect how well $\boldsymbol{\theta}^{mle}$ estimates $\boldsymbol{\theta}^{c}$, and thus the size of bias caused by input modelling. In the first experiment we work with $m = 100$, 500 and 5000. As the amount of input data increases we would expect $\boldsymbol{\theta}^{mle}$ and $\boldsymbol{\theta}^{\star}$ to become closer, we may therefore see fewer cases where the SQP is required as the stopping criterion is already met. An increase in $m$ should also see an improvement in the gradient estimate.

For small $m$ we would expect to see some cases where the estimated traffic intensity is greater than 1, $\rho^{mle} = \lambda/\mu > 1$, in which case, given time, the number in system will increase to full capacity K. In Experiment 1 with traffic intensity, $\rho^{c} = 0.9$ and $m = 100$, we saw $\rho^{mle} > 1$ in 20% of experiments. Note that as $K \to \infty$ the distribution of M/M/1/K converges to the distribution of the $M/M/1$ queueing system which for $\rho > 1$ has infinite expectation, see (Schruben and Kulkarni, 1982).

Within the first experiment we consider six parameter configurations in total. For each configuration $B = 400$ bootstrap samples are collected and Model (16) is used to evaluate the expected number in the system for each set of bootstrap input parameters. For each of the six parameter configurations $G = 500$ macro-replications are completed. In Table 1 we record key measures of our performance including: the proportion of cases in which the SQP is triggered, $\delta$; the average number of iterations completed in the $G =$

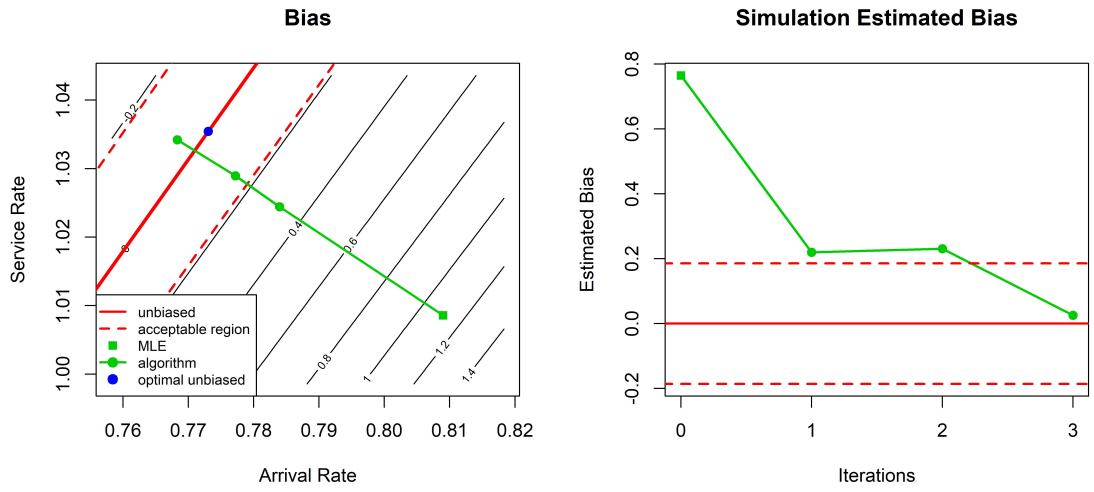


Figure 1: An example application of the method to the M/M/1/K queue, $\rho = 0.8$ and $n = 50$. Left is the trace in the parameter space, right is the simulation estimated bias. Solid red line is the unbiased region, dashed red lines show the tolerance region, the green is the series of inputs from the algorithm and the blue point is the optimal solution of problem (3)-(4).

Table 1: Key measures of performance for testing the M/M/1/K system - considering non-linearity.

| $m$ | $\lambda$ | $\bar{\hat{\beta}}$ | $\delta$ | $\bar{I}$ | $\max(I)$ | $\gamma(\%)$ | $\tau_\lambda$ | $\tau_\mu$ | $\Delta$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | 0.6 | 0.55 | 0.94 | 1.69 | 4 | 90.6 | 0.022 | 0.013 | 0.038 |
| 500 | 0.6 | 0.19 | 0.86 | 1.11 | 3 | 81.9 | 0.022 | 0.013 | 0.038 |
| 5000 | 0.6 | 0.06 | 0.66 | 0.66 | 2 | 63.1 | 0.006 | 0.004 | 0.038 |
| 100 | 0.9 | 15.86 | 0.96 | 2.67 | 5 | 94.5 | 0.030 | 0.027 | 0.225 |
| 500 | 0.9 | 7.32 | 0.95 | 2.34 | 5 | 92.3 | 0.021 | 0.019 | 0.225 |
| 5000 | 0.9 | 1.59 | 0.94 | 1.41 | 4 | 89.1 | 0.008 | 0.007 | 0.225 |

500 macro-replications, $\bar{K}$; the maximum number of iterations observed, $\max(I)$; the average percentage reduction in bias from before to after the SQP in cases where the SQP was triggered, $\gamma$, and the average absolute difference between the MLEs and the final input parameter values, $\tau_\lambda$ and $\tau_\mu$. We also record the average bootstrap estimate of bias $\bar{\hat{\beta}}$ for each scenario, and the value used as the stopping rule of the SQP, $\Delta$. In this experiment the simulation response, Model (16), is known as are the true input parameters, $\lambda^c$ and $\mu^c$, we therefore know our un-biased target exactly, and we set our stopping rule so that the algorithm will terminate in iteration $k$ if the simulation response at $\boldsymbol{\theta}^k$ comes within 5% of $\eta(\boldsymbol{\theta}^c)$.

In Table 1 we see that in all six experiments there is a considerable percentage reduction in the average bias caused by input modelling, $\gamma(\%)$. This reduction is more pronounced in the system with higher traffic intensity $\rho = 0.9$. For both systems we also see that the absolute average difference between the MLEs and the final input parameter values is reasonably small.

As predicted for the higher traffic intensity system the average number of iterations of the SQP is higher. Also predicted was the decrease in the percentage of times the SQP was used as the number of observations of the input distributions, $m$, increased.

