FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

FACULTY OF SCIENCES

CHAIR IN APPLIED ANALYSIS FAU

# MODEL PREDICTIVE CONTROL WITH RANDOM BATCH METHODS FOR A GUIDING PROBLEM

**Dongnam Ko**
Department of Mathematics
Catholic University of Korea

**Enrique Zuazua**
Chair in Applied Analysis (AvH Professorship)
FAU Erlangen-Nürnberg

## 1. A guiding problem

Consider a herding problem: a few shepherd dogs (drivers) try to control a herd of sheep (evaders). The dynamics of drivers $(\mathbf{y}_j, \mathbf{u}_j)$ and the evaders $(\mathbf{x}_i, \mathbf{v}_i)$ can be described as in [EIZ2016] and [KZ2020]:

$$\begin{cases} \dot{\mathbf{x}}_i = \mathbf{v}_i, \quad i = 1, \dots, N, \\ \dot{\mathbf{v}}_i = \dfrac{1}{N-1}\displaystyle\sum_{k=1, k\neq i}^{N} a(\mathbf{x}_k - \mathbf{x}_i)(\mathbf{v}_k - \mathbf{v}_i) + \dfrac{1}{N-1}\sum_{k=1, k\neq i}^{N} g(\mathbf{x}_k - \mathbf{x}_i)(\mathbf{x}_k - \mathbf{x}_i) \\ \qquad -\dfrac{1}{M}\displaystyle\sum_{j=1}^{M} f(\mathbf{y}_j - \mathbf{x}_i)(\mathbf{y}_j - \mathbf{x}_i), \ i = 1, \dots, N, \\ \dot{\mathbf{y}}_j = \mathbf{u}_j(t), \quad j = 1, \dots, M \\ \mathbf{x}_i(0) = \mathbf{x}_i^0, \quad \mathbf{v}_i(0) = \mathbf{v}_i^0, \quad \mathbf{y}_j(0) = \mathbf{y}_j^0. \end{cases}$$

Here, $\mathbf{u}_j(t)$ are the control input we want to find. For a concrete numerical simulation, we may set the interactions $a$, $f$ and $g$ as

$$a(\mathbf{x}) := 1, \quad f(\mathbf{x}) := 4\exp(-8|\mathbf{x}|^2) \quad \text{and} \quad g(\mathbf{x}) := 2\left(1 - \frac{1}{3\sqrt{N}|\mathbf{x}|^2}\right).$$

Our control objective is to guide the evaders to a desired region. In particular, to find proper $\mathbf{u}_j(t)$, we may set the optimal control problem with the cost function

$$J(\mathbf{u}) := \int_0^T \left[\frac{1}{N}\sum_{k=1}^{N}|\mathbf{x}_k - \mathbf{x}_f|^2 + \frac{10^{-4}}{M}\sum_{j=1}^{M}|\mathbf{u}_j|^2 + \frac{10^{-4}}{M}\sum_{j=1}^{M}|\mathbf{y}_j - \mathbf{x}_f|^2\right] dt.$$

The problem is that the numerical simulation quickly becomes unfeasible as $N$ grows. The computational costs is $O(N^2)$ for the controlled trajectories and the gradient of a cost function. To reduce this, we suggest to combine an approximative model of the dynamics with randomness (Random Batch Methods) and a control design method to handle time-evolution error (Model Predictive Control).

## 2. Random Batch Methods

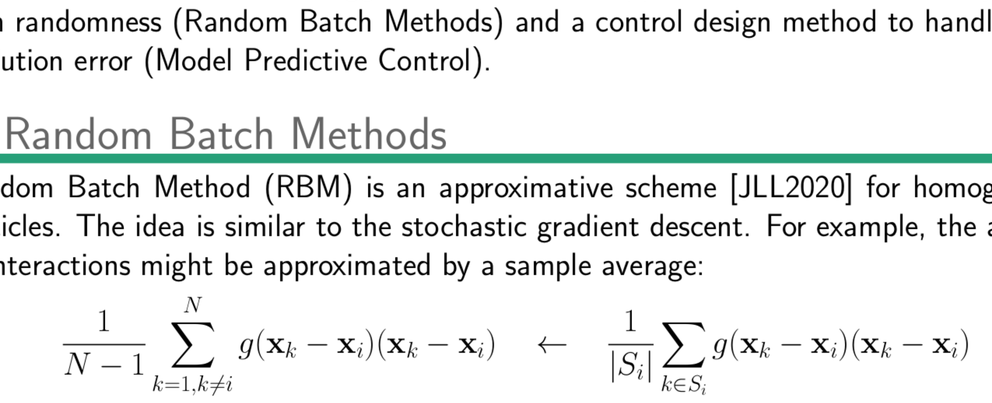
Random Batch Method (RBM) is an approximative scheme [JLL2020] for homogeneous particles. The idea is similar to the stochastic gradient descent. For example, the average of interactions might be approximated by a sample average:

