MODEL PREDICTIVE CONTROL OF AN ONLINE ADVERTISEMENT CAMPAIGN IN A REAL-TIME-BIDDING AUCTION

Bjarke Arnskjær Hastrup and Michael Wamberg
{s151602, s146791}@student.dtu.dk

Technical University of Denmark - DTU

ABSTRACT

In this paper we apply Model Predictive Control (MPC) to the problem of controlling the pace of an online advertising campaign. When a user enters a web-page, an auction is immediately initiated, inviting advertisers to submit a bid in a sealed second-price auction (in advertising called *real-time-bidding* or RTB) in which the winner is awarded with the impression, i.e. the winner's ad (a display banner) is showed to the user who may, or may not, click on the ad. With the advertiser bidding for ad space across various web-pages with ad slots varying in size, visibility and competitiveness, the advertiser is faced with a high-dimensional control problem of choosing a bidding strategy that maximizes the expected number of clicks while staying within budget.
To solve this problem, we use Mean-Variance Economic MPC to maximize the expected number of clicks and reduce the risk of not delivering exactly on budget. The dynamical and competitive auction system is adaptively estimated using Bayesian regression techniques where a Markov Chain Monte Carlo (MCMC) sampler is employed to estimate the expected cost (or spend) as a function of bid price. A randomized bidding strategy (Heisenberg bidding) is then used in order to smoothen the plant gain, i.e. to eliminate dramatic changes in cost for small changes in bid prices, allowing for a more robust control system.
The Mean-Variance Economic MPC shows promising results in a synthetic RTB auction system, in which it manages to deliver a smooth spending over the life time of a campaign while maximizing the number of clicks.

***Index Terms*—** Mean-Variance Economic Model Predictive Control - Real-time-bidding - Heisenberg bidding - Budget pacing - Bayesian hierarchical modelling

1. INTRODUCTION

In recent years, countless industries have experienced a transition to e-commerce, which has had a natural impact on the growth of online advertising. In 2019 the online advertising market was valued at USD 304.0 billion [1] thus surpassing traditional advertising channels like television, radio and newspapers [2] and it is anticipated that the market will continue its dramatic growth in the coming years.
At the core of online advertising are the online auctions which take place on so-called ad exchanges where publishers sell their advertising spaces (or slots) to advertisers. The growth in the industry has led to a more competitive environment on these exchanges, why the production of optimization algorithms which helps the advertiser in their bidding is more important than ever. The purpose of the optimization algorithms is to optimize an objective function defined on the basis of the advertiser's goals, which are typically oriented towards achieving as many clicks as possible within their budget. The advertisers constraints along with the dynamics of the market and the uncertainties involved make this an ideal optimal control problem.
Feedback based control methods have previously been implemented in an online advertising setting [3, 4, 5, 6]. Here the feedback controller monitors various KPI's like click-through-rates (CTR) and cost-per-click (CPC) in order to produce an error value, which is the deviation from the observed value to a reference value. Based on this error value, the feedback controller produces its output signal, which is used in order to adjust the advertiser's bidding.
To ensure that the advertisers meet their predefined KPI values [6] proposes a proportional–integral–derivative controller (PID controller) to perform the bid adjustments. The PID controller applies the bid adjustment based on a linear combination of proportional, integral, and derivative terms, which are computed based on the current and past states of the system. In this paper, we develop a model predictive controller to handle the control problem. As the name indicates the control law of the model predictive controller builds upon a predictive model, which attempts to predict the behavior of the system. The model is typically formulated from the impulse response, step response, transfer function or state space model [7], [8] while in this project we construct our control objective based on a cost and click model using Bayesian inference. Since the objective in this problem is economically oriented, we use a mean-variance economic MPC formulation of the control objective [9]. CVXPY [10], a Python-embedded modelling language for convex optimization is used upon this objective

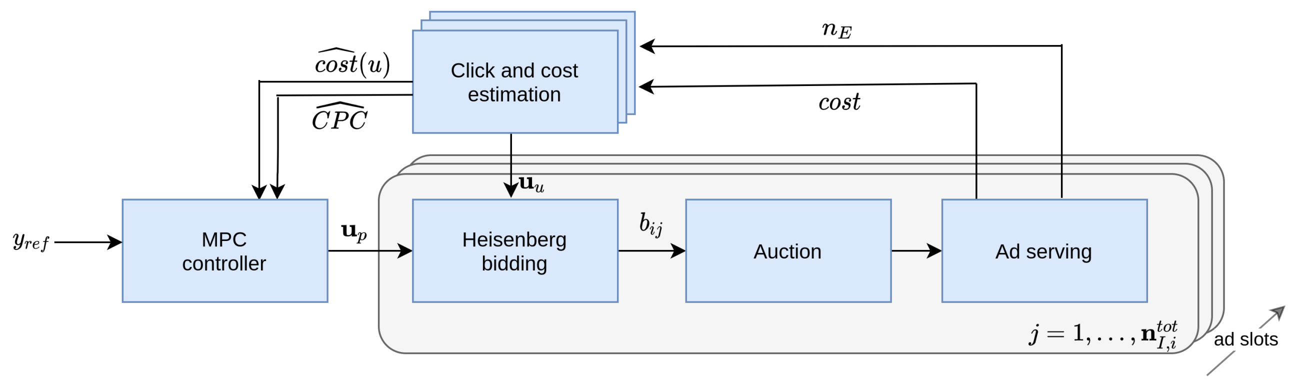

Fig. 1. The MPC block diagram illustrates how the controller, given a desired reference cost trajectory, chooses a vector of nominal bids (future bids are discarded and recalculated at the following time step) that together with the bid uncertainties are perturbed using Heisenberg bidding (section 3) thereby outputting a realized bid price for every auction j, in every ad slot i. If we win the auction, the ad is shown to the end user who or may not click on the ad. Each day click and cost data is collected for every ad slot. This data is used to re-estimate the CPCs and linearized cost functions that are fed back to the controller the following day.

in order to determine the most optimal future trajectory for the control signal. In this optimization process we also define the constraints of the problem and then go through the process repeatedly for each sample, i.e. only the control signal that is determined for the next time step is implemented, while the predictions always sees the same number of time steps into the future. Thus, we optimize in a receding time horizon.
With inspiration from [11], the robustness of our control is ensured by means of the randomized bidding strategy *Heisenberg bidding*.

