Deusto
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
UAM Universidad Autónoma de Madrid

Computational strategies for the control of collective dynamics

Enrique Zuazua[1]

FAU - Erlangen

Joint work with Dongnam Ko and Daniel Veldman

Chair DYNAMICS, CONTROL AND NUMERICS FAU

European Research Council

Unterstützt von / Supported by Alexander von Humboldt Stiftung / Foundation

28.05.2021

Table of Contents

Motivation: Sheepdogs and sheep

Herding Problem: One or several sheepdogs steer a herd of sheep to a final destination.


Objective: Guide the evaders in the right direction and confine them in a given area.

Motivation: "Guidance by repulsion" model

Drivers try to guide the evaders to a given final destination

- **One driver + one evader**[1]
  - The driver induces a **repulsive force** on the evader.
  - The driver is **attracted** by the evader.
  - The driver guides the evader combining elementary motions: stop, move forward and rotate (left and right).
  - The driver (sheepdog) acts following the **instructions of a shepherd** (control).
- **One driver + multiple evaders.**
  - The single driver interacts with **the center of the flock of evaders**.
  - Evaders are mutually **attracted**.
- **Multiple drivers + multiple evaders** [2]
  - **Each driver** interacts with **each evader**.
  - **The shepherd coordinates** the motion of all drivers.

[1] R. Escobedo, A. Ibañez, E. Zuazua, 2016

[2] D. Ko, E. Zuazua, 2020.

True herding

Virtual herding

Multiple drivers/evaders model

$\mathbf{x}_i, \mathbf{v}_i$: the **position, velocity** of the ith evader ($i = 1, \dots, N$) in $\mathbb{R}^2$,
$\mathbf{y}_j$: the position of the jth **driver** ($j = 1, \dots, M$) in $\mathbb{R}^2$.

$$\begin{cases} \dot{\mathbf{x}}_i = \mathbf{v}_i, \quad i = 1, \dots, N, \\ \dot{\mathbf{v}}_i = \dfrac{1}{N-1} \displaystyle\sum_{k=1, k\neq i}^{N} a(\mathbf{x}_k - \mathbf{x}_i)(\mathbf{v}_k - \mathbf{v}_i) \qquad \leftarrow \text{velocity alignment} \\ \qquad + \dfrac{1}{N-1} \displaystyle\sum_{k=1, k\neq i}^{N} g(\mathbf{x}_k - \mathbf{x}_i)(\mathbf{x}_k - \mathbf{x}_i) \qquad \leftarrow \text{position flocking} \\ \qquad - \dfrac{1}{M} \displaystyle\sum_{j=1}^{M} f(\mathbf{y}_j - \mathbf{x}_i)(\mathbf{y}_j - \mathbf{x}_i), \quad i = 1, \dots, N, \leftarrow \text{evading from drivers} \\ \dot{\mathbf{y}}_j = \mathbf{u}_j(t), \quad j = 1, \dots, M \qquad \leftarrow \text{drivers are directly controlled} \\ \mathbf{x}_i(0) = \mathbf{x}_i^0, \quad \mathbf{v}_i(0) = \mathbf{v}_i^0, \quad \mathbf{y}_j(0) = \mathbf{y}_j^0. \end{cases}$$

Independent of how strongly the driver is attracted towards the evader, the shepherd can control its instinct to steer the driver according to the control strategy. This simplifies the equation for the driver.

Table of Contents

Optimal control

Goal: Simulate the locomotion of drivers controlling an ensemble of evaders?

MINIMISE!!!!

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

Note that we penalize the position of the drivers as well. This is known to lead to less oscillatory control strategies (Turnpike)

Some (very few) references:

- Problems on sheep gathering:
  Well-posedness of optimal control [Burger, Pinnau, Roth, Totzeck, Tse, 2016] and its simulations [Pinnau, Totzeck, 2018].
- Repelling birds from airports: [Gade, Paranjape, Chung, 2015],
- Hunting strategies: [Muro, Escobedo, Spector, Coppinger, 2011 and 2014],

Simulation

A numerically simulated optimal control with 36 evaders and 2 drivers toward the target $(0.5, 0.5)$ in the time horizon $[0, 4]$:

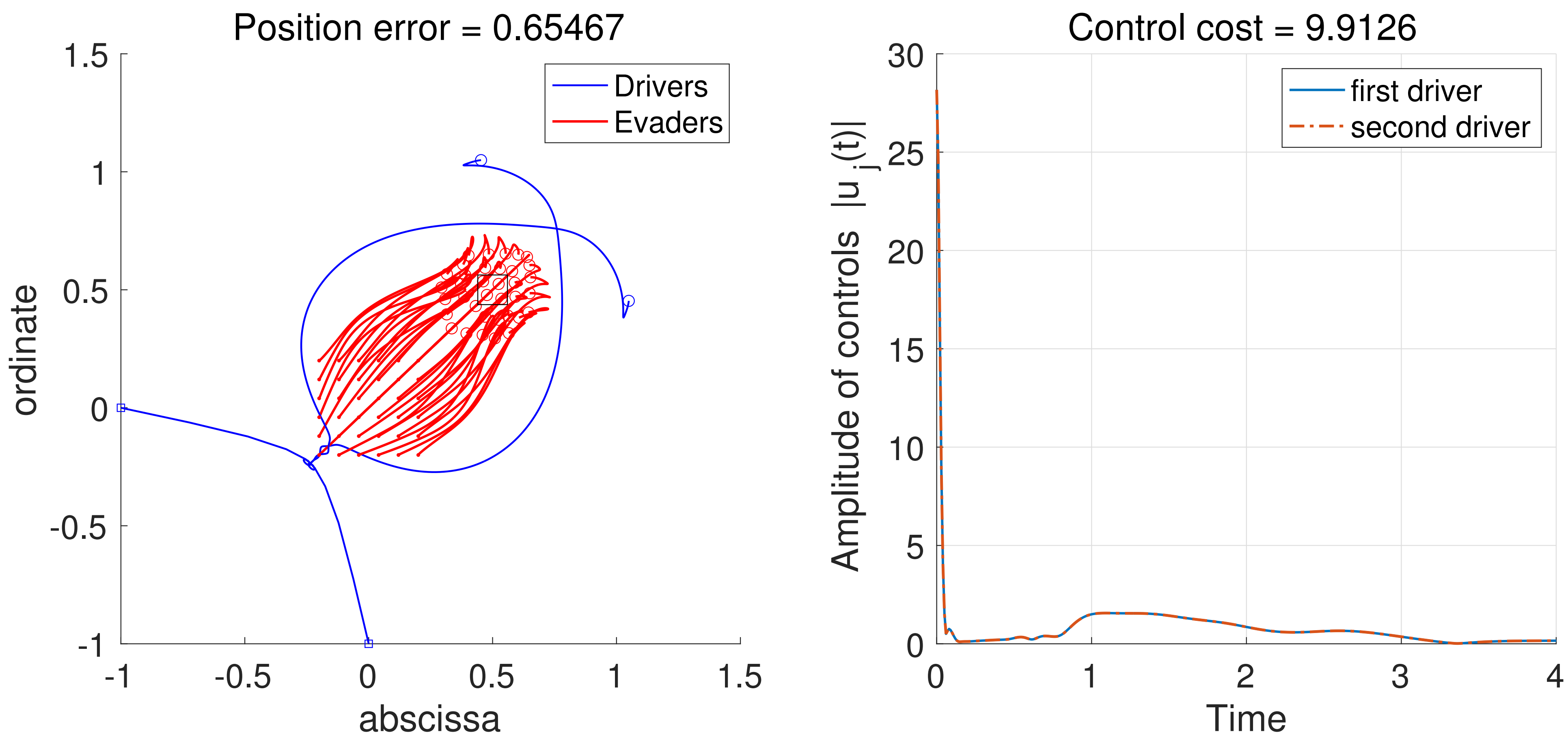
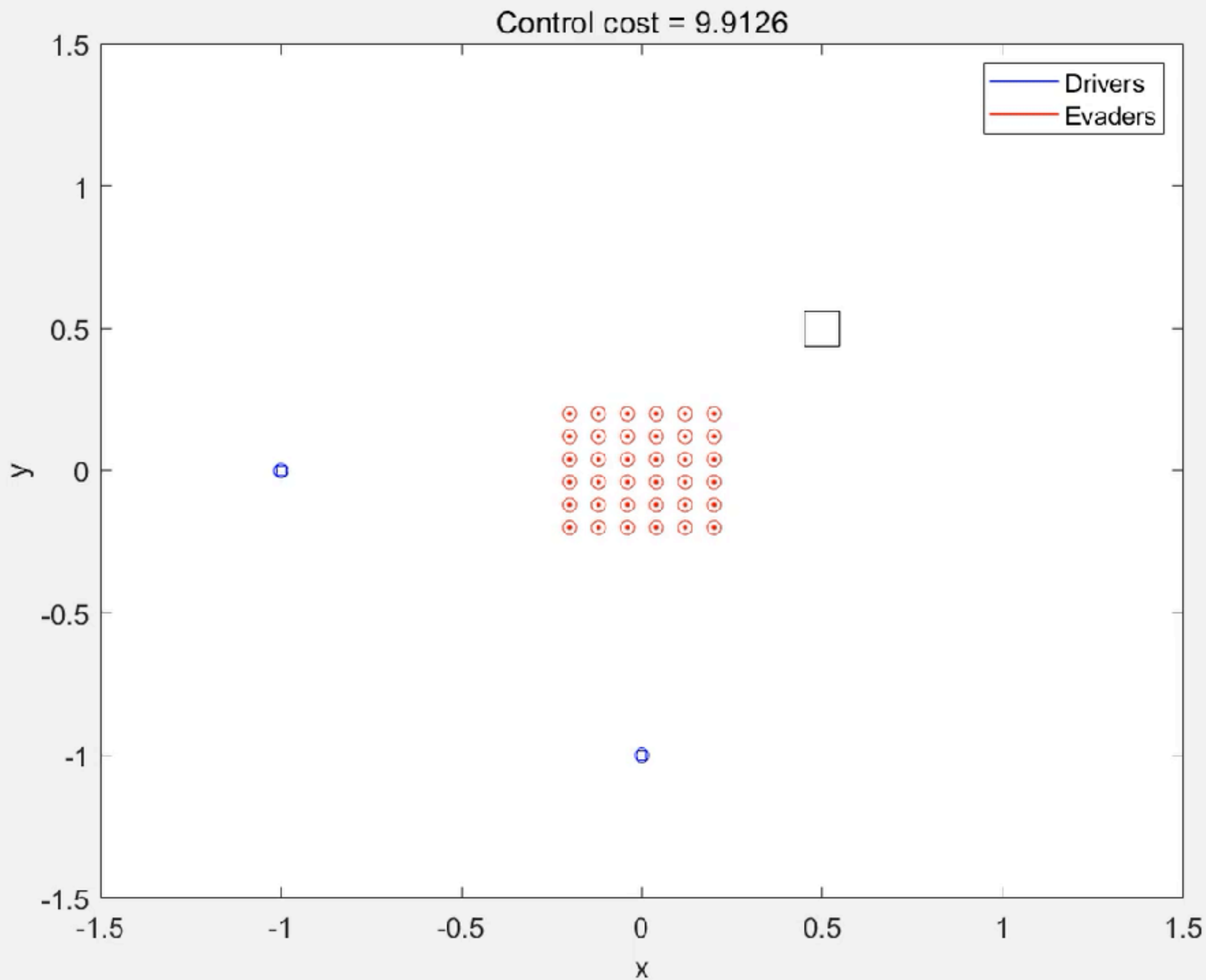