$$\frac{1}{N-1}\sum_{k=1, k\neq i}^{N} g(\mathbf{x}_k - \mathbf{x}_i)(\mathbf{x}_k - \mathbf{x}_i) \quad \leftarrow \quad \frac{1}{|S_i|}\sum_{k\in S_i} g(\mathbf{x}_k - \mathbf{x}_i)(\mathbf{x}_k - \mathbf{x}_i)$$

for a randomly chosen set $S_i$. Of course, this induces an interaction network system, which is qualitatively different from the original system. Therefore, RBM changes $S_i$ for each time-discretization. Let $t_m := m\tau$. For each $m$, we split the set of particles $\{1, 2, \dots, N\}$ into random small subsets (batches) with $P$ particles (that we choose $P$ *a priori*):

$$\{1, 2, \dots, N\} = \mathcal{B}_1^m \cup \mathcal{B}_2^m \cup \cdots \cup \mathcal{B}_n^m, \quad |\mathcal{B}_1^m|, \dots, |\mathcal{B}_{n-1}^m| = P \quad \text{and} \quad |\mathcal{B}_n^m| \leq P.$$

Then, we have an approximative dynamics for each time interval $[t_{m-1}, t_m]$, where we set $S_i$ to be the batch $\mathcal{B}_j^m$ which contains $i$. In this way, the approximated dynamics becomes a switching network system. We may apply RBM also for the adjoint system of control problems. It turns out that the RBM-approximation of the adjoint system is identical to the adjoint system of the RBM-approximated model.

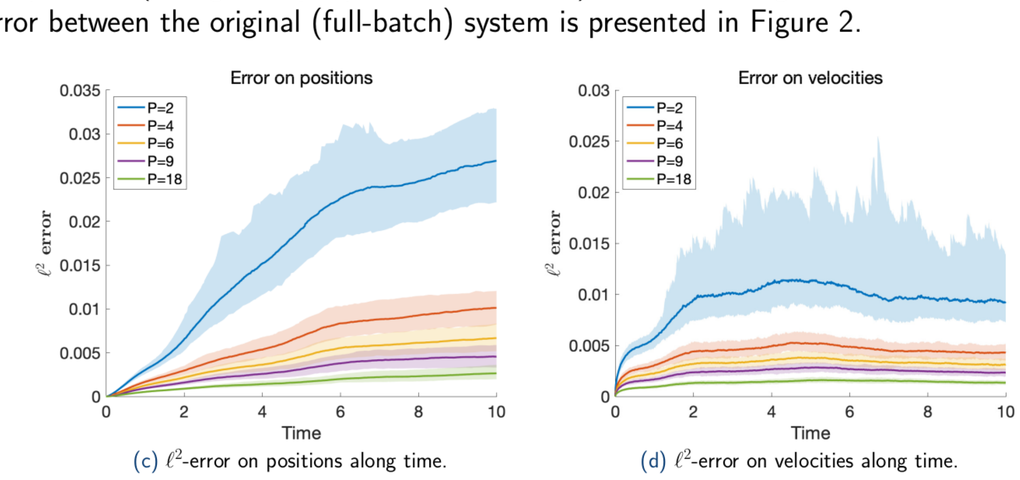



(a) One particle trajectory along time. (b) 36 particles positions at a time.

FIGURE 1. Distribution of RBM-approximated trajectories from 1000 simulations

## 3. Model Predictive Control

RBM is especially proper for the guiding problem since it estimates well the distribution of the particles (though not yet proved how it does) as shown in Figure 1. The time-evolution error between the original (full-batch) system is presented in Figure 2.



