Hierarchical Control of Multi-Agent Systems using Online Reinforcement Learning

He Bai, Jemin George and Aranya Chakrabortty

Abstract—We propose a new reinforcement learning based approach to designing hierarchical linear quadratic regulator (LQR) controllers for heterogeneous linear multi-agent systems with unknown state-space models and separated control objectives. The separation arises from grouping the agents into multiple non-overlapping groups, and defining the control goal as two distinct objectives. The first objective aims to minimize a group-wise block-decentralized LQR function that models group-level mission. The second objective, on the other hand, tries to minimize an LQR function between the average states (centroids) of the groups. Exploiting this separation, we redefine the weighting matrices of the LQR functions in a way that they allow us to decouple their respective algebraic Riccati equations. Thereafter, we develop a reinforcement learning strategy that uses online measurements of the agent states and the average states to learn the respective controllers based on the approximate Riccati equations. Since the first controller is block-decentralized and, therefore, can be learned in parallel, while the second controller is reduced-dimensional due to averaging, the overall design enjoys a significantly reduced learning time compared to centralized reinforcement learning.

I. INTRODUCTION

Conventional reinforcement learning (RL) based control of high-dimensional LTI systems with unknown state-space models using algorithms such as actor-critic methods [1], Q-learning [2], and adaptive dynamic programming (ADP) [3] usually have long learning times. This is because the initialization of these learning algorithms involves a least squares estimation step that requires RL to wait until a minimum amount of time for gathering sufficient amount of state and input data so that the appropriate data matrices can be guaranteed to have full rank. Larger the size of the plant, more is this waiting time.

In this paper we show that reduction in learning time is possible in scenarios where the control objective can be decomposed into a hierarchy of sub-objectives. We consider a large number of self-actuated agents with decoupled open-loop heterogeneous LTI dynamics. The state and input matrices of each agent are assumed to be unknown, while all states and control inputs are assumed to be measured. The agents are assumed to be divided into multiple non-overlapping groups that may arise from various factors such as their geographical proximity, the nature of their mission, or the physical characteristics of the agents. The grouping is imposed only to make the separation in the control objectives well-defined; it does not imply any kind of time-scale separation or redundancy in controllability as in [4]. The goal is to learn two distinct sets of LQR controllers. The first controller is a local LQR controller for each individual group that employs feedback of only the agent states belonging to that group. The overall local control gain matrix is thus block-diagonal in structure. The second controller, on the other hand, is a global LQR controller that is meant to control the relative motion between the centroids of each group. It is, therefore, a reduced-dimensional controller that employs feedback from only the average states of each group.

A model-based version of this hierarchical LQR control for homogeneous LTI models was recently reported in [5], followed by other optimization based designs in [6]–[8]. Motivated by the technique presented in [5], we first redefine the input weighting matrices of the LQR functions in a way that they allow us to decouple their respective algebraic Riccati equations (AREs). Our approach is different from [5] where instead the state weighting matrix was redefined, leading to a different set of Riccati equations. Since the first controller is block-decentralized and, therefore, can be learned in parallel, while the second controller is reduced-dimensional due to averaging, the overall design enjoys a significantly reduced learning time compared to centralized reinforcement learning. We illustrate the effectiveness of the design and also highlight its drawbacks on sub-optimality using an example from hierarchical formation control.

The rest of the paper is organized as follows. Section II formulates the hierarchical LQR control problem, and provides its model-based solution using approximations in the AREs. Section III develops a variant of ADP that learns the hierarchical controllers using state and input measurements. Section IV shows the applicability of this design for multi-agent formation control, illustrated with numerical simulations. Section V concludes the paper.