Our second experiment was to investigate the effect of simulation noise on our method, to do this we used a simulation model to evaluate the number in the system for the M/M/1/K system. To control the noise we used different numbers of replications of the simulation, $n$. More specifically the number of replications was chosen so that the standard error of the simulation output was approximately 0.01, 0.25 and 2 times the size of bias caused by input modelling. This corresponded to $n = 500$, 50 and 5 replications respectively, and allowed us to comment on the simulation noise relative to the size of bias caused by input modelling. For all three levels of noise we considered an M/M/1/K system with traffic intensity $\rho = 0.8$ ($\lambda = 0.8, \mu = 1$), and worked with $m = 500$ observations from the arrival and service distributions. In this experiment the stopping rule suggested in Equation (10) was used. The results from the three parameter configurations in this experiment are displayed in Table 2.

From Table 2 we see an average percentage reduction of bias caused by input modelling of around 80% in all three configurations. We also see that the the absolute average difference between the MLEs and the final input parameter values is small in all three cases, but appears to be largest in the noisiest scenario, where $n = 5$.

In this second experiment as $n$, the number of replications of the simulation, decreases we see that both $\bar{I}$ and $\max(I)$ increase. This intuitively makes sense as the noise in the simulation impacts the measurement of the stopping criterion and the gradient estimate used to choose the direction of each step. As the noise increases we need more steps of the SQP before our stopping condition is satisfied.

Counter-intuitively perhaps as $n$ decreases (noise increases) we see the proportion of times the SQP is required reduces slightly, this is caused by the stopping rule. In Table 2 as $n$ decreases we see the half-width of the confidence interval used in the stopping rule, $\Delta$, increases. This leads to an increased probability of $\bar{Y}_{mle}$ falling within the interval and triggering the stopping rule.

## 4.2 Stochastic Activity Network

We now implement our method for reducing bias caused by input modelling on a Stochastic Activity Network (SAN). In a SAN arcs represent different activities each with a randomly distributed duration. In Figure 2 adapted from (Avramidis and Wilson, 1996) there are 13 arcs. We shall assume that the duration of each activity $X_i$ for $i = 1, 2, \ldots, 13$ is exponentially distributed with mean $\theta_i > 0$, $X_i \sim \text{Exp}(\theta_i^{-1})$. Mean activity durations are given on the arcs of Figure 2, and all activity durations are assumed to be independent.

SAN networks are useful in the realm of project planning; the arcs in Figure 2 could for example represent different activities that must be completed within a project and the directional arrows could be used to indicate the order in which activities must be completed (i.e. activity a must be completed before c or b etc.). The maximum of the paths through the network can therefore be taken to be the project duration. In Figure 2 node $a$ can be thought of as the starting node or starting activity of the project and node $i$ the end node or end activity of the project.

Table 2: Key measures of performance for testing the M/M/1/K system - considering noise relative to bias caused by input modelling.

| $n$ | $\lambda$ | $\bar{\hat{\beta}}$ | $\delta$ | $\bar{I}$ | $\max(I)$ | $\gamma\,(\%)$ | $\tau_\lambda$ | $\tau_\mu$ | $\Delta$ |
|---|---|---|---|---|---|---|---|---|---|
| 500 | 0.8 | 0.603 | 0.97 | 1.01 | 2 | 83.294 | 0.011 | 0.008 | 0.236 |
| 50 | 0.8 | 0.650 | 0.86 | 1.3 | 4 | 80.516 | 0.013 | 0.010 | 0.244 |
| 5 | 0.8 | 0.561 | 0.85 | 3.97 | 26 | 79.439 | 0.031 | 0.036 | 0.260 |

For this system we considered the expected value of project duration and the probability that the project ends late (using different threshold values to define a late result) as the performance measures of interest. Note that in the field of project planning average project duration is often not the performance measure of interest. The behaviour in terms of bias caused by input modelling seen in each of these experiments was similar, we therefore only report results for the expected project duration here.

To calculate the performance measure of expected total project duration we must observe the duration of all individual activities. It is therefore natural to assume that we would have the same number of observations of each activity duration, $m = m_1 = m_2 = \cdots = m_{13}$. Let us consider three levels of input data $m = 50$, 500 and 5000, and from these observations estimate the MLEs $\theta_i^{mle}$ for $i = 1, 2, \ldots, 13$. Note that parameters $\theta_i$ represent project durations so cannot be negative, i.e. they are bounded below by zero, $\theta_i \geq 0$. In this example some of the mean project durations are low, falling close to the feasible boundary. We may therefore expect to step outside of the feasible parameter space with our usual step (17) in some iterations. This will activate our alternative projected gradient step that takes into account the constraint $\theta_i \geq 0.01$ for an activity $i$.

In simulating the SAN system a single observation is generated from each project duration distribution in each replication. To utilise the internal gradient estimate we therefore batch replications to form the pairs $\{\bar{\boldsymbol{\theta}}_l, \bar{Y}_l\}$ where $\bar{\boldsymbol{\theta}}_l$ is the vector of average project durations for batch $l$ and $\bar{Y}_l$ is the average of the output performance measures from the replications within batch $l$.

In this experiment we completed $G = 1000$ macro replications of the bias reduction procedure. In Table 3 we record the average bootstrap bias estimate $\bar{\hat{\beta}}$, the proportion of cases in with the SQP is triggered, the average, $\bar{I}$, and maximum, $\max(I)$, number of iterations, the average percentage reduction in bias caused by

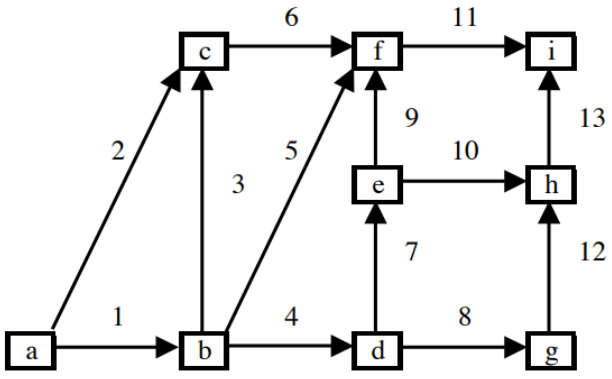


Figure 2: A Stochastic Activity Network

Table 3: Key measures of performance considering the SAN system for $m = 50$, 500 and 5000 observations of each input model over $G = 1000$ macro replications

| $m$ | $\bar{\bar{\beta}}$ | $\delta$ | $\bar{I}$ | $\max(I)$ | $\psi$ (%) | $\gamma$ (%) | $\Delta$ |
|---|---|---|---|---|---|---|---|
| 50 | 1.155 | 0.93 | 3.97 | 24 | 5.7 | 86.3 | 0.270 |
| 500 | 0.084 | 0.87 | 9.91 | 96 | 3.4 | 85.8 | 0.097 |
| 5000 | -0.011 | 0.91 | 14.67 | 106 | 6.2 | 89.1 | 0.064 |

input modelling, $\gamma$, and the half width of the confidence interval around $2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}}$ used in the stopping rule of the SQP, $\Delta$. In addition, for this experiment we also report the percentage of macro replications in which a proposed step fell outside of the feasible parameter space and the alternative projected gradient step was taken, $\psi$.