The remaining part of this paper is organized as follows. Section 2 presents our optimization objectives. The Heisenberg bidding strategy is elaborated in section 3, while section 4 provides a detailed explanation of our established mean-variance economic MPC problem. The components of our model are presented in section 5, the cost model in section 5.1 and the click model in section 5.2. Subsections 5.1.2 and 5.2.2 show how the variance and mean objectives are set up. Section 6 reviews how the synthetic auction system in which we operate is constructed. The simulation results are presented in section 7, and a conclusion is provided in section 8.

2. BUDGET PACING AND CLICK MAXIMIZATION OF AN ONLINE ADVERTISEMENT CAMPAIGN

In this paper, we take on the role of an advertiser who bids for ad space in online (display) ad auctions, with the aim of winning ad impressions which allow us to show our banner ad to internet users. Perhaps more appropriately, we should imagine our role as that of a media agency which performs such ad bidding, on behalf of not only one advertiser, but a large set of advertisement clients, all with the goal of increasing the amount of traffic on their respective web-pages. This necessarily calls for a highly automated setup where ad bidding is performed without human intervention. Various KPIs can be used to measure the success of an individual advertisement campaign and even though the number of conversions is ultimately what creates value to the advertiser, here we will be using the number of ad clicks as the "event of interest", rather than conversions, acquisitions or other ROI (Return-On-Investment) related KPIs, since these are not as easily accessible and are often associated with long delays.

As mentioned, we first need to win impressions by bidding in real-time auctions in order to generate clicks on our client's ads. It is most common in real-time-bidding (RTB) that every auction is a sealed second price auction, meaning that the winner only pays the price of the second highest bidder. Furthermore, we will consider each ad auction to be attributed to a specific *ad slot* which specifies the particular web-page, banner size and so on. It is obvious that some ad slots are associated with higher CTR's which is likely also reflected in the valuation by competitive bidders. Let the number of daily auctions (ad opportunities) in ad slot i be given by $n_{I,i}^{tot}(k)$ (where the I is for *impression*). Then the daily number of awarded impressions is

$$n_I(k) = \sum_{i=1}^{n_{slots}} \sum_{j=1}^{n_{I,i}^{tot}(k)} \mathbb{1}_{\{b_{ij}(k) \geq b_{ij}^*(k)\}}, \tag{1}$$

and due to the nature of second price auctions, the associated daily cost given by

$$c(k) = \sum_{i=1}^{n_{slots}} \sum_{j=1}^{n_{I,i}^{tot}(k)} b_{ij}^*(k) \mathbb{1}_{\{b_{ij}(k) \geq b_{ij}^*(k)\}}, \tag{2}$$

while the expected number of clicks is calculated from the CTR and the impression count, i.e.

$$\mathbb{E}[n_E(k)] = \sum_{i=1}^{n_{slots}} ctr_i(k) \cdot n_{I,i}(k). \tag{3}$$

To further formalize the campaign optimization problem, we are given a budget B and a time horizon T, and it is our task to choose daily bid prices $b_{ij}(k)$ that maximize the number of clicks n_E (E for "event") while spending the budget smoothly throughout the campaign lifetime. This is known as *budget pacing* and is a desirable objective since the occasional rapid spending associated with non-smooth pacing reveals that we are likely paying overprice for impressions. Although somewhat counter intuitive, an equally important reason to spend the budget smoothly, is to guarantee that the entire budget has been spend at time T. No more, no less. Not spending the entire budget is referred to as *under delivery* and it leads to refunds and thereby loss of commission, but also impractical payment handling that hinders a lean a scalable operation. In order to accommodate both the click maximization objective and the budget pacing objective, we employ a mean-variance economic MPC formulation which contains a linear click-maximization term and quadratic term that penalizes deviations from a steady cost reference trajectory. As described above, we wish to choose daily bid prices $b_{ij}(k)$, with i denoting the ad slot and j denoting the auction, such that we maximize clicks while spending the budget as smoothly as possible. However, we do not intend to build a MPC controller which operates at the individual auction level. Rather, we will partially implement the randomized Heisenberg bidding (to be explained in section 3) in which our controller is to decide on a daily nominal bid price for ad slot i, u_{p_i}, where p is short for *price*. From this nominal bid price we perform a random perturbation, using some relative uncertainty measure u_{ui} that reflects our knowledge of ad slot i. As we shall see, this allows for automatic exploration while potentially smoothing the cost response (plant gain). A diagram of the entire control system is shown in Figure 1.

3. BID GENERATION BY HEISENBERG BIDDING

The Heisenberg bidding strategy works by randomly perturbing a nominal bid price u_p according to a specified probability distribution. The strategy is introduced as part of managing the discontinuous plant gain which can arise from drastic changes in costs when the bid is only slightly changed. A discontinuous plant gain is undesirable as it can cause the controller working on the plant to become unstable and therefore a stable solution cannot be found.

An extra good that comes with the introduction of Heisenberg bidding is that it ensures a continuing exploration of ad opportunities. The exploration is carried out since the Heisenberg bidding makes sure that even low-rated impressions can be won as the bids on these will not go to zero. Hence the name *Heisenberg* bidding since the setting resembles the one known from quantum physics where a low energy particle has a non-zero probability of escaping a deep quantum well.