II. PROBLEM FORMULATION

Consider a multi-agent network consisting of \( N \) agents. For \( j = 1, \ldots, N \), the \( j^{th} \) group contains \( p_j \) number of agents, with any agent \( i \) satisfying the dynamics

\[
\dot{x}_i = G_i x_i + H_i u_i, \tag{1}
\]

where \( x_i \in \mathbb{R}^n \) is the state, and \( u_i \in \mathbb{R}^m \) is the control input of the agent, for all \( i = 1, \ldots, p \). \( p = \sum_j p_j \). The matrices \( G_i \) and \( H_i \) are unknown, although their dimensions...
are known. The agents are assumed to be initially uncoupled from each other. Let \( x_j \) and \( u_j \) represent the vector of all states and control inputs in group \( j \). The group-level dynamics are written as

\[
\dot{x}_j = A_j x_j + B_j u_j,
\]

where \( A_j \) and \( B_j \) are block-diagonal concatenations of \( G_i \) and \( H_i \), respectively, for all agents \( i \) belonging to group \( j \). Denoting \( x = [x_1^\top \ldots x_N^\top]^\top \in \mathbb{R}^{pn} \), \( u = [u_1^\top \ldots u_N^\top]^\top \in \mathbb{R}^{pn} \), the network model becomes

\[
\dot{x} = Ax + Bu,
\]

where \( A \in \mathbb{R}^{np \times np} \) and \( B \in \mathbb{R}^{np \times mp} \) are block-diagonal matrices consisting of \( A_j \)'s and \( B_j \)'s, respectively.

Let the control objective be to design a state-feedback controller \( u = -Kx \) to minimize the cost

\[
J = \int_0^\infty x^\top Q x + u^\top R u \, dt,
\]

where \( Q \geq 0 \) and \( R > 0 \) are performance matrices of appropriate dimensions, constrained to (3).

**Assumption 1.** The communication topology among the centroids of the \( N \) groups is an undirected network with a Laplacian matrix \( L \in \mathbb{R}^{N \times N} \).

**Assumption 2.** Performance matrices \( Q \) and \( R \) are given as

\[
R = \text{diag}(R_1, \ldots, R_N), \quad Q = \tilde{Q} + L_w \odot \tilde{Q},
\]

where \( R_j \in \mathbb{R}^{np_j \times np_j} > 0 \), \( \tilde{Q} \in \mathbb{R}^{np \times np} \geq 0 \) for all \( j = 1, \ldots, N \), \( L_w \) is a weighted Laplacian matrix which has the same structure as \( L \), and

\[
\tilde{Q} = \begin{bmatrix}
\frac{1}{p_1} & \frac{1}{p_1} & \ldots & \frac{1}{p_1} \\
\frac{1}{p_2} & \frac{1}{p_2} & \ldots & \frac{1}{p_2} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{1}{p_n} & \frac{1}{p_n} & \ldots & \frac{1}{p_n}
\end{bmatrix}
\otimes I_n.
\]

It can be seen that \( \tilde{Q} \in \mathbb{R}^{np \times np} \geq 0 \). Here, \( \otimes \) is the Kronecker product and \( \odot \) is the Khatri–Rao product.

The two components of the matrix \( Q \) in (5) represent the separation in the control objective. The diagonal component \( \tilde{Q} \) represents group-level local objective, such as maintaining a desired formation for each group in the multi-agent network. The second component, on the other hand, represents a global objective that is meant to coordinate a set of compressed state vectors chosen from each group. In this case, as indicated in (7) we assume the compressed state to be simply the centroid, i.e., the average of the respective group states. More general definitions of this compressed state is possible, but we stick to this assumption for simplicity. Denote the centroid state of the \( j^{th} \) group by \( x_{av,j} \in \mathbb{R}^n \), which is the average of the state vectors of all the agents in that group. Let the quadratic objective for the \( j^{th} \) group be

\[
J_{av,j} = \sum_{\ell \in N_j} (x_{av,j} - x_{av,\ell})^\top Q_{j\ell} (x_{av,j} - x_{av,\ell}),
\]

where \( N_j \) is the neighbor set of the \( j^{th} \) centroid following the structure of the Laplacian matrix \( L \), and \( Q_{j\ell} \geq 0 \) is a given \( n \times n \) design matrix.