Figure: **Left**: trajectories in $2D$ space, **Right**: control function along time.

Two drivers starting from $(0, -1)$ and $(-1, 0)$.


Control cost = 9.9126
Drivers
Evaders
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5
x
y

Accelerating simulations

The computational complexity increases rapidly when the number of evaders N grows.
We propose an approximate control design combining:

1. **Random Batch Methods (RBM)** to approximate dynamics.[3]
2. **Model Predictive Control (MPC)** to correct the deviation introduced by the RBM[4].

[3] S. Jin, L. Li, J-G Liu, 2020.
[4] L. Grüne, J. Pannek, 2017.

Table of Contents

Approximative dynamics: Random Batch Methods (RBM)

Divide $[0, T]$ into subintervals

$$[0, T] = \bigcup_{m=1}^{M} [t_{m-1}, t_m], \quad 0 = t_0 < t_1 < \cdots < t_M = T.$$

We split the set of particles into N/P small random subsets (batches) with P particles:

$$\{1, 2, \ldots, N\} = \mathcal{B}_1^m \cup \mathcal{B}_2^m \cup \ldots \cup \mathcal{B}_n^m, \quad |\mathcal{B}_i^m| = P \quad \text{for} \quad \forall i.$$

The model is reduced considering only interactions within each batch:

$$\frac{1}{N-1} \sum_{k=1, k\neq i}^{N} a(\mathbf{x}_k - \mathbf{x}_i)(\mathbf{v}_k - \mathbf{v}_i) \quad \rightarrow \quad \frac{1}{P-1} \sum_{k \in [i]_m, k\neq i} a(\mathbf{x}_k - \mathbf{x}_i)(\mathbf{v}_k - \mathbf{v}_i),$$

where $[i]_m$ denotes the batch containing i for $t \in [t_{m-1}, t_m]$,

We then control this reduced dynamics, which leads to a stochastic mini-batch gradient descent method.

Simulations using the RBM

Simulations show that the RBM properly approximates the distribution of evaders (better than the trajectory of individual evaders). The convergence analysis is to be done to a large extent.