Now following [11] the perturbation of the nominal bid price is done according to a gamma distribution defined by u_p and a bid uncertainty u_u[1]. Thus, a final bid price b_j is generated by

$$b_j \sim \text{Gamma}\left(\frac{1}{u_u^2}, \frac{1}{u_p u_u^2}\right), \quad j = 1, ..., n_I^{tot}, \tag{4}$$

with $\alpha = 1/(u_u^2)$ and $\beta = 1/(u_p u_u^2)$ defining respectively the shape and inverse scale parameter of the gamma distribution. From the properties of the gamma distribution it follows that

$$\mathbb{E}[b_j] = \frac{\alpha}{\beta} = u_p, \tag{5}$$

$$\mathbb{V}ar[b_j] = \frac{\alpha}{\beta^2} = u_p^2 u_u^2. \tag{6}$$

In Figure 2 is visualized examples of the gamma probability density functions parameterized by the bid price u_p and the bid uncertainty u_u, from which it can be seen how the increased uncertainty spreads out the distribution.

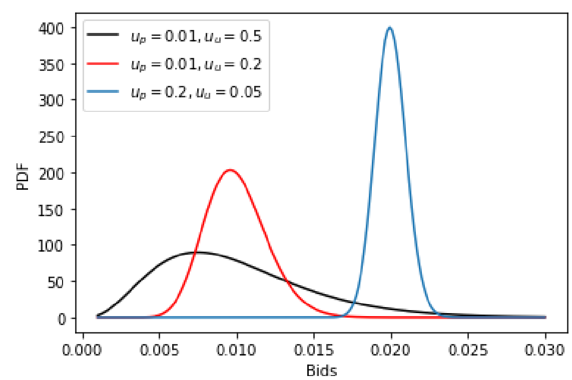

Fig. 2. Gamma probability density function parameterized by the bid price u_p and the bid uncertainty u_u.

Since auction j is won when b_j is greater than or equal to b_j^*, the probability of winning an impression can be defined by

$$\Pr\left(b_j \geq b_j^*\right) = \int_{b_j^*}^{\infty} f(b_j|\alpha, \beta) db_j, \tag{7}$$

with $f(\cdot)$ defining the gamma probability density function, i.e.

$$f(b_j|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} b_j^{\alpha-1} \exp(-\beta b_j), \tag{8}$$

[1]Notice we are omitting subscripts i to simplify the notation. Naturally, we specify Heisenberg distributions for each ad slot individually.

where $\Gamma(\alpha)$ is the gamma function defined by $\Gamma(\alpha) = \int_0^\infty \exp(-x)x^{\alpha-1}dx$. In this paper, we assume we have too little information regarding the distribution of the highest competitive bid prices b_j^* such that the analytical expression of equation (7) therefore cannot be used directly. Instead we will revert to empirical linearization, as shown in section 5.1 in order to estimate the winning probability conditional on a nominal bid (actually it is the conditional cost that is modelled and not the explicit winning probability).

4. MEAN-VARIANCE ECONOMIC MPC

Now turning to the economic MPC problem with optimization horizon N, we seek to find $\mathbf{U_p} \in R_+^{n_{slots} \times N}$ of nominal prices that minimize the mean-variance objective function. The objective function consists of two terms. The mean term (the *economic* term) relates to the click maximization:

$$\phi_{eco} = \mathbb{E}_\omega[n_E^{tot}(\mathbf{U_p})]. \tag{9}$$

Here we have expressed that n_E^{tot} is a function of $\mathbf{U_p}$. The expectation is calculated from n_ω forecasted values of the total number of clicks obtained over the N-day optimization horizon. The forecasts are constructed from Bayesian sampling of the cost and clicks models presented in section 5.1 and 5.2

The "variance term" represents the expected magnitude by which the realized cost deviates (in a mean-square sense) from the reference cost trajectory, which is the N-day trajectory that brings the cost back onto the target cost (see Figure 3). It is calculated as a weighted sum (via $\mathbf{Q}$) over the N-day horizon, where for each day we calculate the expected mean of squared deviations (over samples ω) from the desired cost (reference cost) that day

$$\phi_{var} = \sum_{n=1}^{N} \left[q_n^2 \frac{1}{n_\omega} \sum_{\omega=1}^{n_\omega} [c_\omega^{acc}(n) - y_{ref}(n)]^2 \right]. \tag{10}$$

Here the accumulated cost function is of course a function of the nominal bids, although this is not explicitly expressed to save notation. It could alternatively be expressed as

$$\phi_{var} = \frac{1}{n_\omega} \sum_{\omega=1}^{n_\omega} ||\mathbf{c}_\omega^{\mathbf{acc}} - \mathbf{y_{ref}}||_{\mathbb{Q}}^2 \tag{11}$$

in which the outer summation is done over "Bayesian worlds" ω, and for each ω we consider the traditional squared deviation of the trajectory relative to the reference trajectory.

Combining the above terms, we get the total objective function

$$\phi_{MV} = -\alpha_{MV}\phi_{eco} + (1-\alpha_{MV})\left[\phi_{var} + \phi_{\Delta u}\right], \tag{12}$$

$\alpha \in [0;1]$. Notice the minus sign in front of ϕ_{eco}, effectively making it a click maximization objective. As an additional variance term we have also added the intra-horizon fluctuation in the control input sequence (the manipulated variable) $\phi_{\Delta u} = \sum_{n=1}^{N} ||\Delta \mathbf{u_p}(n)||^2$, where $\mathbf{u_p}(n) \in \mathbb{R}_+^{n_{slots}}$ is the n'th column of $\mathbf{U_p}$.

The MPC problem can now be stated as

$$\min_{\boldsymbol{U_p}} \quad \phi_{MV}(\boldsymbol{U_p}) \tag{13}$$

subject to

$$x_{n+1} = A_k x_n + B_k u_n + E_k d_k, \quad n = 1, \dots, N \tag{14}$$

$$u_p^{min} < \mathbf{u_p}(n) < u_p^{max}, \quad n = 1, \dots, N \tag{15}$$

$$\Delta u_p^{min} < \Delta \mathbf{u_p}(n) < \Delta u_p^{max}, \quad n = 1, \dots, N \tag{16}$$

where the state x_n would be a two-dimensional vector of accumulated cost and clicks the for campaign. We have not done the trouble of deriving the dynamical systems matrices of equation (14) in quite such a compact form as states above because we have n_ω different versions of the matrices at any given every time point k and the notation therefore gets a little too cluttered. But rest assured, these have indirectly been derived in section 5.2.2 and 5.1.2 where we find expressions for both the expected cost and click trajectories.