Since the network is assumed to be undirected, the Laplacian matrix can be written as \( L = DD^\top \), where \( D \) is the incidence matrix. Define the weighted Laplacian as

\[
L_w = (D \otimes I_n)Q(D^\top \otimes I_n),
\]

where \( Q \) is a block-diagonal matrix with \( Q_{j\ell} \)'s as diagonal entries. Equation (8) for the entire network can be written as

\[
J_{av} = \sum_{j} J_{av,j} = x_{av}^\top L_w x_{av},
\]

where \( x_{av} := [x_{av,1}^\top \ldots x_{av,N}^\top]^\top \in \mathbb{R}^{nN} \). Also, since

\[
x_{av} := \text{diag} \left\{ \frac{1}{p_1}, \frac{1}{p_2}, \ldots, \frac{1}{p_N} \right\} \otimes I_n \in \mathbb{R}^{nN}
\]

(10) can be further written as

\[
J_{av} = x^\top (M \otimes I_n) L_w (M \otimes I_n) x,
\]

which justifies the definition of \( \tilde{Q} = M^\top M \) in (7).

**Assumption 3.** \( (A, B) \) is controllable and \( (Q^{1/2}, A) \) is observable.

The optimal control input for minimizing (4) is

\[
u = -K^* x = -R^{-1} B^\top P^* ,
\]

where \( P^* \in \mathbb{R}^{np \times np} \) is the unique positive definite solution of the following Riccati equation:

\[
P^* A + A^\top P^* + Q - P^* B R^{-1} B^\top P^* = 0.
\]

As \( (A, B) \) are unknown, (14) cannot be solved directly. Instead it can be solved via RL using measured values of \( x \) and \( u \). One may disregard the separation property in (5), and solve for \( P^* \) from (14) using the centralized RL algorithm in [1], but the drawback in that case will be a long learning time owing to the large dimension of \( P^* \). The benefit will be that \( P^* \) is optimal. Our approach, in contrast, is to make use of the separation property in (5) to learn and implement an RL controller using two separate and parallel components, thereby reducing the learning time. In fact, as will be shown next, the learning phase in this case reduces to learning only the individual group-level local controllers. Once learned, these local controllers can be used to compute the global component of \( u \) through a simple matrix product. The drawback is that the learned \( u \) is no longer optimal. We next describe the construction of this sub-optimal control input using an approximation for \( R \).
A. Approximate Control

Define \( P = \text{diag}\{P_1, \ldots, P_N\} \), where \( P_j \in \mathbb{R}^{n_{pj} \times n_{pj}} \) are symmetric positive-definite matrices. Define \( P_{Aj} = P_j A_j \) and \( P_{Bj} = P_j B_j R_j^{-1} B_j^T P_j \) for \( j = 1, \ldots, N \). Following [5], we also define

\[
\mathcal{R}^{-1} = R^{-1} + \tilde{R},
\]

where the expression for \( \tilde{R} \) will be derived shortly. Then,

\[
\mathcal{P} A + A^T \mathcal{P} + Q - \mathcal{P} B R^{-1} B^T \mathcal{P} = \mathcal{P} A + A^T \mathcal{P} + Q - \mathcal{P} B R^{-1} B^T \mathcal{P} + L_w \odot \tilde{Q} - \mathcal{P} B \tilde{R} B^T \mathcal{P} = \text{diag}\{P_{A1} + P_{A1}^T + \tilde{Q} - P_{B1} B_1^{-1} B_1^T P_{A1}, \ldots, P_{AN} + P_{AN}^T + \tilde{Q} - P_{BN} B_N^{-1} B_N^T P_{AN} \} + L_w \odot \tilde{Q} - \mathcal{P} B \tilde{R} B^T \mathcal{P}.
\]

Compared to the approximation suggested in [5], we propose to fix \( \tilde{Q} \) and instead adjust \( \tilde{R} \) to account for the coupled terms in \( Q \). Adjusting \( \tilde{R} \) is more amenable for this design since perturbing \( Q \) will severely degrade the system performance while adjusting \( \tilde{R} \) will only increase (or decrease) the control demand. Furthermore, considering the structure of the RHS of (16), it is easier to choose \( \tilde{R} \) to cancel out the coupling term \( L_w \odot \tilde{Q} \) than choosing \( Q \). Thus, if \( \tilde{R} \) is selected so that

\[
\mathcal{P} B \tilde{R} B^T \mathcal{P} = L_w \odot \tilde{Q}, \tag{17}
\]

then each individual matrix \( P_j \) satisfies

\[
P_j A_j + A_j^T P_j + \tilde{Q}_j - P_j B_j R_j^{-1} B_j^T P_j = 0, \tag{18}
\]

for \( j = 1, \ldots, N \). The control gain follows as

\[
K = \mathcal{R}^{-1} B^T \mathcal{P} = \mathcal{R}_0^{-1} B^T \mathcal{P} + \tilde{R}_g B^T \mathcal{P} \tag{19}
\]