In Table 3 we see that there is a large percentage reduction in bias caused by input modelling for all three levels of input data. We also see similar behaviour of the average, $\bar{I}$, and maximum, $\max(I)$, number of iterations to the $M/M/1/K$ system experiments with the both metrics increasing as the amount of input data $m$ increases.

Of interest in this example is the percentage of macro-replications in which the alternative, lower bound constrained, step was utilised. We see that for all three levels of input data this percentage is small and with no real pattern across the different values of $m$. It was interesting to note that the average number of iterations of macro-replications hit the boundary was considerably higher than the overall mean (8.12, 23.47 and 32.24 respectively). This suggests that longer runs lead to an increased probability of stepping outside the parameter space.

We performed a similar experiment, using the probability of the project overrunning as the key performance measure, using two different late thresholds (70 and 80 days). These experiments showed very similar results, with a slightly higher $\bar{K}$, $\psi$ and $\gamma$. Interestingly, $\bar{K}$, $\psi$ and $\gamma$ increased further in the higher threshold setting. It is also worth noting that, in one instance the bootstrap estimate of bias lead to a negative $2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}}$ which is not possible and could not be reached through our algorithm. This highlights that one must consider the system properties and the bias estimator before launching into the algorithm. This could be rectified by rerunning the nominal experiment with a higher $n$, or increasing the number of bootstrap samples used to estimate $\hat{\beta}$.

In this section we presented two examples of our method for reducing bias caused by input modelling. In all experiments the percentage reduction of bias caused by input modelling was large. In the M/M/1/K example, where $\eta(\cdot)$ was known, each bias correction calculation took on average less than 0.002 seconds using a 2.1GHz processor running R. Under the same settings, in the longest running SAN example each

bias correction calculation took on average less than 31 seconds, and of these only 0.71 seconds on average was required for the bias correction step.

# 5 The Impact on MSE Caused by Input Modelling

We complete the evaluation by considering the impact of the bias correction on the IU and MSE caused by input modelling for the M/M/1/K system.

A key feature of the method presented in this paper is its ability to re-calibrate the input parameters of the model under investigation so that running the simulation with them would incur reduced bias caused by input modelling. However, by changing the input parameters we also effect the IU passed to the simulation output. In this experiment we consider the impact of this re-calibration on both the input uncertainty and the total MSE caused by input modelling. The MSE caused by input modelling combines both bias caused by input modelling and IU as follows

$$\text{MSE} = \text{IU} + \text{Bias}^2 = \text{Var}[\eta(\boldsymbol{\theta}^{mle})] + \left(\mathbb{E}[\eta(\boldsymbol{\theta}^{mle}) - \eta(\boldsymbol{\theta}^{c})]\right)^2.$$

As in Section 4.1 we consider an uncongested and congested system ($\rho = 0.6$ and 0.9 respectively). Here we again exploit the tractable form of the equation for number in the system, Model (16), to produce noise free estimates of bias and within the calculation of IU within each macro-replication of the experiment. To consider the impact of our bias correction technique we estimate the bias, IU and total MSE before and after the use of the procedure.

To estimate IU in macro-replication $j$ we used a simple bootstrapping approach. Within each bootstrap replication $k$ a set of input data of size $m$ was created. This input data was then sampled with replacement $m$ times, to mimic further data collection, and was then used to re-estimate the MLEs. We denote the bootstrap MLEs by $\{\lambda^\star, \mu^\star\}$. Finally Model (16), was used to evaluate the simulation response and the IU was calculated using

$$\text{IU} = \frac{1}{(B-1)} \sum_{k=1}^{B} (\eta(\lambda_k^\star, \mu_k^\star) - \frac{1}{B} \sum_{k=1}^{B} \eta(\lambda_k^\star, \mu_k^\star))^2$$

see Nelson (2013) noting that the use of Model (16) removes the need to consider simulation noise.

The full experiment ran as follows. In macro replication $j$ of the experiment a data set of size $m = 1000$ was generated using the true parameters, and bias, IU and MSE were calculated from this sample. The bias correction technique was then applied to gain bias-corrected input parameters and bias, IU and MSE caused by input modelling were re-estimated from samples generated using the re-calibrated parameters.

Repeating this process for $G = 1000$ macro replications allowed comparison of the error before and after our bias correction procedure.

Figures 3 and 4 provide box plots of the errors caused by input modelling for an uncongested and congested system respectively. Note that for clarity, the box plots in Figure 4 for the pre-bias corrected system show only the bottom 90% of the MSE values and the corresponding bias and IU estimates that came out of those experiments. When the traffic intensity $\rho$ is close to 1 we would expect to observe some scenarios where the estimated traffic intensity $\rho^{mle}$ surpasses 1 in which case the number in the system has quite different behavior leading to large input bias and thus large MSE. In the congested experiment there were a number of macro-replications where the MSE was very large, the maximum being 3688.1.

Figures 3 and 4 paint a positive picture of the impact of the bias correction technique on the MSE caused by input modelling. There is a clear reduction in MSE and the variability of the MSE outputs across macro-replications. In 98.0% of the macro replications from the uncongested system there was a reduction in the MSE; this was 95.4% in the congested case.