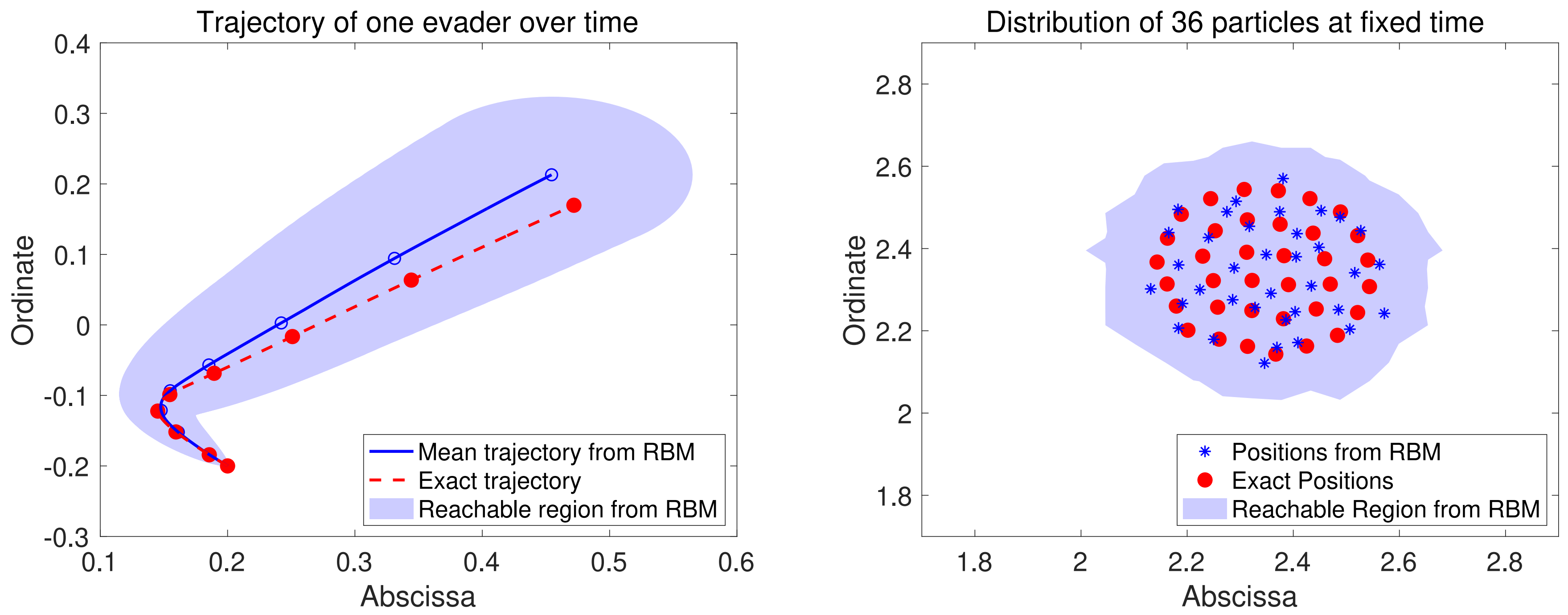


Figure: Simulation along $t \in [0, 4]$ (left) and at $t = 10$ (right). **Red**: positions from original system, **Blue**: positions from RBM, **Colored region**: 95% confidence region with 200 simulations.

An added tool is need to reduce the error in the control of the dynamics, which increases in long time-horizons.

Model Predictive Control (MPC)

MPC adapts the control obtained through the reduced dynamics to the full system in an iterative manner. This is achieved by optimizing a finite

time-horizon, but only implementing the current timeslot and then optimizing again, repeatedly.

MPC leads to a semi-feedback strategy.