Note that the global component does not need to be learned. Once \( \mathcal{P} \) is learned from (18) the global controller can simply be computed using \( \mathcal{P} \), \( \tilde{R} \) and \( B \). As \( B^T \mathcal{P} \) is block diagonal, the structure in \( \tilde{R} \) dictates the structure of the global control.

The problem, however, lies in the fact that it may be difficult to find a \( \tilde{R} \) which satisfies (17). If \( B \) is a square full rank matrix (i.e., each agent is a fully actuated system) then \( \tilde{R} \) follows in a straightforward way by computing the inverse of the square matrices \( \mathcal{P} B \) and \( B^T \mathcal{P} \). Otherwise, one has to compute a least square estimate for \( \tilde{R} \) as

\[
\tilde{R}^* = (B^T \mathcal{P} B)^{-1} B^T \mathcal{P} (L_w \odot \tilde{Q}) \mathcal{P} B (B^T \mathcal{P} B)^{-1} \tag{20}
\]

Since \( \mathcal{P} B \) is block-diagonal, we can write it as

\[
\mathcal{P} B = I_N \odot \text{diag}\{P_1 B_1, \ldots, P_N B_N\}. \tag{21}
\]

The matrix \( (B^T \mathcal{P} B)^{-1} \) can be written in a similar fashion. In that case, it follows that

\[
\tilde{R}^* = (B^T \mathcal{P} B)^{-1} B^T \mathcal{P} (I_N \odot \text{diag}\{\tilde{Q}\}) \mathcal{P} B (B^T \mathcal{P} B)^{-1}
\]

\[
= (I_N \odot \text{diag}\{(B_j^T P_j B_j)^{-1}\})(I_N \odot \text{diag}\{(B_j^T P_j B_j)^{-1}\}) (L_w \odot \tilde{Q}) (I_N \odot \text{diag}\{P_j B_j\}) (I_N \odot \text{diag}\{(B_j^T P_j B_j)^{-1}\})
\]

\[
= L_w \odot \tilde{Q}' \tag{22}
\]

where the expression for \( \tilde{Q}' \) is shown in (23). \( \tilde{Q}' \) is close to \( \tilde{Q} \) in the least square sense. The drawback of the approximate controller (19), therefore, is that instead of minimizing the original objective function (4), it minimizes

\[
J = \int_0^\infty x^T Q' x + u^T R u \, dt, \tag{24}
\]

where \( Q' = Q + L_w \odot \tilde{Q}' \), and \( R \) follows from (15).

Also, because \( \mathcal{P} B \) and \( B^T \mathcal{P} \) are block diagonal, they only represent different scalings of the agent states in (20). Thus, any communication structure imposed in \( \tilde{R}^* \) and \( \tilde{R}^* B^T \mathcal{P} \), which is the global control gain. To implement \( \tilde{R}^* \), neighboring agents need to share their \( \mathcal{P} B \) vectors. The loss from the optimal objective function (4) to the approximated objective function (24) (or equivalently the loss from \( Q \) to \( Q' \)) can be numerically shown to become smaller as the number of agents increases. Thus, the true benefit of the design is when the number of agents is large. Theorem 4.1 in [9] can be used to exactly quantify the loss in \( J \) in terms of the difference between \( \tilde{Q} \) and \( \tilde{Q}' \). We skip that derivation, and refer the interested reader to this theorem.