The impact on IU is less clear. For both systems we see fewer outliers after the bias-correction step, and the number of macro-replications where the IU was reduced after bias correction was around 50%. Intuitively it makes sense that a parameter change correcting for bias does not necessarily lead to a reduction in IU. Queueing systems are known to be highly heteroscedastic with increasing traffic intensity, (Cheng and Kleijnen, 1999). Thus when performing the bias correction in a setting where the original MLEs of the input parameters had traffic intensity lower than $\rho^c$ we are likely to be increasing the amount of input uncertainty. In the M/M/1/K scenario, input bias is clearly a larger problem than input uncertainty, and so the total MSE

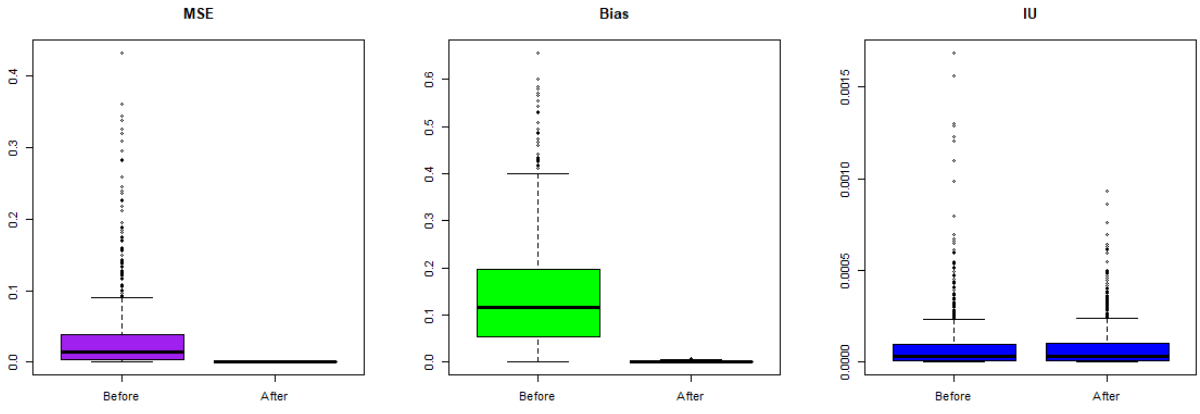


Figure 3: Boxplots of MSE, Bias and IU before and after the bias correction procedure for the M/M/1/K system with traffic intensity $\rho = 0.6$ (uncongested).

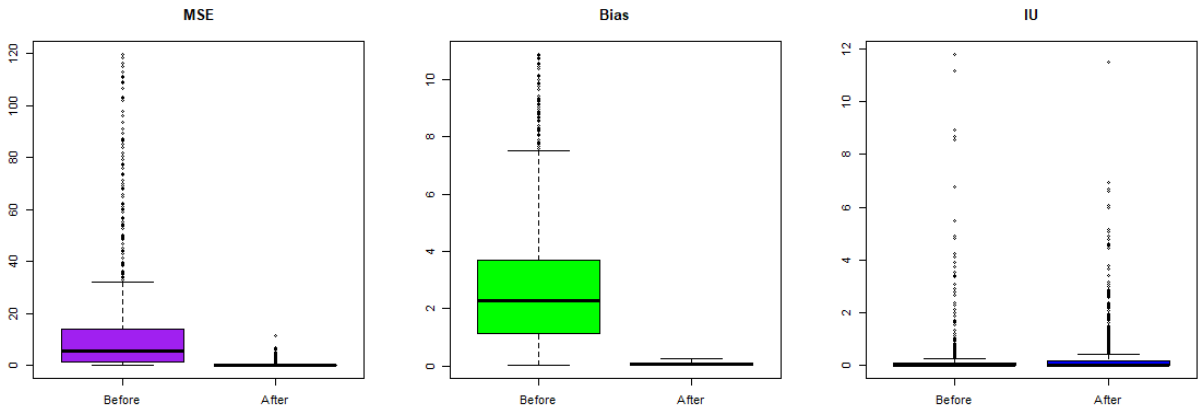


Figure 4: Boxplots of MSE, Bias and IU before and after the bias correction procedure for the M/M/1/K system with traffic intensity $\rho = 0.9$ (congested).

is still reduced by the bias correction procedure, but this is not guaranteed to always be the case. Further investigation into the effect of correcting for input parameter bias is something we leave to further work.

# 6 Conclusion

This paper has presented a novel method to modify the input parameters of a simulation model to correct for bias caused by input modelling. The formulation is solved using a SQP approach, where the quadratic programs can be solved without a numerical solver as long as the parameter space is defined by bound constraints. The simplicity makes this method accessible, with minimal software requirements.

An empirical evaluation of the method has been provided for two example settings. These results show that bias can be reduced significantly, and for large bias, the reduction occurs with relatively few iterations.

Our approach has been shown to successfully reduce bias caused by input modelling but a there are still open questions that remain. For example our current formulation treats all input parameters equally - a deviation from the MLE is penalised identically. However, when we have more data for some of the input distributions than others, we may have more confidence in some input parameters (with tighter confidence intervals on the MLEs). In this case, we may be less willing to make large modifications to these parameters. For multi-parameter distributions, there are also correlations between parameter estimates. Therefore, changing a pair of parameters may be preferable to modifying one alone. Further work could explore how to incorporate these considerations into the framework we have presented. In particular a reformulation of the problem to consider minimising the change to the joint parameter likelihood (as opposed to a change to the parameters themselves) would account for different levels of confidence in the parameters. Moving any of

the inputs away from their MLEs would decrease the likelihood but a change to a parameter we have more confidence in would lead to a larger reduction in the likelihood.

Another consideration of interest is how our method affects the total mean squared error (MSE) caused by input modelling. By modifying the input parameters to correct for bias caused by input modelling the input variance (input uncertainty) passed through the model to the simulation output will also change. Our formulation aims to minimise the change to the model input parameters but we cannot guarantee that we are not moving to an area of the input space where the input parameters propagate a greater amount of input uncertainty and in effect cause greater MSE caused by input modelling. In Section 5 we saw that for the number in system of an M/M/1/K queue the overall MSE cause by input modelling was reduced in the vast majority of cases after re-calibration of the input parameters. The combined study of input uncertainty and bias caused by input modelling is yet to be covered in the literature, but we believe our method to be a step on the road to considering a reduction of both forms of error.

Our final thought on this method is its potential use for calibrating the input parameters of systems configurations that have yet to be observed so that they pass reduced bias caused by input modelling to the output of a simulation. By using meta-model based methods we can calculate the required modification of input parameters over a number of system configurations, with say system parameters $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_H$, and use this information to infer the possible reduction in bias caused by input modelling at some unseen configuration of system parameters $\boldsymbol{x}^\star$. We leave this as a direction for further work.

## Acknowledgements

The authors acknowledge the EPSRC funded EP/L015692/1 STOR-i Centre for Doctoral Training, the financial and computing support from Penn State's Smeal College of Business, the Durham University Business School, and the The High End Computing facility at Lancaster University.