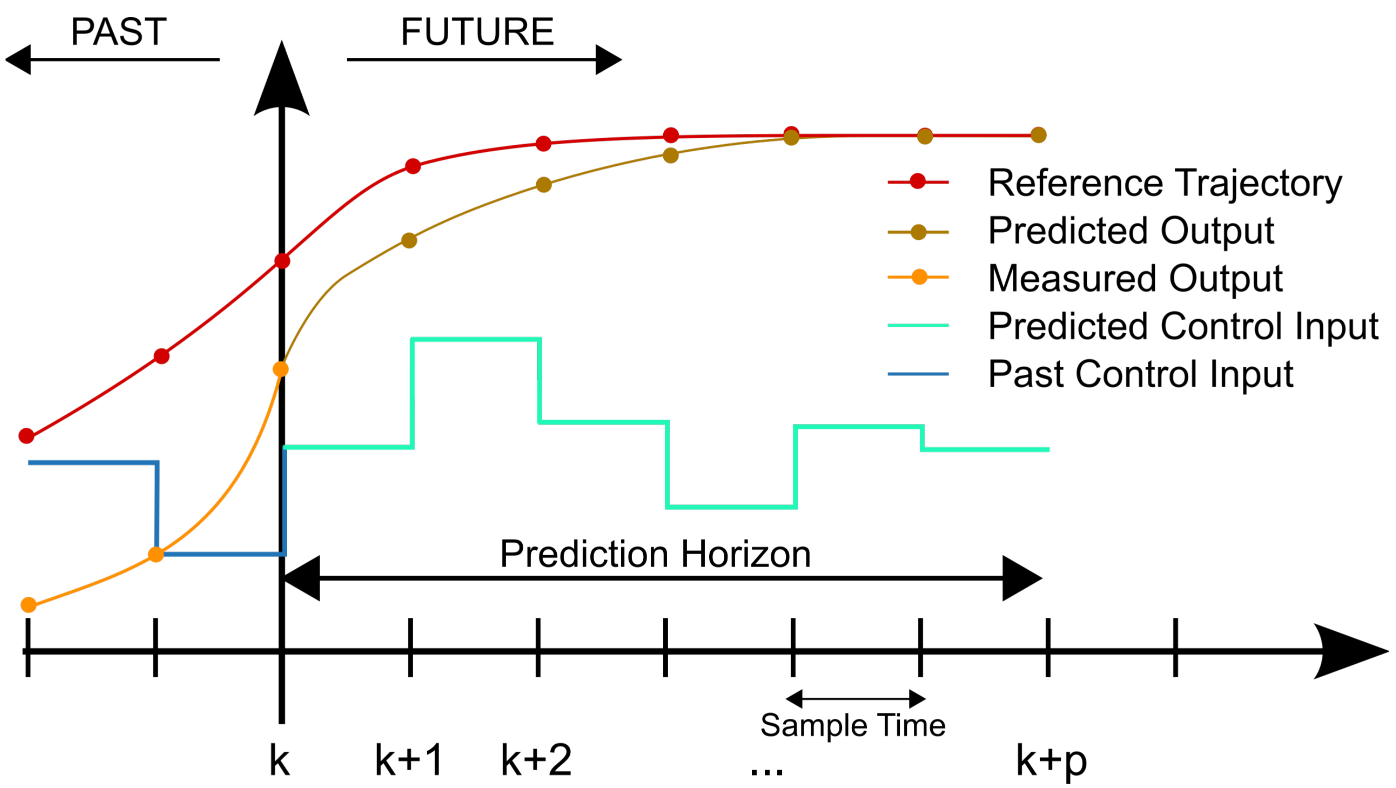
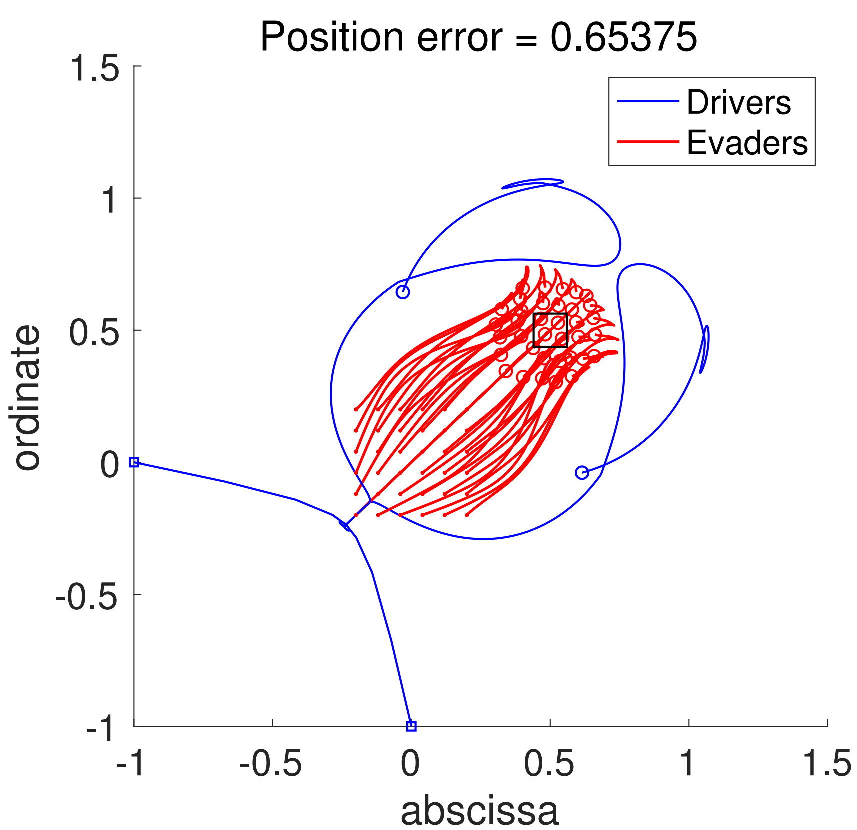
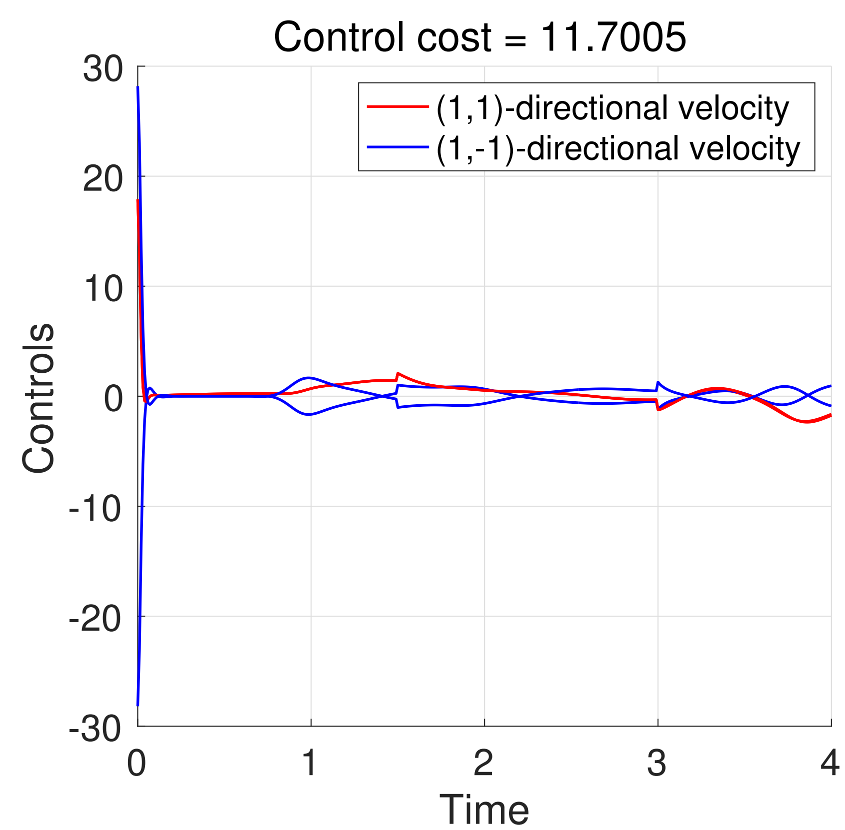


Figure: Iterative control by MPC.