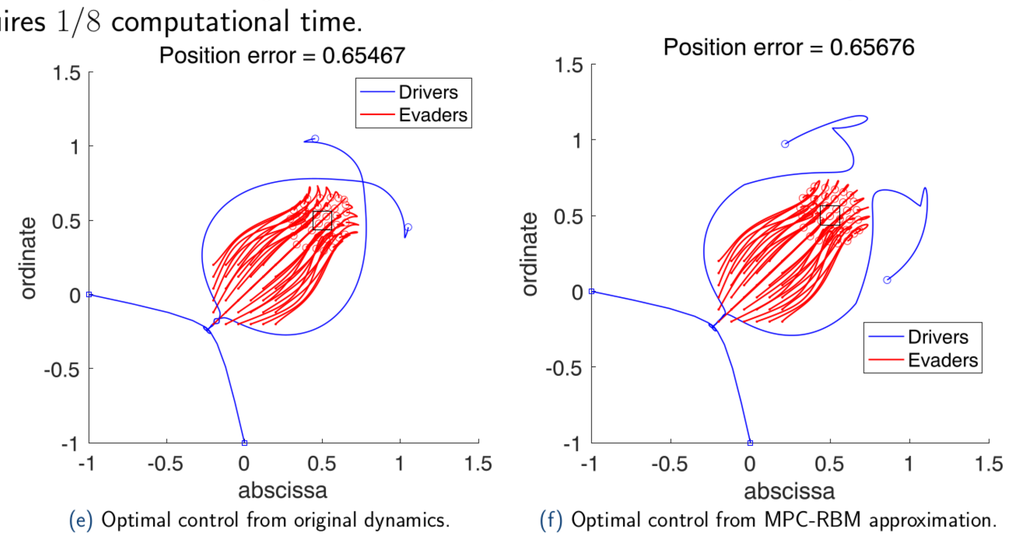

(c) $\ell^2$-error on positions along time. (d) $\ell^2$-error on velocities along time.

FIGURE 2. Distribution of the $\ell^2$-error along time from 1000 simulations

A first try of finding $\mathbf{u}_j$ using RBM might be an analogue of stochastic gradient descent; for each optimization step, we may choose different collection of random batches to get the optimal control of the original system. However, this algorithm commonly fails to converge in the guiding problem, probably due to highly nonlinear landscape of the cost function. Model Predictive Control (MPC) is a control design [GPM1989], mainly used for infinite-time horizon when we only have approximated models but not the original system. Note that, for a fixed RBM-approximated model, computing the optimal control is an easy task in $O(NP)$ computational complexity. MPC applies this control function to the original system for a short time duration. First, it computes the optimal control for $[0, T_p]$ with an approximated model, and applies it into the original system only for $[0, T_c] \subset [0, T_p]$. From the current data at $t = T_c$, we repeat the same process for $[T_c, 2T_c]$ to refresh the error periodically, and so on.

## 4. Numerical experiments

The following table shows the computational time and the costs of control functions from each simulation. The simulation is for 36 evaders and 2 drivers on the time interval $[0, 4]$ with $T_c = 1.5$ and $T_p = 3$. The control trajectories are shown in Figure 3, where two blue drivers properly capture 36 red evaders near $(0.5, 0.5)$ in two dimensional space. Note that RBM and MPC-RBM algorithms have only 2% differences from the optimal control but requires $1/8$ computational time.




(e) Optimal control from original dynamics. (f) Optimal control from MPC-RBM approximation.

FIGURE 3. Controlled trajectories simulated with original dynamics

| Adaptive GD | GD iter. | Cost $J$ | Comp. time | (Ratio) |
|---|---|---|---|---|
| Full system | 3123 | 0.6646 | 767.96 | (8.004) |
| RBM $(P = 2)$ | 2721 | 0.6665 | 95.95 | (1.000) |
| RBM $(P = 4)$ | 3054 | 0.6651 | 164.02 | (1.710) |
| RBM $(P = 6)$ | 3027 | 0.6651 | 214.05 | (2.231) |
| RBM $(P = 9)$ | 2604 | 0.6648 | 229.57 | (2.393) |
| RBM $(P = 18)$ | 2654 | 0.6648 | 392.71 | (4.093) |
| MPC-RBM $(P = 2)$ | 2983 | 0.6668 | 82.96 | (0.865) |

Table: The computation time (Comp. time) in the adaptive gradient descent algorithm for the guiding problem. The stopping criterion is set with tolerance $10^{-6}$ on the relative cost $(J_{new} - J_{old})/J_{old}$.

## Selected publications

[EIZ2016] ESCOBEDO, R., IBAÑEZ, A. AND ZUAZUA, E. (2016). Optimal strategies for driving a mobile agent in a "guidance by repulsion" model. *Commun. Nonlinear. Sci.*, 39:58–72.

[GPM1989] GARCÍA, C. E., PRETT, D. M. AND MORARI, M. (1989). Model predictive control: Theory and practice—A survey. *Automatica*, 25:335-348.

[JLL2020] JIN, S., LI, L. AND LIU, J.-G. (2020). Random Batch Methods (RBM) for interacting particle systems. *J. Comput. Phys.*, 400:108877.

[KZ2020] KO, D. AND ZUAZUA, E. (2020). Asymptotic behavior and control of a "guidance by repulsion" model. *Math. Models Methods Appl. Sci.*, 30:765–804.

caa-avh.nat.fau.eu

04/2021

DDS Department of DATA SCIENCE

Unterstützt von / Supported by

Alexander von Humboldt Stiftung/Foundation