## References

Avramidis, A. N. and Wilson, J. R. (1996). Integrated Variance Reduction Strategies for Simulation. *Operations Research*, 44(2):327–346.

Barton, R. R. (2012). Tutorial: Input Uncertainty in Outout Analysis. In Laroque, C., Himmelspach, J., Pasupathy, R., Rose, O., and Uhrmacher, A., editors, *Proceedings of the 2012 Winter Simulation Conference*, pages 1–12, Piscataway, New Jersey. IEEE.

Cheng, R. C. and Kleijnen, J. P. (1999). Improved Design of Queueing Simulation Experiments with Highly Heteroscedastic Responses. *Operations Research*, 47(5):762–777.

Efron, B. (1982). *The Jackknife, the Bootstrap, and Other Resampling Plans*, volume 38. Siam.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian Calibration of Computer Models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464.

Lam, H. (2016). Advanced Tutorial: Input Uncertainty and Robust Analysis in Stochastic Simulation. In Roeder, T. M. K., Frazier, P. I., Szechtman, R., Zhou, E., Huschka, T., and Chick, S. E., editors, *Proceedings of the 2016 Winter Simulation Conference*, pages 178–192, Piscataway, New Jersey. IEEE.

Lin, Y., Song, E., and Nelson, B. L. (2015). Single-experiment Input Uncertainty. *Journal of Simulation*, 9(3):249–259.

Morgan, L. E., Nelson, B. L., Titman, A. C., and Worthington, D. J. (2019). Detecting Bias Due to Input Modelling in Computer Simulation. *European Journal of Operational Research*, 279(3):869–881.

Nelson, B. (2013). *Foundations and Methods of Stochastic Simulation: A First Course*. Springer Science & Business Media.

Nocedal, J. and Wright, J. (2006a). *Numerical Optimization*. Springer-Verlag, New York, NY, second edition.

Nocedal, J. and Wright, S. J. (2006b). Sequential Quadratic Programming. *Numerical optimization*, pages 529–562.

Reichert, P. and Schuwirth, N. (2012). Linking Statistical Bias Description to Multiobjective Model Calibration. *Water Resources Research*, 48(9).

Schruben, L. and Kulkarni, R. (1982). Some Consequences of Estimating Parameters for the M/M/1 Queue. *Operations Research Letters*, 1(2):75–78.

Song, E., Nelson, B. L., and Pegden, C. D. (2014). Advanced Tutorial: Input Uncertainty Quantification. In Tolk, A., Diallo, S. Y., Ryzhov, I. O., Yilmaz, L., Buckley, S., and Miller, J. A., editors, *Proceedings of the 2014 Winter Simulation Conference*, pages 162–176, Piscataway, New Jersey. IEEE.

Wieland, J. R. and Schmeiser, B. W. (2006). Stochastic Gradient Estimation Using a Single Design Point. In Perrone, L. F., Wieland, F. P., Liu, J., Lawson, B. G., Nicol, D. M., and Fujimoto, R. M., editors, *Proceedings of the 2006 Winter Simulation Conference*, pages 390–397, Piscataway, New Jersey. IEEE.

Withers, C. S. and Nadarajah, S. (2014). Bias Reduction: The Delta Method Versus the Jackknife and the Bootstrap. *Pakistan Journal of Statistics*, 30(1):143–151.

# Appendices

## A Optimality for the Equality Constrained Quadratic Program

In this Appendix, we prove the following result for the SQP sub-problem with no bounds on the parameter space (Section 3.2 of the main article).

**THEOREM A.1.** *The optimal solution to the SQP is*

$$\mathbf{p}^k = \left( \frac{(2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}} - m(\boldsymbol{\theta}^k)) + \nabla m(\boldsymbol{\theta}^k)^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle})}{||\nabla m(\boldsymbol{\theta}^k)||^2} \right) \nabla m(\boldsymbol{\theta}^k) - (\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle}), \tag{17}$$

*Proof.* Proof of Theorem A.1 For ease of notation, we denote the target simulation output, $2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}}$, as $\bar{Y}^{bc}$ and the gradient of the meta-model at $\boldsymbol{\theta}^k$ as $\nabla m_k$ (dropping the explicit dependence on $\boldsymbol{\theta}^k$). We consider the KKT conditions (for example, Nocedal and Wright, 2006a, page 321) for an equality constrained problem:

$$\nabla \mathscr{L}(\mathbf{p}^*, \lambda^*) = \mathbf{0}$$
$$(m(\boldsymbol{\theta}^k) - \bar{Y}^{bc}) + \nabla m_k^T \mathbf{p}^* = 0$$

where $\mathscr{L}(\mathbf{p}, \lambda)$ is the Lagrangian function:

$$\begin{aligned} \mathscr{L}(\mathbf{p}, \lambda) &= ||(\boldsymbol{\theta}^k + \mathbf{p}) - \boldsymbol{\theta}^{mle}||^2 - \lambda((m(\boldsymbol{\theta}^k) - \bar{Y}^{bc}) + \nabla m_k^T \mathbf{p}) \\ &= (\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle})^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle}) + 2(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle})^T\mathbf{p} + \mathbf{p}^T\mathbf{p} - \lambda((m(\boldsymbol{\theta}^k) - \bar{Y}^{bc}) + \nabla m_k^T \mathbf{p}) \\ \Rightarrow \qquad \nabla \mathscr{L}(\mathbf{p}, \lambda) &= 2(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle}) + 2\mathbf{p} - \lambda \nabla m_k. \end{aligned}$$

Consider the first condition:

$$\begin{aligned}\nabla\mathscr{L}(\mathbf{p}^*,\lambda^*) &= 2(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle})+2\mathbf{p}^*-\lambda^*\nabla m_k \\ &= 2(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle})+2\left(\frac{(\bar{Y}^{bc}-m(\boldsymbol{\theta}^k))+\nabla m_k^T(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle})}{||\nabla m_k||^2}\right)\nabla m_k-2(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle})-\lambda^*\nabla m_k \\ &= 2\left(\frac{(\bar{Y}^{bc}-m(\boldsymbol{\theta}^k))+\nabla m_k^T(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle})}{||\nabla m_k||^2}\right)\nabla m_k-\lambda^*\nabla m_k.\end{aligned}$$