Simulations: MPC + RBM

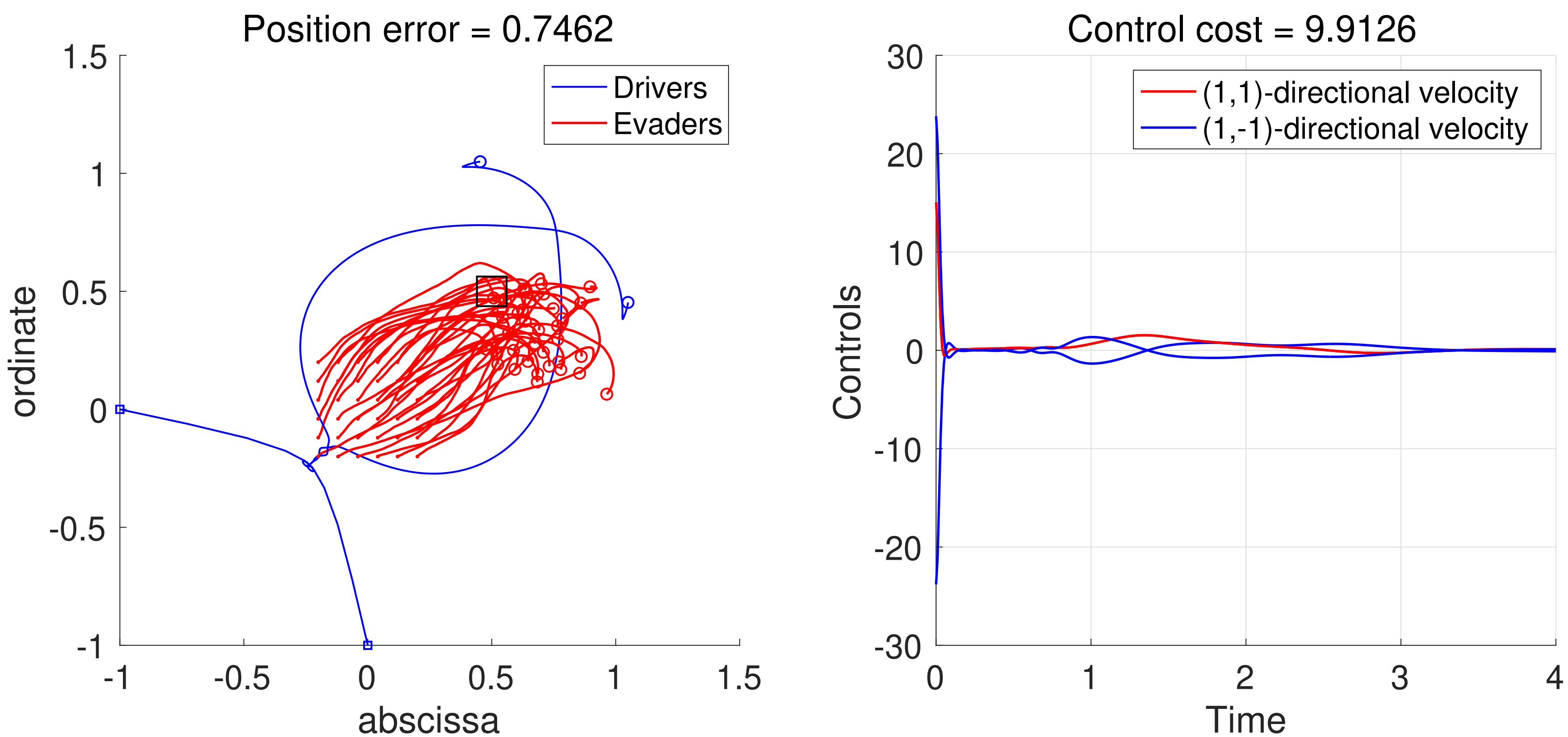
- Results are almost as successful as when controlling the full system, but at a lower computational cost.
- We observe a more complex dynamics of the controllers at the final time. This is due to the anticipative effect that MPC introduces.