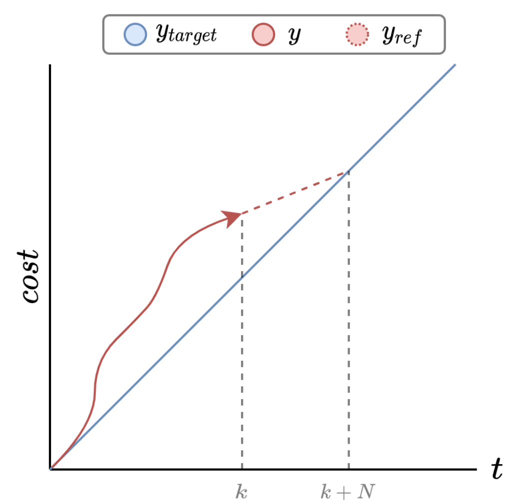

Fig. 3. Illustration of the cost variance control which shows how y_{ref} is chosen as the trajectory that brings the cost back onto y_{target} in N steps.

5. MODELLING

In this section we go over all model components required to compute the mean and variance terms for the MPC objective function. In section 5.1 we show how to calculate the variance objective of equation (10) while section 5.2 shows how to derive the expected number of clicks from equation (9).

5.1. Cost Modelling

5.1.1. *Cost linearization*

To be able to steer the cost along a predefined reference trajectory, the controller needs a reliable model of the daily cost in an ad slot as a function of nominal bid price, i.e. we wish to model $c_i(u_{p_i})$. We will assume a linear relation between nominal bid price and total daily cost in an ad slot. The linear models are estimated separately for each ad slot i, keeping only the past $L = 14$ days of nominal bids and realized costs, i.e. on day k we consider a batch of the L most recent nominal bid prices and their corresponding realized costs

$$\mathbf{u}_{\mathbf{p}_i}^L(k), \quad \mathbf{c}_i{}^L(k) \in \mathbb{R}_+^L. \tag{17}$$

The linearization of the cost model is *local* meaning that we will be subtracting the weighted mean of the nominal bids

$$u_{p_i}^*(k) = \sum_{l=1}^{L} \frac{w_l u_{p_i}(k-l)}{\sum_{l'} w_{l'}}, \tag{18}$$

with $\mathbf{w}$ being a vector of decreasing weights, thereby placing greater emphasis on more recent observations. Thus, the regression model on day k is centered around $u_{p_i}^*(k) \in \mathbb{R}_+$, i.e. we consider past nominal bids relative to $u_{p_i}^*(k)$

$$\widetilde{\mathbf{u}_{\mathbf{p}}}_i(k) = \mathbf{u}_{\mathbf{p}_i}^L(k) - u_{p_i}^*(k). \tag{19}$$

The local linear model on day k can then be states as

$$c_i = a_i \cdot \widetilde{u_{p_i}} + b_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2) \tag{20}$$

i.e.

$$c_i \sim \mathcal{N}(a_i \cdot \widetilde{u_{p_i}} + b_i, \sigma^2). \tag{21}$$

where we have dropped the explicit dependence on k to make the notation less cluttered. The intercept b_i is the expected cost in ad slot i should we make a nominal bid equal to the weighted mean of the past L days of nominal bids.

The estimation is handled by PyStan[2], which is capable of evaluating the log posterior density function and its gradient function in order to perform the MCMC method Hamiltonian Monte Carlo. The weighting is implemented in PyStan by a simple weighting of the log posteriors. The results of the inference can be examined by plotting the regression lines whose parameters are sampled from the posterior distributions, which is shown in Figure 4.

[2] A Python package for Bayesian inference using the No-U-Turn sampler, a variant of Hamiltonian Monte Carlo.

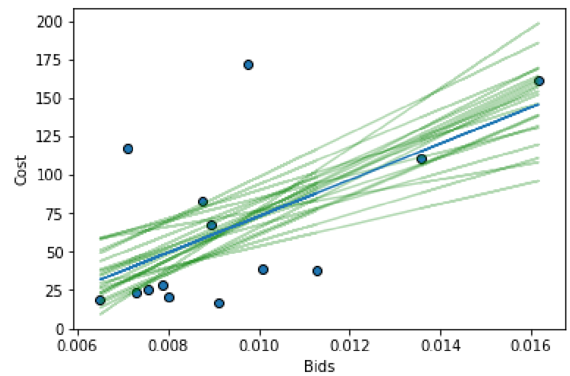

Fig. 4. The figure shows $n_\omega = 20$ Bayesian samples of weighted linear regression models, fitted using bid and cost observations from the previous 14 days.

The series of sampled values for the cost gradients a is assessed in the trace and posterior distribution plot shown in Figure 5.

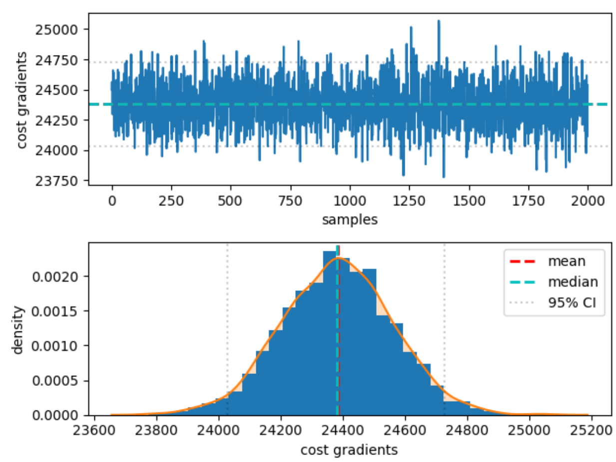

Fig. 5. Trace and posterior distribution for the cost gradients. Here the number of samples $n_\omega = 2000$ to show that the samples are well converged (the top plot appears to be stationary).