Taking

$$\lambda^* = 2\left(\frac{(\bar{Y}^{bc}-m(\boldsymbol{\theta}^k))+\nabla m_k^T(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle})}{||\nabla m_k||^2}\right)$$

gives the required condition. Now consider the second condition:

$$\begin{aligned}&(m(\boldsymbol{\theta}^k)-\bar{Y}^{bc})+\nabla m_k^T\mathbf{p}^* \\ &\quad= (m(\boldsymbol{\theta}^k)-\bar{Y}^{bc})+\nabla m_k^T\left[\left(\frac{(\bar{Y}^{bc}-m(\boldsymbol{\theta}^k))+\nabla m_k^T(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle})}{||\nabla m_k||^2}\right)\nabla m_k-(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle})\right] \\ &\quad= (m(\boldsymbol{\theta}^k)-\bar{Y}^{bc})+\left(\frac{(\bar{Y}^{bc}-m(\boldsymbol{\theta}^k))+\nabla m_k^T(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle})}{||\nabla m_k||^2}\right)\nabla m_k^T\nabla m_k-\nabla m_k^T(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle}) \\ &\quad= (m(\boldsymbol{\theta}^k)-\bar{Y}^{bc})+\bar{Y}^{bc}-m(\boldsymbol{\theta}^k)+\nabla m_k^T(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle})-\nabla m_k^T(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle}) \\ &\quad= 0.\end{aligned}$$

So $\mathbf{p}^*$ is feasible and satisfies the KKT conditions. Thus the step given by (17) is optimal for the SQP sub-problem. □

# B Optimality for the Equality and Bound Constrained Quadratic Program

In this Appendix, we consider the SQP sub-problem in which bound constraints exist for some or all of the input parameters (see Section 3.3 of the main article). The algorithm to generate the solution is given in Algorithm 3.

**THEOREM B.1.** *The solution to the SQP with bound constraints obtained through Algorithm 3 is optimal.*

*Proof.* Proof of Theorem B.1 The step generated by Algorithm 3 is the sum of the original step from Equation (17), plus a series of steps projecting onto the feasible region and moving in the projected gradient

**Algorithm 3** Projected gradient approach

1: Calculate the step $\mathbf{p}^k$ from Equation (17)
2: Let projected gradient be $\tilde{\nabla} m(\boldsymbol{\theta}^k) = \nabla m(\boldsymbol{\theta}^k)$
3: **while** $(\boldsymbol{\theta}^k + \mathbf{p}^k)$ has infeasible parameters **do**
4: Set components of projected gradient, $\tilde{\nabla} m(\boldsymbol{\theta}^k)$, corresponding to bound parameters to 0
5: Set the infeasible parameters to the boundary of the feasible region ($L_i$ or $U_i$), using a correction step $\mathbf{e}$, to give new position $\boldsymbol{\theta}^{k'}$
6: Calculate $m(\boldsymbol{\theta}^{k'}) = m(\boldsymbol{\theta}^k) + \nabla m(\boldsymbol{\theta}^k)^T(\boldsymbol{\theta}^{k'} - \boldsymbol{\theta}^k)$
7: Add to the proposed step:

$$\mathbf{p}^k \leftarrow \mathbf{p}^k + \mathbf{e} + \frac{(2\bar{Y}_{\boldsymbol{\theta}^{mle}} - \bar{\bar{Y}} - m(\boldsymbol{\theta}^{k'}))}{||\tilde{\nabla} m(\boldsymbol{\theta}^k)||^2} \tilde{\nabla} m(\boldsymbol{\theta}^k) \tag{18}$$

8: **end while**
9: **return** $(\boldsymbol{\theta}^k + \mathbf{p}^k)$ as the next solution.

direction (the gradient with all components on their respective bounds set to zero). The sequence of steps produces a sequence of solutions $\boldsymbol{\theta}^{(j)}$. We define $\boldsymbol{\theta}^{(0)}$ to be the solution obtained through Equation (17).

Suppose that a sequence of $J > 0$ iterations of Algorithm 3 are required, each being the projection onto the feasible region followed by a step along the projected gradient direction. For iteration $j$, denote the projected gradient of the linear model as $\tilde{\nabla} m_k^{(j)}$ and the projections onto the feasible region as $\mathbf{e}^{(j)}$. Then the final step made is given by

$$\mathbf{p}^* = \frac{\bar{Y}^{bc} - m(\boldsymbol{\theta}^k) + \nabla m_k^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle})}{||\nabla m_k||^2} \nabla m_k - (\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle}) + \sum_{j=1}^{J} \left[ \mathbf{e}^{(j)} + \frac{\bar{Y}^{bc} - m(\boldsymbol{\theta}_*^{(j)})}{||\tilde{\nabla} m_k^{(j)}||^2} \tilde{\nabla} m_k^{(j)} \right], \tag{19}$$

where the first two terms coming from Equation (17) and $\boldsymbol{\theta}_*^{(j)} = \boldsymbol{\theta}^{(j-1)} + \mathbf{e}^{(j)}$ is the projection of $\boldsymbol{\theta}^{(j-1)}$ onto the feasible parameter space.

The inclusion of bound constraints change the SQP sub-problem to:

$$\min_{\mathbf{p}} ||(\boldsymbol{\theta}^k + \mathbf{p}) - \boldsymbol{\theta}^{mle}||^2 \tag{20}$$

$$\text{subject to} \qquad m(\boldsymbol{\theta}^k) + \nabla m_k^T \mathbf{p} = \bar{Y}^{bc} \tag{21}$$

$$\theta_i^k + p_i \geq L_i \qquad \forall i \in \mathscr{I}_L \tag{22}$$

$$\theta_i^k + p_i \leq U_i \qquad \forall i \in \mathscr{I}_U. \tag{23}$$

The construction of $\mathbf{p}^*$ using the correction terms $\mathbf{e}^{(j)}$ means that it satisfies the inequality constraints.