Failure of classical opren-loop control strategies in the presence of noise

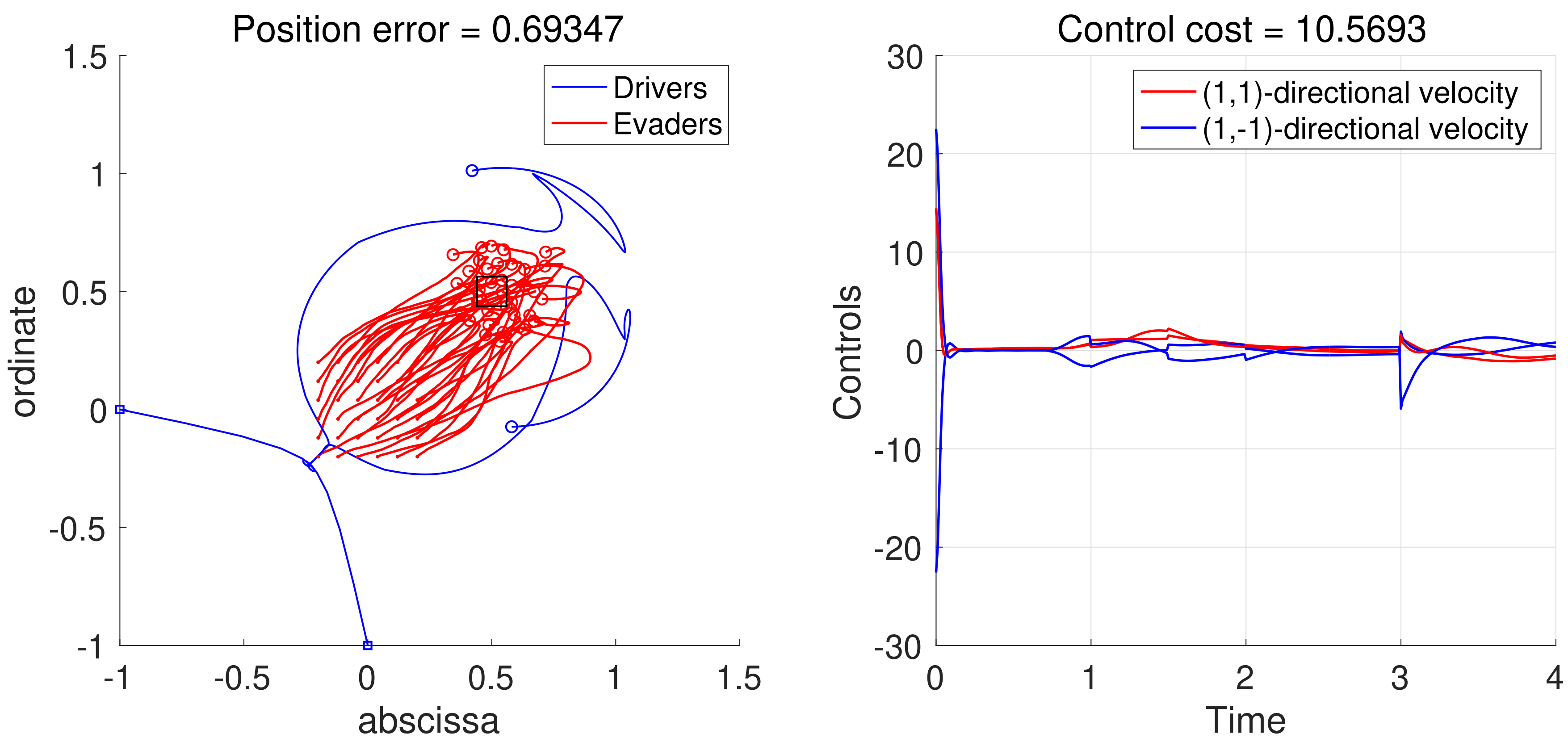
When an unexpected noise perturbs the dynamics of the system, the classical open-loop strategy fails to regulate the system successfully.



The optimal open-loop control is not able to compensate the perturbation introduced by the noise.

The cure of the combined MPC-RBM strategy

The combined MPC-RBM strategy is able to cope with unexpected noisy events.