III. Controller Design Using RL

Equation (18) indicates that each local controller can be learned independently using measurements of the local group-level states. The global controller, on the other hand, does not need to be learned owing to the block-diagonal structure of \( A \) and \( B \). Once the local controllers are learned, the global controller can be simply computed as the second component on the right hand side of (19). Algorithm 1 lists the detailed steps for learning the local RL controllers using ADP based on the approximate LQR in (24). An important point to note is that the group-level state matrix \( A_j \) in (2) for formulating the LQR problem is assumed to be block-diagonal. However, since ADP is a model-free design, Algorithm 1 is applicable even if \( A_j \) is not block-diagonal. We will encounter this scenario in our target-tracking example in Section IV, where we will show that Algorithm 1 still successfully learns the desired model-free controller within a short learning time. From (3) it follows that if the exploration noise \( u_{0i}(t) \) is persistently exciting, then \( K_{\text{local}} \) (and \( K_{\text{global}} \) computed from it) in Algorithm 1 will asymptotically converge to the respective solutions of the modified LQR problem (24).

IV. Application to Formation Control

We next demonstrate how to make use of the proposed hierarchical learning algorithm for formation control and target tracking applications in multi-agent systems. We first show how this problem can be posed in terms of the optimal control formulation in (4), and then demonstrate the performance of the learning algorithm using a simulation example.

A. Problem formulation

We consider \( p \) robots, whose dynamics are given by

\[
m_i \ddot{q}_i + c_i \dot{q}_i = u_i, \quad i = 1, \ldots, p, \tag{25}
\]
where $q_i \in \mathbb{R}^2$ denotes the 2D position of robot $i$, $m_i \in \mathbb{R}_+$ is the mass of agent $i$, $c_i \in \mathbb{R}_+$ is a damping coefficient that models friction and drag effects, and $u_i \in \mathbb{R}^2$ is the force that acts as a control input. Denote $x_i = [q_i^T \dot{q}_i^T]^T$. We have
\begin{align}
\dot{x}_i &= G_i x_i + H_i u_i, \\
G_i &= \begin{pmatrix} 0_2 & I_2 \\ 0_2 & -c_i I_2 \end{pmatrix}, \\
H_i &= \begin{pmatrix} 0_2 \\ -m_i I_2 \end{pmatrix}.
\end{align}
(26)
(27)
We assume that $c_i$ and $m_i$ are unknown parameters.

The robots are divided into $N$ groups to track $N$ different targets. Each group has $p_j$ robots. The state of $i$th agent within group $j$ is denoted by $x_j^i$, $i = 1, \ldots, p_j$. We assume that the locations of the targets, $q_j^T(t)$, $j = 1, \ldots, N$, are known and that target assignment is completed so that each group has the knowledge of its assigned target.

The control objective is to ensure that each group converges to a desired formation of its assigned target at the center of the formation while keeping the groups as close as possible, e.g., to maintain a connected communication network. Specifically, for the formation control objective, we choose a reference agent, say agent 1 in group $j$, and require
\[ |q_j^i - q_j^1 - q_j^{i,d}| \to 0, \quad \forall i \in \{2, \ldots, p_j\}, \]
(28)
for some predesigned $q_j^{i,d}$. For the target tracking objective, we require
\[ \sum_{i=1}^{p_j} \frac{1}{N} |q_j^i - q_j^T| \to 0, \quad \forall j. \]
(29)
To keep the groups close, we choose to minimize the distance between the centroids of the groups.

We next formulate these objectives as the optimal control problem (4). Towards this end, we rewrite the agent dynamics within a group as
\[ \dot{x}_j = A_j x_j + B_j u_j \]
(30)
where $x_j = \begin{pmatrix} (x_j^1)^T & \cdots & (x_j^{p_j})^T \end{pmatrix}^T$, $A_j = \text{diag}\{A_j^1, \ldots, A_j^{p_j}\}$, $B_j = \text{diag}\{B_j^1, \ldots, B_j^{p_j}\}$, and $u_j = \begin{pmatrix} (u_j^1)^T & \cdots & (u_j^{p_j})^T \end{pmatrix}^T$.

