ASE 389: Modeling Multi-Agent Systems – Project Proposal
Covariance Steering Games with Squared Wasserstein Distance Cost

Isin Balci : imb537

Martin Braquet : mb63855

I. Introduction

In this project, we address the problem of steering a discrete-time linear dynamical system from an initial Gaussian distribution to a final distribution in a game-theoretic setting. Stochastic games deal with instances where a stochastic process is jointly controlled by two players. It is assumed that the players have perfect measurement of the state at each time instant and that the initial state is sampled from a given Gaussian distribution. This problem finds applications in motion planning for diverse autonomous systems, for example to control the probabilistic density of a swarm of autonomous agents [1]. It will be solved through three different methods, and extensive simulations results will provide a comparison between them.

More formally we consider a discrete-time system dynamics:

$$x_{k+1} = A_k x_k + B_k u_k + D_k v_k + w_k \tag{1}$$

where $x_0 \sim \mathcal{N}(\mathbf{0}, \Sigma_0)$, $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$ is the action of player-1 and $v_k \in \mathbb{R}^{n_v}$ is the action of player-2, $\mathbb{R}^{n_w} \ni w_k \sim \mathcal{N}(\mathbf{0}, \Sigma_w)$ is the i.i.d. noise process.

Each player has a distinct desired terminal state distribution which are Gaussians with mean $\mu_1, \mu_2 \in \mathbb{R}^n$ and covariance $\Sigma_1, \Sigma_2 \in \mathbb{S}_n^+$.

Players try to achieve their objective by minimizing their respective cost functions which are shown below:

$$J_1(\boldsymbol{u}, \boldsymbol{v}) = \mathbb{E}\left[\sum_{k=0}^{N-1} u_k^{\mathrm{T}} R_k^u u_k\right] + W_2^2(\rho_N, \rho_d^u), \tag{2a}$$

$$J_2(\boldsymbol{u}, \boldsymbol{v}) = \mathbb{E}\left[\sum_{k=0}^{N-1} v_k^{\mathrm{T}} R_k^v v_k\right] + W_2^2(\rho_N, \rho_d^v), \tag{2b}$$

where $W_2^2(\rho_1, \rho_2)$ denotes the squared Wasserstein distance between Gaussian distributions ρ_1 and ρ_2, $\boldsymbol{u} = [u_0^{\mathrm{T}}, \dots, u_{N-1}^{\mathrm{T}}]^{\mathrm{T}}$ and $\boldsymbol{v} = [v_0^{\mathrm{T}}, \dots, v_{N-1}^{\mathrm{T}}]^{\mathrm{T}}$. Note that, both cost functions J_1 and J_2 depend on both players actions since ρ_N represents the distribution of the terminal state x_N.

Even though evaluating the squared Wasserstein distance between two arbitrary distributions does not admit a closed form solution, the squared Wasserstein distance between two Gaussian distributions ($\rho_i = \mathcal{N}(\mu_i, \Sigma_i)$ where $i \in \{1, 2\}$) has a closed form solution:

$$\begin{aligned} W_2^2(\rho_1, \rho_2) = \|\mu_1 - \mu_2\|_2^2 + \mathrm{tr}(\Sigma_1 + \Sigma_2) \\ - 2\,\mathrm{tr}\left((\Sigma_2^{1/2}\Sigma_1\Sigma_2^{1/2})^{1/2}\right). \end{aligned} \tag{3}$$

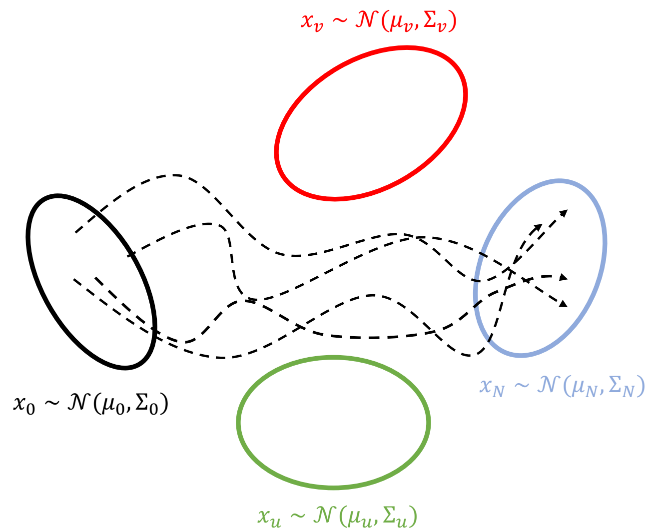

Fig. 1. Black, blue, red, and green ellipses show the confidence region of the initial, terminal and desired distributions of player 1 and player 2 respectively. The dashed lines show the state trajectory samples.