More drivers are welcome

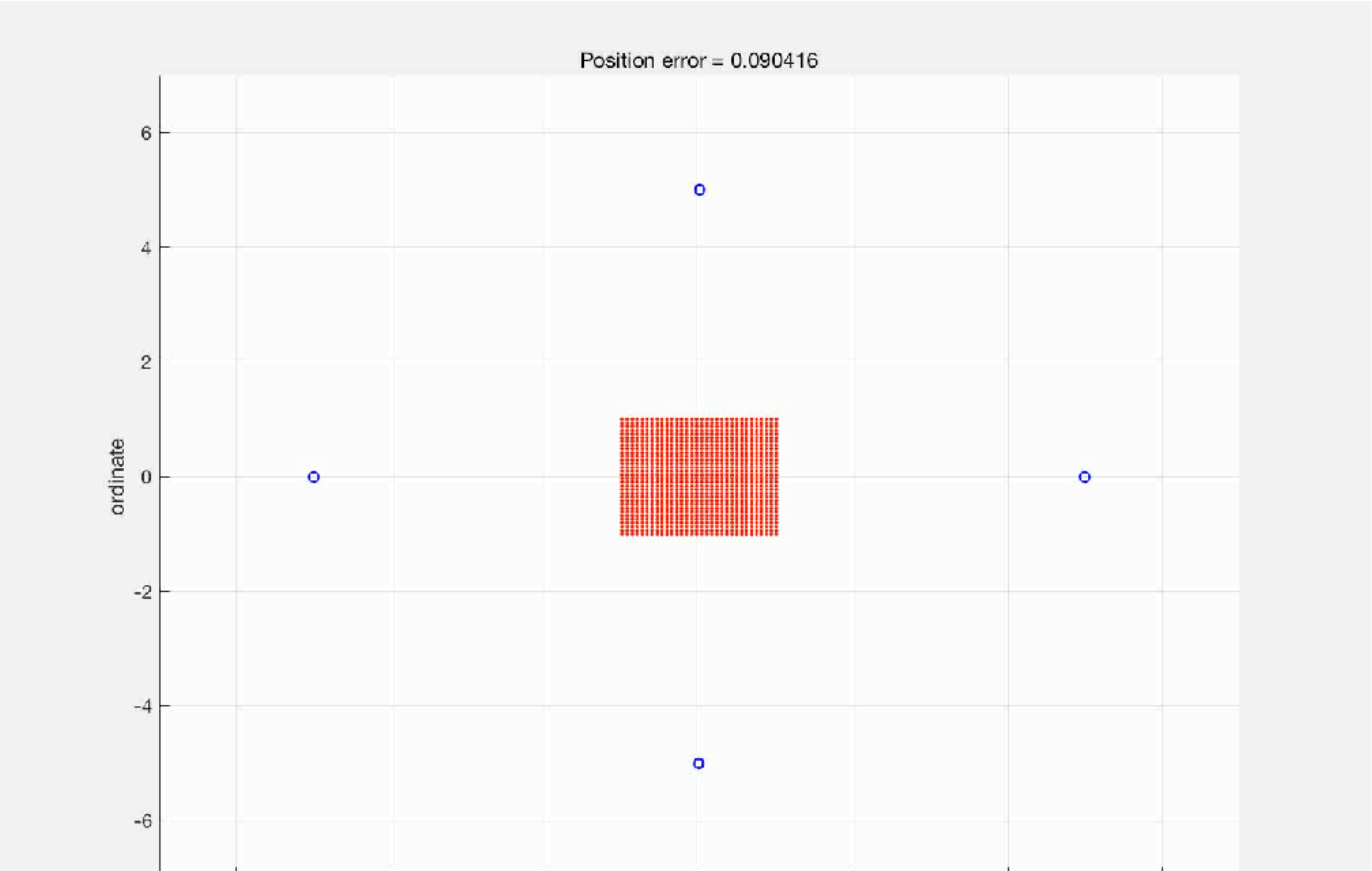


Table of Contents

The RBM in an LQR-problem

We apply the RBM to approximate the minimizer $u^*(t)$ of

$$\min_{u \in L^2(0,T)} J(u) = \int_0^T \left(|x(t) - x_d(t)|^2 + |u(t)|^2 \right) dt, \tag{1}$$

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad x(0) = x_0. \tag{2}$$

Step 1 Decompose the matrix A as

$$A = \sum_{m=1}^{M} A_m. \tag{3}$$

Step 2 Enumerate the 2^M subsets of $\{1, 2, \ldots, M\}$ as $S_1, S_2, \ldots S_{2^M}$. Assign to each subset S_ℓ a probability p_ℓ.

Step 3 Divide $[0, T]$ into subintervals $[t_{k-1}, t_k)$ of length $\leq h$. Randomly choose an index $\ell(k) \in \{1, 2, \ldots, 2^M\}$ in each $[t_{k-1}, t_k)$ according to the probabilities p_ℓ.

The RBM in an LQR-problem

Step 4 Define the matrix $A_h(t)$

$$A_h(t) = \sum_{m \in S_{\ell(k)}} \frac{A_m}{\pi_m}, \qquad t \in [t_{k-1}, t_k), \tag{4}$$

where π_m is the probability that m is an element of the selected subset, i.e.

$$\pi_m = \sum_{\{\ell | m \in S_\ell\}} p_\ell. \tag{5}$$

Step 5 Compute the minimizer $u_h^*(t)$ of the 'simpler' LQR problem

$$\min_{u \in L^2(0,T)} J_h(u) = \int_0^T \left(|x_h(t) - x_d(t)|^2 + |u(t)|^2 \right) dt, \tag{6}$$

$$\dot{x}_h(t) = A_h(t) x_h(t) + Bu(t), \qquad x(0) = x_0. \tag{7}$$

Convergence results

Is it likely that $u_h^*(t)$ is a good approximation of $u^*(t)$?

Theorem

There exists a constant $C > 0$ such that

$$\mathbb{E}[|J_h(u_h^*) - J(u^*)|] \leq Ch, \qquad \mathbb{E}[|J(u_h^*) - J(u^*)|] \leq Ch. \tag{8}$$

By Markov's inequality, also

$$\mathbb{P}[|J(u_h^*) - J(u^*)| > \delta] \leq \frac{Ch}{\delta} \tag{9}$$

Theorem

There exists a constant $C > 0$ such that

$$\mathbb{E}[|u_h^* - u^*|^2_{L^2(0,T)}] \leq Ch. \tag{10}$$

Conclusion: $u_h^*(t)$ is likely a good approximation of $u^*(t)$ when the spacing of the temporal grid h is small enough.

Table of Contents

Summary and perspectives

- The algorithm combines **MPC and RBM**, to compute a reliable control strategy reducing computational cost.
- RBM reduces the computation cost on the forward and adjoint dynamics, from order $O(N^2)$ to $O(NP)$.
- MPC allows to correct the control variations introduced by the RBM.
- In a computational experiment 36 evaders and 2 drivers, the computation cost is reduced to 16%, while the performance of control J differs only about 0.5%.
- Plenty to be done towards a complete rigorous analysis of the convergence of the whole process.

The error analysis of RBM has been developed mainly for contractive systems [Jin, Li, Liu, 2020, JCP], though numerical simulations show good performances [Carrillo, Jin, Li, Zhu, 2019], [Ha, Jin, Kim, 2019].

More complex models

- **From a computational perspective:** Interesting possible extensions for models in non-flat topographies and 3-d models.
- **From the analysis perspective:** Plenty to be done to rigorously analyze the actual controllability properties of these systems. Existing results are limited to the Linear Quadratic Regulator (LQR) model.

Thank you for your kind invitation and attention!