Given the dynamics of $x_j$, we consider a coordinate transformation $T$ such as
\[ z_j := T x_j = \begin{pmatrix} x_j^2 - x_j^1 \\ x_j^3 - x_j^1 \\ \vdots \\ x_j^{p_j} - x_j^1 \\ \frac{1}{p_j} \sum_{i=1}^{p_j} x_j^i \end{pmatrix} \begin{pmatrix} z_j^1 \\ z_j^2 \\ \vdots \\ z_j^{p_j} \\ \bar{z}_j \end{pmatrix}. \]
(31)
Then the dynamics of $z_j$ is given by
\[ \dot{z}_j = T A_j T^{-1} z_j + T B_j u_j. \]
(32)

**Algorithm 1** Off-policy ADP for Hierarchical Controller (19)

**Step 1 - Data storage:** Each group $i = 1, \ldots, N$ is assigned a coordinator, say denoted as $C_i$, that stores $x_i(t)$ and exploration noise $u_{i0}(t)$ for an interval $(t_1, t_2, \ldots, t_l)$, with sampling time $T$. Total data storage time is $T \times N$ number of learning steps. Assume that there exists a sufficiently large number of sampling intervals for each control iteration step such that $\text{rank}(I_{x_i}, I_{x_i} u_{i0}) = n(n+1)/2 + nm$. This rank condition makes sure that the system is persistently excited. Coordinator $C_i$ constructs the following matrices:
\[ \delta_{x_i} = [x_i \otimes \cdots \otimes x_i]_{t_1}^{t_2}, \quad I_{x_i} = [f_{t_1}^{t_2} (x_i \otimes x_i) d\tau], \quad I_{x_i} u_{i0} = [f_{t_1}^{t_2} (x_i \otimes u_{i0}) d\tau]. \]

**Step 2 - Learning step:** Starting with a stabilizing controller $K^\text{local}_i$, coordinator $C_i$ solves for $P_i$ and $K^\text{local}_i$ iteratively as:
\[ \begin{pmatrix} \delta_{x_i} - 2I_{x_i} (I_{p,n} \otimes K^\text{local}_{i,k}) R_i \end{pmatrix}, \begin{pmatrix} \text{vec}(P_i) \\ \text{vec}(K^\text{local}_{i,k+1}) \end{pmatrix} = -I_{x_i} \text{vec}(Q_{i,k}). \]

$P_i$ and $K^\text{local}_{i,k+1}$ are iterated till $|P_{i,k} - P_{i,k-1}| < \epsilon$, where $\epsilon > 0$ is a chosen small threshold.

**Step 3 - Computing global controller:** Once the learning step converges, $\tilde{R}^*$ is computed distributively between the coordinators following (20). Since $K^\text{local}_{i,k}$ converges to $R_i^{-1} B_i^T P_i$ and since $R_i$ is known, $B_i^T P_i$ is available. Thereafter, the coordinator $C_i$ computes the global control input for the $i$th group following (19) as the $i$th block of the following vector:
\[ u_{i,j} = \tilde{R}^* R_{i,j}^\text{local} x_i. \]
(33)

Considering the $i$th row of $\tilde{R}^*$ is available to $C_i$, this implies that the $i$th co-ordinator must share $R_i K^\text{local}_{i,k} x_i$ with its neighboring coordinators, and vice versa, to compute their respective global controllers distributively.

**Step 4 - Applying joint controller:** Finally, every agent $j$ in the $i$th group actuates their control signal as
\[ u_{i,j} = \{K^\text{local}_{i,k} x_i\}(j) + \{u_{i,j}\}(j) \]
where, $u_{i,j}$ is the $i$th block of $u_{i,j}$, and $\{j\}$ means the $j$th element of the vector contained in $\{i\}$, $i = 1, \ldots, N$.

Note that $\bar{z}_j^i$ includes both relative position and velocity between agent $i + 1$ and agent 1. Let $C = [I_2 \ 0_2]$. Thus, for the formation control objective, we specify desired setpoints of $C \bar{z}_j^i$ as $q_j^{i,d}$, $i = 1, \ldots, p_j - 1$. Similarly, for the centroid
tracking objective, we specify the setpoint of $C\bar{z}_j$ as $q_j^T$. Because the setpoints for $C\bar{z}_j$ and $C\tilde{z}_j$ are non-zero, we take a Linear Quadratic Integral (LQI) control approach [10] and introduce an integral control to (32). Let $\tilde{q}^j = [(q_2^{j,d})^T, \cdots , (q_{p_j}^{j,d})^T, (q_{p_j+1}^{j,d})^T]^T$. We define the integral control as

$$\tilde{\zeta}_j = (I_{p_j} \otimes C)z_j - \tilde{q}^j.$$ (35)