5.1.2. *Deriving the variance objective*

The MCMC estimation yields n_ω samples of the slope and intercept parameters for each ad slot. We therefore consider parameter matrices

$$\mathbf{A}, \ \mathbf{B} \in \mathbb{R}_+^{n_\omega \times n_{slots}} \tag{22}$$

of slopes and intercepts as in equation (21). As the output from the MPC optimization is a matrix $\widetilde{\mathbf{U}} \in \mathbb{R}^{n_{slots} \times N}$ of excess bids (relative to the weighted mean of past bids, see equation (19) which is for a particular ad slot), then the excepted daily cost can be calculated as

$$\mathbf{C} = \mathbf{A}\widetilde{\mathbf{U}} + \mathbf{B}\mathbb{1}_{n_{slots} \times N}, \tag{23}$$

where the excepted daily cost matrix $\mathbf{C} \in \mathbb{R}_+^{n_\omega \times N}$ contains the expected daily cost for all N days and for all n_ω model samples. The last term is a matrix multiplication of the intercept matrix and a matrix of ones with the specified dimensions. The multiplication effectively functions as a summation over ad slot intercepts, while simultaneously extending these along the time dimension, i.e. the elements in the resulting matrix product $\mathbf{B}\mathbb{1}_{n_{slots} \times N} \in \mathbb{R}_+^{n_\omega \times N}$ are the sums of all ad slot intercepts, and all columns are identical since we have just repeated the these intercept sums along the optimization horizon dimension, as there is no reason to believe the intercept will evolve in any particular direction within the optimization horizon N.

From here it is simple to construct an expected accumulated cost matrix as

$$\begin{aligned}\mathbf{C}^{acc} &= \mathbf{C}\mathbf{I}_u + \mathbf{C_k} \\ &= \left(\mathbf{A}\widetilde{\mathbf{U}} + \mathbf{B}\mathbb{1}_{n_{slots} \times N}\right)\mathbf{I}_u + \mathbf{C_k},\end{aligned} \tag{24}$$

with $\mathbf{I}_u$ being an upper triangular matrix of ones and $\mathbf{C_k}$ is the total realized amount of money spent up until day k. Each row of $\mathbf{C}^{acc} \in \mathbb{R}_+^{n_\omega \times N}$ corresponds to an expected cost trajectory over the N-day horizon.

For each expected accumulated cost trajectory we can now subtract the new N-day reference trajectory to get the cost deviation matrix (d for deviation)

$$\mathbf{C}_d^{acc} = \mathbf{C}^{acc} - \mathbf{Y_{ref}}, \tag{25}$$

with $\mathbf{Y_{ref}} \in \mathbb{R}_+^{n_\omega \times N}$ being the $\mathbf{y_{ref}}$ trajectory repeated in every row. In order to penalize deviations further out in the future (closer to the terminal date) $\mathbf{C}_d^{acc}$ is multiplied with a diagonal matrix $\mathbf{Q}$ of increasing and positive weights, i.e. $\mathbf{C}_d^{acc}\mathbf{Q}$. Lastly, the sum of squares of all these weighted cost deviation terms is calculated using CVXPY's `sum_squares()` method, thus obtaining a scalar optimization measure.

5.2. Click modelling

5.2.1. CPC estimation

In order to perform the click rate estimation it is assumed that the daily click count n_E (for simplicity we have dropped the index i that specifies the particular ad slot) is the realization of a random variable with an underlying Poisson distribution. Then, the probability mass function describing the click count can be stated as

$$\pi(n_E|c, CPC^{-1}) = \frac{\left(\frac{c}{CPC}\right)^{n_E} \exp(-\frac{c}{CPC})}{n_E!}, \tag{26}$$

where CPC defines the cost-per-click. By applying a prior distribution $\pi(CPC^{-1})$ on the inverse CPC we can use Bayesian inference in order to derive the posterior probability $\pi(CPC^{-1}|n_E, c)$. Since the Gamma distribution is the conjugate prior of the Poisson distribution, the prior on the inverse CPC is assumed to be given by Gamma(α_0, β_0). Then, by applying Bayes theorem

$$\pi(CPC^{-1}|n_E, c) = \frac{\pi(n_E|c, CPC^{-1})\pi(CPC^{-1})}{\pi(n_E|c)}, \tag{27}$$

it can be shown that the inverse CPC is distributed according to Gamma$(\alpha_0 + n_E, \beta_0 + c)$, which yields the following recursive update scheme for the shape and scale parameter

$$CPC^{-1} \sim \text{Gamma}\left(\alpha_0 + \sum_{j=1}^{k} n_E(j), \beta_0 + \sum_{j=1}^{k} c(j)\right). \tag{28}$$

By letting the parameters follow a Markov process with the state $[\alpha\ \beta]_k^T$ and initializing them by $[\alpha\ \beta]_0^T = [\alpha_0\ \beta_0]^T$ the updating scheme can simply be stated as

$$\begin{bmatrix}\alpha \\ \beta\end{bmatrix}_k = \begin{bmatrix}\alpha \\ \beta\end{bmatrix}_{k-1} + \begin{bmatrix}n_E \\ c\end{bmatrix}_k \tag{29}$$

A forgetting factor $\lambda \in (0; 1)$ is introduced in order to reduce the effect of old observations and increase the effect of newer ones. Thus, the parameters end up being updated according to

$$\alpha(k) = \lambda^k \alpha_0 + \sum_{j=1}^{k} \lambda^{k-j} n_E(j), \tag{30}$$

$$\beta(k) = \lambda^k \beta_0 + \sum_{j=1}^{k} \lambda^{k-j} c(j), \tag{31}$$

which we can also define on the form

$$\begin{bmatrix}\alpha \\ \beta\end{bmatrix}_k = \lambda \begin{bmatrix}\alpha \\ \beta\end{bmatrix}_{k-1} + \begin{bmatrix}n_E \\ c\end{bmatrix}_k \tag{32}$$

Using equation (28) we can draw n_ω samples of the inverse CPC for each ad slot. These are used in deriving the linear click maximization term below.