Now consider the bias constraint:

$$
\begin{aligned}
&m(\boldsymbol{\theta}^k) + \nabla m_k^T \mathbf{p}^* \\
&= m(\boldsymbol{\theta}^k) + \frac{\bar{Y}^{bc} - m(\boldsymbol{\theta}^k) + \nabla m_k^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle})}{||\nabla m_k||^2} \nabla m_k^T \nabla m_k - \nabla m_k^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle}) + \nabla m_k^T \sum_{j=1}^{J} \left[ \mathbf{e}^{(j)} + \frac{\bar{Y}^{bc} - m(\boldsymbol{\theta}_*^{(j)})}{||\tilde{\nabla} m_k^{(j)}||^2} \tilde{\nabla} m_k^{(j)} \right] \\
&= m(\boldsymbol{\theta}^k) + \bar{Y}^{bc} - m(\boldsymbol{\theta}^k) + \nabla m_k^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle}) - \nabla m_k^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle}) + \sum_{j=1}^{J} \left[ \nabla m_k^T \mathbf{e}^{(j)} + \frac{\bar{Y}^{bc} - m(\boldsymbol{\theta}_*^{(j)})}{||\tilde{\nabla} m_k^{(j)}||^2} \nabla m_k^T \tilde{\nabla} m_k^{(j)} \right] \\
&= \bar{Y}^{bc} + \sum_{j=1}^{J} \left[ \nabla m_k^T \mathbf{e}^{(j)} + \bar{Y}^{bc} - m(\boldsymbol{\theta}_*^{(j)}) \right].
\end{aligned}
$$

The last equality uses the fact that $\nabla m_k^T \tilde{\nabla} m_k^{(j)} = ||\tilde{\nabla} m_k^{(j)}||^2$ because $\tilde{\nabla} m_k^{(j)}$ only differs from $\nabla m_k$ in the components that are zero in $\tilde{\nabla} m_k^{(j)}$. Now note that $\boldsymbol{\theta}_*^{(j)}$ is the result of moving $\mathbf{e}^{(j)}$ from the "unbiased" but infeasible $\boldsymbol{\theta}^{(j-1)}$ (this is because the solution from the previous iteration must satisfy (21)). Therefore:

$$
m(\boldsymbol{\theta}_*^{(j)}) = m(\boldsymbol{\theta}^{(j-1)}) + \nabla m_k^T \mathbf{e}^{(j)} = \bar{Y}^{bc} + \nabla m_k^T \mathbf{e}^{(j)}. \tag{24}
$$

Substituting this in gives:

$$
m(\boldsymbol{\theta}^k) + \nabla m_k^T \mathbf{p}^* = \bar{Y}^{bc} + \sum_{j=1}^{J} \left[ \nabla m_k^T \mathbf{e}^{(j)} + \bar{Y}^{bc} - \bar{Y}^{bc} - \nabla m_k^T \mathbf{e}^{(j)} \right] = \bar{Y}^{bc},
$$

Thus the new step is feasible.

Including the bound constraints, the Lagrangian becomes:

$$
\mathscr{L}(\mathbf{p}, \lambda, \boldsymbol{\mu}^L, \boldsymbol{\mu}^U) = ||(\boldsymbol{\theta}^k + \mathbf{p}) - \boldsymbol{\theta}^{mle}||^2 - \lambda \left( m(\boldsymbol{\theta}^k) + \nabla m_k^T \mathbf{p} - \bar{Y}^{bc} \right) - \sum_{i \in \mathscr{I}_L} \mu_i^L (\theta_i^k + p_i - L_i) - \sum_{i \in \mathscr{I}_U} \mu_i^U (U_i - \theta_i^k - p_i)
$$

with gradient

$$
\nabla \mathscr{L}(\mathbf{p}, \lambda, \boldsymbol{\mu}^L, \boldsymbol{\mu}^U) = 2(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle}) + 2\mathbf{p} - \lambda \nabla m_k - \boldsymbol{\mu}^L + \boldsymbol{\mu}^U. \tag{25}
$$

At the proposed step, Equation (19):

$$\begin{aligned}
&\nabla \mathscr{L}(\mathbf{p}^*, \lambda, \boldsymbol{\mu}^L, \boldsymbol{\mu}^U) \\
&= 2(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle}) + 2\left(\frac{\bar{Y}^{bc} - m(\boldsymbol{\theta}^k) + \nabla m_k^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle})}{||\nabla m_k||^2}\right)\nabla m_k - 2(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle}) + 2\sum_{j=1}^{J}\left[\mathbf{e}^{(j)} + \frac{\bar{Y}^{bc} - m(\boldsymbol{\theta}_*^{(j)})}{||\tilde{\nabla} m_k^{(j)}||^2}\tilde{\nabla} m_k^{(j)}\right] \\
&\qquad - \lambda \nabla m_k - \boldsymbol{\mu}^L + \boldsymbol{\mu}^U \\
&= 2\left(\frac{\bar{Y}^{bc} - m(\boldsymbol{\theta}^k) + \nabla m_k^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle})}{||\nabla m_k||^2}\right)\nabla m_k + 2\sum_{j=1}^{J}\left[\mathbf{e}^{(j)} - \frac{\nabla m_k^T \mathbf{e}^{(j)}}{||\tilde{\nabla} m_k^{(j)}||^2}\tilde{\nabla} m_k^{(j)}\right] - \lambda \nabla m_k - \boldsymbol{\mu}^L + \boldsymbol{\mu}^U. \quad (26)
\end{aligned}$$

where the final equality uses Equation (24).

Consider the quantity $\nabla m_k^T \mathbf{e}^{(j)}$. Note that $e_i^{(j)}$ is either 0 or in the opposite direction of $p_i^*$, as otherwise there would be no need to correct. If $m(\boldsymbol{\theta}^k)$ is less than $\bar{Y}^{bc}$, then $p_i^*$ will be in the same direction as $(\nabla m_k)_i$, as the search seeks to increase the output. Thus, $(\nabla m_k)_i e_i^{(j)} \leq 0$ for all components, and so $\nabla m_k^T \mathbf{e}^{(j)} \leq 0$. Similarly, if $m(\boldsymbol{\theta}^k)$ is greater than $\bar{Y}^{bc}$, $(\nabla m_k)_i e_i^{(j)} \geq 0$, so $\nabla m_k^T \mathbf{e}^{(j)} \geq 0$.