Let $X_j = [z_j^T, \zeta_j]^T$. The formation control and target tracking objectives for group $j$ can be achieved by minimizing the objective function

$$J_j = \int_0^\infty \left( z_j^T \tilde{Q}_j z_j + u_j^T R_j u_j \right) dt$$

$$= \int_0^\infty X_j^T \tilde{Q}_j X_j + u_j^T R_j u_j dt,$$ (36)

where

$$\tilde{Q}_j = \begin{bmatrix} \tilde{Q}_{z,j} & \tilde{Q}_{z\zeta,j} \\ \tilde{Q}_{\zeta z,j} & \tilde{Q}_{\zeta \zeta,j} \end{bmatrix} \geq 0, \quad \tilde{Q}_{\zeta \zeta,j} > 0.$$ (37)

Because the LQI control minimizing (36) stabilizes the closed-loop system, we guarantee $\tilde{\zeta}_j \to 0$, which means $(I_{p_j} \otimes C)z_j \to \tilde{q}^j$ for constant $\tilde{q}^j$. The stabilizing LQI gains can be learned using Algorithm 1 without $\tilde{q}^j$.

We define $X = [X_1^T, \cdots , X_N^T]^T$ and let $S$ be a matrix such that $\bar{z} = SX_j$. We further define

$$\bar{z} = [ (\bar{z}^1)^T, \cdots , (\bar{z}^N)^T]^T$$ (38)

which consists of the centroids of all the groups. Note that $\bar{z} = (I_N \otimes S)X$ and $\bar{z}^T(L_w \otimes I_n)\bar{z} = X^T(L_w \otimes S^T S)X$.

To minimize inter-group distance given a communication topology $L_w$, we define the global objective function

$$J_g = \int_0^\infty X^T(L_w \otimes S^T S)X dt.$$ (39)

Optimizing $J_g$ will constrain the motion of the centroids to be close to their neighbors. Let $\tilde{Q} = (L_w \otimes S^T)S$, $\tilde{u} = [u_1^T, \cdots , u_N^T]^T$, $\tilde{Q} = \text{diag} \{ \tilde{Q}_1, \cdots , \tilde{Q}_N \}$, and $R = \text{diag} \{ R_1, \cdots , R_N \}$. The overall objective function is given by

$$J = \sum_{j=1}^N J_j + J_g = \int_0^\infty X^T(\tilde{Q} + \tilde{Q})X + \tilde{u}^T R \tilde{u} dt$$ (40)

which is in the form of (4).

B. Simulation example

We consider 4 groups of robots with group 1 and 4 having 3 agents and group 2 and 3 having 4 agents, i.e., $p_1 = p_4 = 3$, $p_2 = p_3 = 4$. We assume that the dynamics of the agents in (26)–(27) are the same within each group. The 4 targets are located at $[5,5]$, $[5,-5]$, $[-5,5]$, $[-5,-5]$ meters. The initial conditions of the agents are randomly generated. The communication topology between the groups is a star graph where group 1, 2, and 3 have bidirectional communication with group 4. We set $Q_j = 0.1I$, $R_j = I$, and $\tilde{Q} = 0.1(L \otimes S^T S)$, where $L$ is the unweighted graph Laplacian matrix. The mass $m_j$ and damping $c_j$ for group $j$ are set to $j$ and $0.1/j$, respectively.

To learn the controller using Algorithm 1, we add exploration noise for the initial 6, 15, 15, 6 seconds for the four groups, respectively. The sampling time $T$ of data during the initial learning period is 0.01 seconds. The desired formation of each group is an equilateral triangle and a square for the 3-agent groups and for the 4-agent groups, respectively. The side length of each polygon is 1.