5.2.2. Deriving the click objective

The expected number of daily clicks in a particular ad slot is calculated as the clicks-per-cost (inverse CPC) times the expected cost

$$\begin{aligned}n_E &= \frac{1}{CPC_i} c \\ &= \frac{1}{CPC}\left(a\widetilde{u}_p + b\right) \\ &= \frac{a}{CPC}\widetilde{u}_p + \frac{b}{CPC}\end{aligned} \tag{33}$$

where we have dropped both the day index k and the ad slot index i on all quantities to keep the notation less cluttered.

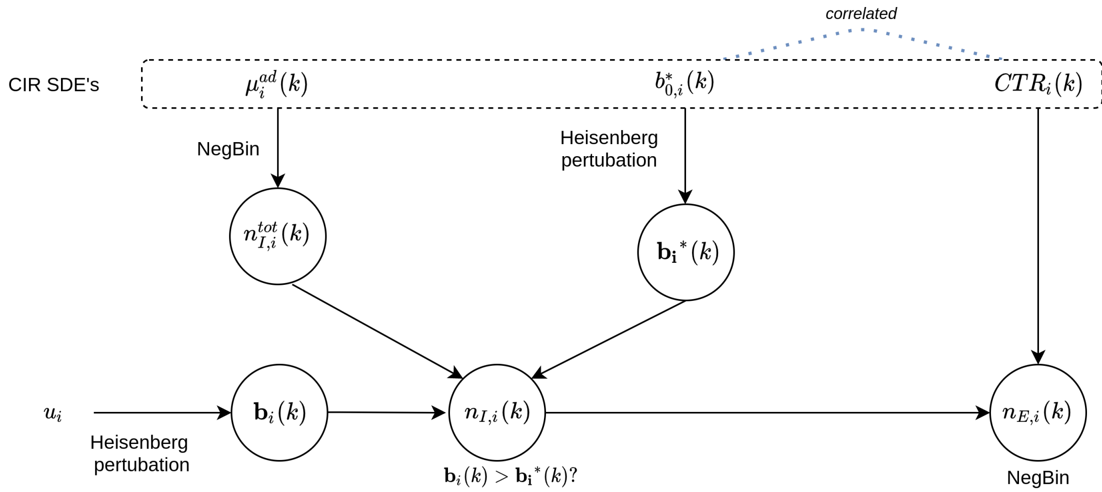

Fig. 6. Simulation illustration. As shown in the top layer, the three underlying market parameters, i.e. the ad opportunities rate, the highest competitive bid, and the click-through rate, evolve according to a Cox-Ingersoll-Ross SDE (equation (40)). Notice that all these are scalar processes for each ad slot, and whereas $\mu_i^{ad}(k)$ and $b_{0\,i}^*(k)$ are relatived to the auction, $CTR_i(k)$ is instead related to the following ad serving (see Figure 1). The daily number of ad opportunities, $\mathbf{n}_{I,i}^{tot}(k)$, in each ad slot is drawn from a Negative Binomial distribution (equation (39)), while the highest competitive bid in each auction is drawn as a Heisenberg perturbation from a common nominal bid price for that ad slot (with a fixed bid uncertainty of 50%.). The number of awarded impressions in each ad slot is then trivially determined by comparing our advertiser's bids with the highest competitive bids. The number of clicks are simulated from a Negative Binomial distribution with mean value equal to click-through rate times impressions.

To again incorporate the MCMC sampling dimension (n_ω), the matrices from section 5.1.2 can be reused to construct a matrix $\mathbf{n_E} \in \mathbb{R}_+^{n_\omega \times N}$ of expected daily clicks as

$$\mathbf{n_E} = \mathbf{P}\widetilde{\mathbf{U}} + \mathbf{S}\mathbb{1}_{n_{slots} \times N}, \tag{34}$$

where $\mathbf{P}, \mathbf{S} \in \mathbb{R}_+^{n_\omega \times n_{slots}}$ are matrices with elements

$$\left[\frac{a_i^\omega}{CPC_i^\omega}\right]_{\omega,i} \quad \text{and} \quad \left[\frac{b_i^\omega}{CPC_i^\omega}\right]_{\omega,i}$$

with $\omega = 1, .., n_\omega$ and $i = 1, .., n_{slots}$, corresponding to the click slope coefficient and click intercept of equation (33). The matrix $\mathbf{n_E}$ is a $n_\omega \times N$ of expected daily clicks in each sample scenario ω, and summing each row yields $n_{E,\omega}^{tot}$, $\omega = 1, .., n_\omega$. The mean objective is then simply calculated as the mean of these total number of clicks over the N-day horizon which is exactly the scalar optimization measure that is used in the MPC computation

$$\phi_{eco} = \mathbb{E}_\omega[n_E^{tot}] = \frac{1}{n_\omega}\sum_{\omega=1}^{n_\omega} n_{E,\omega}^{tot}. \tag{35}$$

6. SIMULATION

In order to rigorously assess the performance of the controller described in the previous sections, we need to be able to interact with a synthetic RTB auction system that captures the essential dynamical patterns of a real-life system. Writing down the governing dynamical systems equations from first principles is no easy task, but here we present how we chose to simulate the three fundamental quantities relevant to the auction (and ad serving) as stochastic realizations of probability distributions whose underlying parameters evolve according to a stochastic differential equation (SDE).
The three necessary stochastic quantities are:

- The daily number of total ad opportunities/auctions

$$n_{I,i}^{tot}(k), \quad i = 1, ..., n_{slots}. \tag{36}$$

  This is the maximal number of impressions our advertiser can win in each adslot (here k specifies the day).

- The highest competitive bid price vectors

$$\mathbf{b}_i^*(k), \quad i = 1, ..., n_{slots}, \tag{37}$$

  where the dimensions are $n_{I,i}^{tot}(k)$.

- The number of clicks

$$n_{E\,i}(k), \quad i = 1, ..., n_{slots}, \tag{38}$$

  which depends on the number of impressions awarded and the underlying click-through rate.