Furthermore, using (1), we can express the concatenated state, input and the noise vectors as follows:

$$\boldsymbol{x} = \mathbf{G}_0 x_0 + \mathbf{G_u}\boldsymbol{u} + \mathbf{G_v}\boldsymbol{v} + \mathbf{G_w}\boldsymbol{w} \tag{4}$$

where $\boldsymbol{x} = [x_0^{\mathrm{T}}, x_1^{\mathrm{T}}, \dots, x_N^{\mathrm{T}}]^{\mathrm{T}}$, and $\boldsymbol{w} = [w_0^{\mathrm{T}}, w_1^{\mathrm{T}}, \dots, w_{N-1}^{\mathrm{T}}]^{\mathrm{T}}$. The reader can refer to [2], [3] for the derivations of the matrices $\mathbf{G}_0, \mathbf{G_u}, \mathbf{G_v}, \mathbf{G_w}$. Under the assumption that the final state has a Gaussian distribution, the cost functions J_1 and J_2 can be expressed in terms of $\mathbb{E}[\boldsymbol{u}]$, $\mathbb{E}[\boldsymbol{v}]$, $\mathrm{Cov}(\boldsymbol{u})$ and $\mathrm{Cov}(\boldsymbol{v})$ where $\mathbb{E}[\cdot]$ and $\mathrm{Cov}(\cdot)$ denotes expected value and covariance of a random vector.

II. Specific Aims

To solve the problem described in Introduction, we propose to use three different methods which are gradient based learning of Nash Equilibrium [4], iterative best response algorithm [5] and iterative linearization of the squared Wasserstein distance to approximate the game as linear quadratic game as in [6].

Gradient Based Learning Approach: Suppose both players use affine state feedback policy:

$$u_k = \bar{u}_k + K_k(x_k - \bar{x}_k) \tag{5a}$$

$$v_k = \bar{v}_k + H_k(x_k - \bar{x}_k) \tag{5b}$$

Then, we can write them more compactly as follows:

$$\boldsymbol{u} = \bar{\boldsymbol{u}} + \boldsymbol{K}(\boldsymbol{x} - \bar{\boldsymbol{x}}) \tag{6a}$$

$$\boldsymbol{v} = \bar{\boldsymbol{v}} + \boldsymbol{H}(\boldsymbol{x} - \bar{\boldsymbol{x}}) \tag{6b}$$

where $\boldsymbol{K} = \mathrm{bdiag}(K_0, \ldots, K_{N-1}, \boldsymbol{0})$ and $\boldsymbol{H} = \mathrm{bdiag}(H_0, \ldots, H_{N-1}, \boldsymbol{0})$ So, the policy and the cost function J_i of each player is parametrized by a finite number of decision variables which are $\bar{\boldsymbol{u}}, \bar{\boldsymbol{v}}, \boldsymbol{K}$ and $\boldsymbol{H}$. The derivatives of J_i with respect to decision variables can be computed via standard matrix calculus operations [7]. Thus, the gradient descent methods described in [4] can be used to compute Nash Equilibrium.

Iterative Best Response: If the policy of one player is fixed, then the problem becomes an instant of the problem solved in [2] which is shown to be a difference of convex functions program and can be solved via convex concave procedure (CCP) [8]. This allows us to use the CCP to solve the game iteratively to find the best response of each player.

Iterative LQG Game: The control effort terms in cost functions J_i are already quadratic in control inputs of each agent u_k and v_k. Also, the first term of the squared Wasserstein distance (3) is quadratic in the terminal state mean $\mathbb{E}[x_N]$ and the second term is linear in terminal state covariance $\mathrm{Cov}(x_N)$. By linearizing the third term of (3) around some $\Sigma_N^0 \in \mathbb{S}_{n_x}^+$ and writing the terminal state covariance as $\mathrm{Cov}(x_n) = \mathbb{E}[(x_n - \bar{x}_n)(x_n - \bar{x}_n)^{\mathrm{T}}]$ we can turn the cost function into a quadratic form. Thus, the linearized problem can be recast as a LQ game in which the closed form solution can be found by dynamic programming. Since the optimal policies of each player in LQ games are in the form of (5a), the new terminal covariance can be computed using these policies. So, this procedure gives us an iterative method to compute the policies that satisfy the Nash Equilibrium.