Let us consider Equation (26) componentwise, based on whether the component is on a lower or upper boundary, or unconstrained at the point $\boldsymbol{\theta}^k + \mathbf{p}^*$. First consider an unconstrained component, $i$. In this case, $e_i^{(j)} = 0$ for all $j$ and the component of the projected gradient remains unchanged, i.e. $(\tilde{\nabla} m_k^{(j)})_i = (\nabla m_k)_i$ for all $j$. Furthermore, through the complementary slackness criterion, $\mu_i^L = \mu_i^U = 0$. Therefore:

$$\begin{aligned}
\nabla \mathscr{L}(\mathbf{p}^*, \lambda, \boldsymbol{\mu}^L, \boldsymbol{\mu}^U)_i &= 2\left(\frac{\bar{Y}^{bc} - m(\boldsymbol{\theta}^k) + \nabla m_k^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle})}{||\nabla m_k||^2}\right)(\nabla m_k)_i - 2\sum_{j=1}^{J}\left[\frac{\nabla m_k^T \mathbf{e}^{(j)}}{||\tilde{\nabla} m_k^{(j)}||^2}\right](\nabla m_k)_i - \lambda(\nabla m_k)_i \\
&= \left\{2\left(\frac{\bar{Y}^{bc} - m(\boldsymbol{\theta}^k) + \nabla m_k^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle})}{||\nabla m_k||^2}\right) - 2\sum_{j=1}^{J}\left[\frac{\nabla m_k^T \mathbf{e}^{(j)}}{||\tilde{\nabla} m_k^{(j)}||^2}\right] - \lambda\right\}(\nabla m_k)_i.
\end{aligned}$$

Thus taking

$$\lambda^* = 2\left(\frac{\bar{Y}^{bc} - m(\boldsymbol{\theta}^k) + \nabla m_k^T(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{mle})}{||\nabla m_k||^2}\right) - 2\sum_{j=1}^{J}\left[\frac{\nabla m_k^T \mathbf{e}^{(j)}}{||\tilde{\nabla} m_k^{(j)}||^2}\right] \quad (27)$$

leaves all free components satisfying $\nabla \mathscr{L}(\mathbf{p}^*, \lambda^*, \boldsymbol{\mu}^L, \boldsymbol{\mu}^U)_i = 0$.

Now consider a component, $i$, of $\boldsymbol{\theta}^k + \mathbf{p}^*$ that sits on its lower bound. Then $p_i^* \leq 0$ and $\mu_i^U = 0$ by complementary slackness. Suppose that it was free until iteration $J_i$. Then $e_i^{(J_i)} > 0$ and $e_i^{(j)} = 0$ for $j \neq J_i$ and:

$$(\tilde{\nabla} m_k^{(j)})_i = \begin{cases} (\nabla m_k)_i & j < J_i \\ 0 & j \geq J_i. \end{cases}$$

Therefore, component $i$ of $\nabla\mathscr{L}(\mathbf{p}^*,\lambda,\boldsymbol{\mu}^L,\boldsymbol{\mu}^U)$ can be written as:

$$\begin{aligned}
&\nabla\mathscr{L}(\mathbf{p}^*,\lambda^*,\boldsymbol{\mu}^L,\boldsymbol{\mu}^U)_i \\
&\quad= 2\left(\frac{\bar{Y}^{bc}-m(\boldsymbol{\theta}^k)+\nabla m_k^T(\boldsymbol{\theta}^k-\boldsymbol{\theta}^{mle})}{||\nabla m_k||^2}\right)(\nabla m_k)_i + 2e_i^{(J_i)} - 2\sum_{j=1}^{J_i-1}\left[\frac{\nabla m_k^T\mathbf{e}^{(j)}}{||\tilde{\nabla} m_k^{(j)}||^2}\right](\nabla m_k)_i - \lambda^*(\nabla m_k)_i - \mu_i^L \\
&\quad= 2e_i^{(J_i)} + 2\sum_{j=J_i}^{J}\left[\frac{\nabla m_k^T\mathbf{e}^{(j)}}{||\tilde{\nabla} m_k^{(j)}||^2}\right](\nabla m_k)_i - \mu_i^L
\end{aligned}$$

where the second equality comes from substituting Equation (27) for $\lambda^*$. Suppose that $m(\boldsymbol{\theta}^k)$ is less than $\bar{Y}^{bc}$. Then, as $p_i^* \leq 0$, $(\nabla m_k)_i$ must also be negative and $\nabla m_k^T\mathbf{e}^{(j)} \leq 0$ (as shown above). Similarly, if $m(\boldsymbol{\theta}^k)$ is greater than $\bar{Y}^{bc}$, then, as $p_i^* \leq 0$, $(\nabla m_k)_i$ must be positive and $\nabla m_k^T\mathbf{e}^{(j)} \geq 0$ (as shown above). Therefore, in either case $\nabla m_k^T\mathbf{e}^{(j)} \times (\nabla m_k)_i \geq 0$ for all $j$. So choosing

$$\mu_i^{L*} = 2e_i^{(J_i)} + 2\sum_{j=J_i}^{J}\left[\frac{\nabla m_k^T\mathbf{e}^{(j)}}{||\tilde{\nabla} m_k^{(j)}||^2}\right](\nabla m_k)_i \geq 0 \tag{28}$$

satisfies the KKT condition that the Lagrangian multiplier for an inequality must be non-negative and that $\nabla\mathscr{L}(\mathbf{p}^*,\lambda^*,\boldsymbol{\mu}^{L*},\boldsymbol{\mu}^U)_i = 0$ for all components lying on a lower bound.

If component $i$ lies on the upper bound, $\mu_i^L = 0$, $p_i^* \geq 0$ and $e_i^{(J_i)} < 0$. Thus, the derivative of the Lagrangian is given by:

$$\nabla\mathscr{L}(\mathbf{p}^*,\lambda^*,\boldsymbol{\mu}^{L*},\boldsymbol{\mu}^U)_i = 2e_i^{(J_i)} + 2\sum_{j=J_i}^{J}\left[\frac{\nabla m_k^T\mathbf{e}^{(j)}}{||\tilde{\nabla} m_k^{(j)}||^2}\right](\nabla m_k)_i + \mu_i^U.$$

Using a similar argument to the lower bound case (with signs exchanged) we can show that

$$\mu_i^{U*} = -2e_i^{(J_i)} + 2\sum_{j=J_i}^{J}\left[\frac{\nabla m_k^T\mathbf{e}^{(j)}}{||\tilde{\nabla} m_k^{(j)}||^2}\right](\nabla m_k)_i \geq 0. \tag{29}$$

Therefore, using Equations (19), (27), (28) and (29), we have that:

$$\nabla\mathscr{L}(\mathbf{p}^*,\lambda^*,\boldsymbol{\mu}^{L*},\boldsymbol{\mu}^{U*}) = \mathbf{0} \tag{30}$$

$\mathbf{p}^*$ is feasible and $\boldsymbol{\mu}^{L*},\boldsymbol{\mu}^{U*} \geq 0$. This completes the proof that the step $\mathbf{p}^*$ is the optimal solution to the SQP sub-problem as it satisfies the KKT conditions. □