To enforce the unpredictability of the number of ad opportunities we use an overdispersed Poisson distribution, i.e.

$$\mathbf{n}_{I,i}^{tot}(k) \sim \text{NegBin}(\mu_i^{ad}(k), \phi_i^{ad}(k)), \tag{39}$$

where the alternative parameterization of the Negative Binomial distribution contains the mean μ, and the dispersion ϕ, which controls the extra width of the distribution beyond Poisson. The smaller ϕ is, the broader the distribution is, since the variance is given by $\mu + \frac{\mu^2}{\phi}$. We assume that the mean value parameter in each ad slot follows a Cox-Ingersoll-Ross stochastic differential equation which in generic form is given by

$$dx_t = -\lambda(x_t - \theta)dt + \sigma\sqrt{x_t}dW_t, \tag{40}$$

where $\theta > 0$, $\sigma > 0$. It is a mean-reverting process where the diffusion term is proportional to the square root of the state, thus ensuring a positive ad opportunity expectation, $\mu_i^{ad}(k)$. The dispersion is chosen to be proportional to $\mu_i^{ad}(k)$, i.e. $\phi_i^{ad}(k) = \phi_0 \cdot \mu_i^{ad}(t)$, such that the additional variance scales linearly with the expectation.

Moving forwards, the highest competitive bid prices $\mathbf{b}_i^*(k)$ for all auctions in ad slot i can be described as Heisenberg perturbations from a common nominal bid price, $b_{0i}^*(k)$ which is also assumed to evolve according to equation (40).

Finally, we model the number of clicks in each ad slot i as

$$n_{Ei}(k) \sim \text{NegBin}(\mu_i^E(k), \phi_i^E(k)), \tag{41}$$

with $\mu_i^E(k) = CTR_i(k) \cdot n_{Ii}(k)$ and $\phi_i^E(k) \propto \mu_i^E(k)$. Here the click-through rate $CTR_i(k)$ is also assumed to evolve according to equation (40).

In order to model a shared perception of the quality of an ad slot, we introduce a strong positive correlation in the Wiener processes of CTR_i and b_{0i}^*. An illustration of the simulations setup is shown in Figure 6.

The mean-variance economic MPC implementation is done in Python and constructed in the class `MPC` which takes input in the form of initial values for highest competitive bid, CTR, ad opportunity rate, initial bid price and associated bid uncertainty together with the underlying parameters belonging to the stochastic differential equations governing our synthetic RTB auction system.

7. RESULTS

To show the potential of our mean-variance economic MPC a campaign is set up with a budget of 6000. The campaign has length of $T = 25$ days and prediction horizon of $N = 5$ days is used and there are 4 available ad slots. In order to use the economic MPC controller on this campaign, an initial period is constructed in which bidding is done without the use of MPC, i.e. the control signal is determined from a slight perturbation of the CTR values. After this *warm-up* we now have batch of historical costs and bids which is necessary for cost linearization. Now, starting on the $k = 0$ day we first evolve the underlying market parameters, i.e. the expected number of impression opportunities, expectation value of competitor bids and the CTR values for each ad slot. From here the cost, impressions and clicks from the auctions are observed before the α and β parameters in the Heisenberg bidding are updated in order to set the bid uncertainty and sample the inverse CPC's. In this run, $n_\omega = 50$ CPC and cost samples are used, where cost samples are obtained via the weighted Bayesian regression described in section 5.1.1 using the last 14 days of observations. By determining the reference trajectory y_{ref} based on the cost and the defined target cost y_{target}, the mean and variance objective can be constructed. The economic MPC problem is then solved through CVXPY by inputting our defined constraints. Finally, using the control signal U_p the new bid is set by adding back the weighted mean of the nominal bid u_p^*. Figure 7 illustrates the realized spending for the 25 day horizon. The plot shows that the campaign meets the budget pacing objective, which was defined in section 2. The budget is used smoothly over the lifetime of the campaign and the total budget is used at the campaigns expiration. The plot also zooms in on the period from day $k = 10$ to day $k + N = 15$ from which it is seen that the accumulated cost follow the reference trajectory y_{ref} back onto the target trajectory y_{target}. To monitor that the market dynamics and

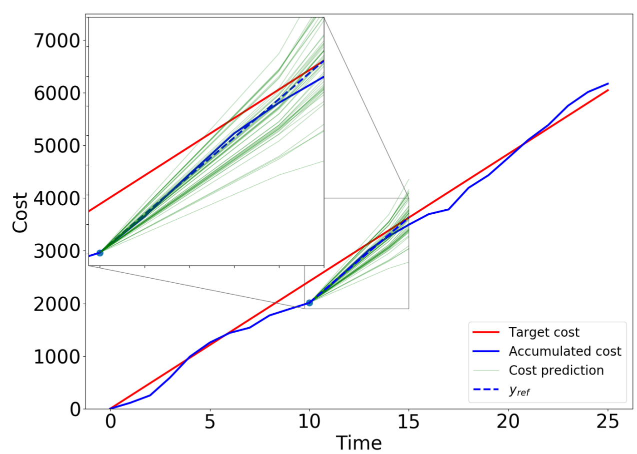

Fig. 7. Accumulated cost plotted together with the target and reference trajectory. The n_ω sampled cost predictions are shown for day $k = 10$ and $N = 5$ days ahead, which is the specified prediction horizon.

our involvement in the market are consistent, the visual control room shown in Figure 8 is constructed. From these plots, a nice correlation is seen between the given bids and the CTR values. In ad slots with high estimated CTR values, the bids are increased and decreased in those with low CTR values. This leads to a higher number of auctions being won in these ad slots, thus achieving more clicks, which is also reflected in the plot showing the bid uncertainties. Due to the increased