The main objective of this project is to explore the solution methods of the Covariance Steering Games with Wasserstein distance cost function. The methods proposed will be evaluated experimentally with different parameters and the convergence properties of each method will be observed. If convergence occurs, we will analyze whether the algorithms converge to a Nash Equilibrium. If we observe issues with convergence, we will try to remedy the issue with minor modifications in the algorithms that we use. Finally, we will compare three methods that we mentioned in terms of convergence rate.

III. Literature Review

This project is based on optimal covariance steering with squared Wasserstein distance terminal cost [2]. Given the initial probability distribution of a single agent, this paper develops an algorithm to steer the probability distribution of the terminal state of the system close to a desired Gaussian distribution. The closeness between the terminal state distribution and the desired (goal) distribution is measured in terms of the squared Wasserstein distance. The authors recast the stochastic optimal control problem as a finite-dimensional nonlinear program whose performance index can be expressed as the difference of two convex functions. This representation of the performance index allows them to find local minimizers of the original nonlinear program via the so-called convex-concave procedure. Our project will extend this paper by incorporating a game between agents who aim to steer the system to two different final probability distributions.

A similar paper deals with a finite-horizon covariance control problem for discrete-time, stochastic linear systems with complete state information subject to input constraints [3]. The boundary condition on the final state guarantees that the terminal state covariance is less than the desired covariance. The authors recast the covariance control problem, into semi-definite program which is then solved via specialized solvers.

Another paper handles the case of continuous state dynamics [9], where the authors aim to steer the state covariance to a desired value while minimizing the total energy spent. The optimum policy is found to be in the form of linear time-varying state feedback which is computed through solving a pair of Lyapunov equations which are couple through their boundary values.

Covariance steering for game has already been developed in different environments. For example in [10], they steer a discrete-time linear dynamical system from an initial Gaussian distribution to a final distribution in a game-theoretic setting. One of the two players strives to minimize a quadratic payoff, while at the same time tries to meet a given mean and covariance constraint at the final time-step. The other player maximizes the same payoff (to maximize the total input effort of the other player), but it is assumed to be indifferent to the terminal constraint.

In [11], the authors consider a linear quadratic stochastic game as the main setup. However, the goal is to design the terminal state cost which is different than our objective. The authors assume that after the terminal cost is picked the agents follow the Nash policies. So the main objective is to design a terminal state cost so that terminal covariance is equal to the desired one. Even though the authors provide coupled lyapunov equations which can provide optimal solution, they do not provide a numerical procedure to solve these equations.

References

[1] N. Demir, U. Eren, and B. Açıkmeşe, "Decentralized probabilistic density control of autonomous swarms with safety constraints," *Autonomous Robots*, vol. 39, no. 4, pp. 537–554, 2015.

[2] I. M. Balci and E. Bakolas, "Covariance steering of discrete-time stochastic linear systems based on wasserstein distance terminal cost," *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 2000–2005, 2020.

[3] E. Bakolas, "Finite-horizon covariance control for discrete-time stochastic linear systems subject to input constraints," *Automatica*, vol. 91, pp. 61–68, 2018.

[4] E. Mazumdar, L. J. Ratliff, and S. S. Sastry, "On gradient-based learning in continuous games," *SIAM Journal on Mathematics of Data Science*, vol. 2, no. 1, pp. 103–131, 2020.

[5] D. Fudenberg, F. Drew, D. K. Levine, and D. K. Levine, *The theory of learning in games*, vol. 2. MIT press, 1998.

[6] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1475–1481, IEEE, 2020.

[7] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on circuits and systems*, vol. 25, no. 9, pp. 772–781, 1978.
[8] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural computation*, vol. 15, no. 4, pp. 915–936, 2003.
[9] Y. Chen, T. T. Georgiou, and M. Pavon, "Optimal steering of a linear stochastic system to a final probability distribution, part i," *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1158–1169, 2015.
[10] V. R. Makkapati, T. Rajpurohit, K. Okamoto, and P. Tsiotras, "Covariance steering for discrete-time linear-quadratic stochastic dynamic games," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 1771–1776, IEEE, 2020.
[11] Y. Chen, T. T. Georgiou, and M. Pavon, "Covariance steering in zero-sum linear-quadratic two-player differential games," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 8204–8209, IEEE, 2019.