After learning, the control gains from Algorithm 1 are implemented. Figure 1 shows the comparison between the trajectories generated from the optimal control and the learned approximate control. As one can see, the learned approximate control achieves the formation control and target tracking objectives. It also yields similar agent trajectories to the optimal control. In Figure 3, the control inputs are compared for agent 1 and 3 from group 1 and 3, respectively, which shows almost the same performance. Thus, the learned control approximately recovers the optimal control performance. As $Q$ increases, the discrepancy between the two controls becomes more visible. Figure 2 and 4 show the same comparison between the trajectories and the control inputs, respectively, when $Q$ is increased 10 times. We observe from Fig. 2 that although the learned approximate control achieves the formation control and target tracking objectives, the agent trajectories exhibit observable differences from the true optimal trajectories. Similarly, the difference between the learned control and the optimal control is more pronounced in Fig. 4 than in Fig. 3.

V. Conclusion

We propose a hierarchical LQR control design using model-free reinforcement learning. The design can address global and local control objectives for large multi-agent systems with unknown heterogeneous LTI dynamics, by
Fig. 2: Trajectories of the agents for $\tilde{Q} = (L \otimes S^T S)$. Solid line: optimal control. Dash-dotted line: learned approximate control. Targets are denoted by ‘+’s. Red, black, green, and blue colors indicate group 1 to 4, respectively.

Fig. 3: Comparison of control inputs for two agents (agent 1 and 3 from group 1 and 3, respectively) between optimal control (solid lines) and learned approximate control (dash-dotted lines).

Fig. 4: Comparison of control inputs for two agents (agent 1 and 3 from group 1 and 3, respectively) between optimal control (solid lines) and learned approximate control (dash-dotted lines) when $\tilde{Q}$ is increased 10 times.

dividing the agents into distinct groups. The local control for all groups can be learned in parallel, and the global control can be computed algebraically from it, thereby saving learning time. In our future work, we would like to investigate how the RL loops can be implemented in a distributed way in case the open-loop dynamics of the agents are coupled.

REFERENCES

[1] F. L. Lewis and D. Vrabie, “Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control,” IEEE Circuits and Systems Magazine, vol. 9, no. 3, 2009.
[2] K. G. Vamvoudakis, “Q-learning for Continuous-Time Linear Systems: A Model Free Infinite Horizon Optimal Control Approach,” Systems and Control Letters, vol. 100, 2017.
[3] Y. Jiang and Z. P. Jiang, Robust Adaptive Dynamic Programming, Wiley-IEEE press, 2017.
[4] S. Mukherjee, H. Bai, and A. Chakrabortty, “On Model-free Reinforcement Learning of Reduced-order Optimal Control for Singularly Perturbed Systems,” IEEE Conference on Decision and Control, Miami, FL, 2018.
[5] D. Nguyen, T. Narikiyo, M. Kawanishi, and S. Hara, “Hierarchical Decentralized Robust Optimal Design for Homogeneous Linear Multi-agent Systems,” arXiv, July 2016.
[6] D. Tsubakino, T. Yoshioka, and S. Hara, “An Algebraic Approach to Hierarchical LQR Synthesis for Large-Scale Dynamical Systems,” in proceedings of the 9th Asian Control Conference, 2013.
[7] T. Ishizaki, K. Kashima, A. Girard, J. Imura, L. Chen, and K. Aihara, “Clustered Model Reduction of Positive Directed Networks,” Automatica, vol. 59, 2015.
[8] S. Fatemi, G. Fazelnia, J. Lavaei, and M. Arcak, “Transformation of Optimal Centralized Controllers into Near-Globally Optimal Static Distributed Controllers,” IEEE Transactions on Automatic Control, vol. 64(1), 2018.
[9] L. Zhoua, Y. Lin, and Y. Wei, and S. Qiao, “Perturbation Analysis and Condition Numbers of Symmetric Algebraic Riccati Equations,” Automatica, vol. 45, pp. 1005-1011, 2009.
[10] P. C. Young and J. C. Willems, “An approach to the linear multivariable servomechanism problem,” International Journal of Control, vol. 15, no. 5, pp. 961–979, 1972.
[11] J. Liu, Y. Liu, A. Nedic, and T. Basar, “An Approach to Distributed Parametric Learning with Streaming Data,” IEEE Conference on Decision and Control, Melbourne, Australia, 2017.