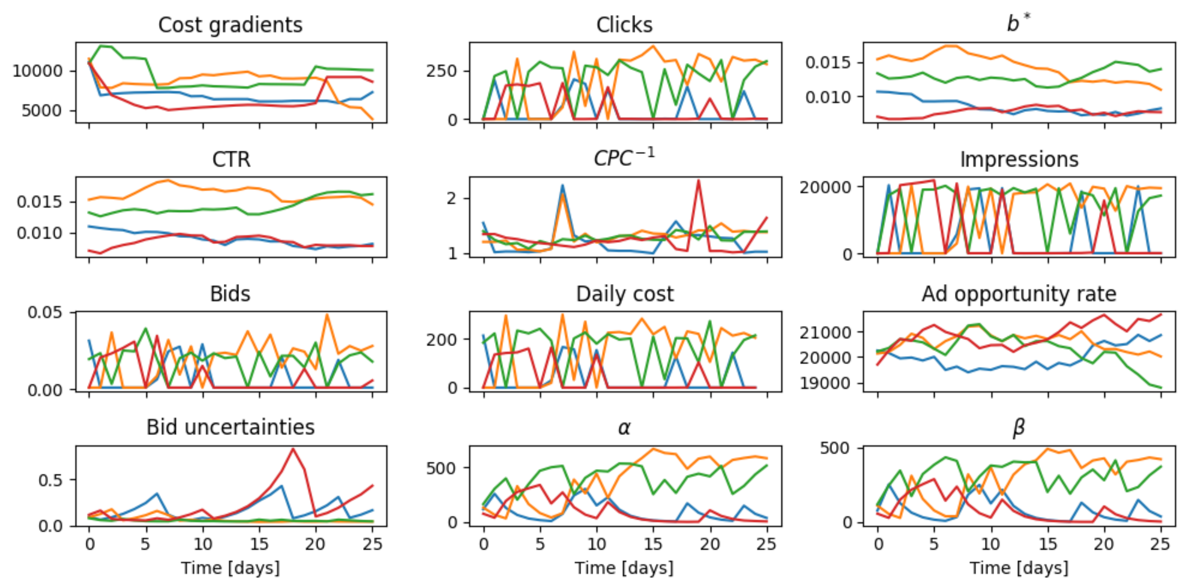

Fig. 8. Visual control room, which here monitors the market dynamics and our activity in the market with 4 available ad slots.

number of clicks and impressions in these ad slots, the bid uncertainties are vanishingly small compared to the ad slots in which almost no auctions are won. In the ad slots where few auctions are won, we see a surging trend in the bid uncertainties, which is a demonstration of the effect of the Heisenberg bidding strategy. Although the CTR in these ad slots are very low and this could result in bids converging to zero, it is seen that the Heisenberg bidding strategy ensures us that at least a few clicks are generated when the bid uncertainty increases. Obtaining these clicks reduces the bid uncertainty again and thus exploration is carried out throughout the life time of the campaign. From the impressions plot, we notice that for this particular run, we either win all auctions or lose all auctions. It could therefore be desirable to constrain the controller to make slightly smaller bid changes.

8. CONCLUSION

In this paper we have presented a model for a real-time-bidding auction system, that captures the essential dynamic patterns of a real-life system. We use Mean-Variance Economic Model Predictive Control (Mean-Variance Economic MPC) to manipulate our bids such that the advertiser gets the maximum number of clicks while spending his budget smoothly over the life of the campaign.
The simulation validation of the Economic MPC performed on a campaign setup on the synthetic RTB auction system verifies the effectiveness of our implementation.

Future research of economic MPC applied to advertising systems should include an increased focus on computation time. The current controller has an unnecessarily large bottleneck in the MCMC samping in Stan. For a future version of the controller it is desirable to discard Stan entirely by implementing a conjugate prior for the linear weighted regression model, but also the cost linearization itself might be reconsidered, and especially how it relates to the MPC controller. The current controller receives no explicit information regarding the bid uncertainties that will be used in perturbing the nominal bids. Smaller bid uncertainties lead to narrower Heisenberg distributions and less exploration, yielding either extremely high or extremely low win ratios, since the narrow Heisenberg distribution will be either entirely above or entirely below the competitive bid distribution. Finding a way to pass uncertainty information from the cost/click estimator into the controller is therefore highly desirable.

Lastly, it would be interesting to investigate the effect of varying the mean-variance objective parameter α_{MV}. This would hopefully sketch out an efficient frontier which would indicate just how many clicks can be obtained for a given tolerance of cost fluctuation around the target trajectory.

9. ACKNOWLEDGMENTS

We thank John Bagterp Jørgensen for encouraging us to investigate how to implement MPC in a computational advertising setting, and for suggesting relevant literature on randomized bidding.

10. SUPPLEMENTARY MATERIALS

All the code used in this project is available on GitHub via this link: github.com/bhastrup/mpc-project.

References

[1] "Online advertising market - growth, trends and forecast," 2020, mordorintelligence.com/industry-reports/online-advertising-market.

[2] Kurt Wagner, "Digital advertising in the us is finally bigger than print and television.," 2019, https://www.vox.com/2019/2/20/18232433/digital-advertising-facebook-googlegrowth-tv-print-emarketer-2019.

[3] Kuang-Chih Lee, Ali Jalali, and Ali Dasdan, "Real time bid optimization with smooth budget delivery in online advertising," in *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, New York, NY, USA, 2013, ADKDD '13, Association for Computing Machinery.

[4] Berkhin P. Anderson B. Chen, Y. and N. R. Devanur, "Real-time bidding algorithms for performance-based display ad allocation," 2011, https://www.vox.com/2019/2/20/18232433/digital-advertising-facebook-googlegrowth-tv-print-emarketer-2019.

[5] Niklas Karlsson, "Control problems in online advertising and benefits of randomized bidding strategies," *European Journal of Control*, vol. 30, pp. 31 – 49, 2016, 15th European Control Conference, ECC16.

[6] Rong Y. Wang J. Zhu T. Zhang, W. and X. Wang, "Feedback control of real-time display advertising," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 2016, pp. 407–416.

[7] J.M. Maciejowski, *Predictive Control with Constraints.*, Prentice Hall, England., 2002.

[8] Karl Johan Åström and Richard M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, 2004.

[9] R. Halvgaard, *Model Predictive Control for Smart Energy Systems.*, Ph.d, DTU., 2014.

[10] Steven Diamond and Stephen Boyd, "Cvxpy: A python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[11] N. Karlsson, "Adaptive control using heisenberg bidding," in *2014 American Control Conference*, 2014, pp. 1304–1309.