Model-Free Optimal Consensus Control for Multi-Agent Systems Using Kernel-Based ADP Method

Wei Wang, Xin Chen*, Luefeng Chen, and Min Wu
School of Automation, China University of Geosciences, Wuhan, China
Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan, China
*Email: chenxin@cug.edu.cn

Abstract—Adaptive dynamic programming (ADP) is a prevalent way to solve the coupled Hamilton-Jacobi-Bellman (HJB) equations of the optimal consensus control for multi-agent systems (MAS). Neural networks (NNs) are normally used to approximate the value functions in ADP. However, NNs with manually designed features may influence the approximation ability. In this study, kernel-based methods which do not need to set the value function model structure in advance are adopted for value functions approximation. Moreover, to overcome the deficiency that most of the system dynamics are unknown, or the system is too complex to obtain the accurate dynamics. Local action value functions are defined, and kernel-based methods are used to approximate the local action value functions. Thus, an action dependent heuristic dynamic programming (ADHDP) approach using kernel-based local action value functions approximation is developed to achieve the optimal consensus control model-free. The developed approach uses historical sample data to learn the system dynamics, and avoids the traditional system identification scheme. Simulation results are provided to demonstrate the effectiveness of the presented approach.

Index Terms—Multi-agent system, adaptive dynamic programming, optimal consensus control, kernel method, model-free.

I. INTRODUCTION

Distributed control for MASs has received extensive attention in the past years. It has been used in many real-world applications, such as unmanned air vehicles [1], flocking [2], and distributed sensor network [3]. Consensus control that makes all the agents reach synchronization is one of the most important problems in MASs. In [4], the fundamental framework to solve consensus problems for MASs has been presented. Some recent surveys for consensus problems of MASs can be found in [5]. Recently, optimal consensus problem which not only consider synchronization but also minimize the energy cost under communication digraphs has been put forward [6].

To overcome optimality guarantees, game theory which provides an environment for formulating multi-agent decision control problems for dynamic interacting systems has formed a framework for optimal consensus control problem [7]. Under the game-theoretic framework, each agent optimizes its performance index independently to determine its optimal control policy. For MASs, since every agents action depends on the outcomes of itself and all the neighborhood agents. This results in coupled HJB equations which are generally impossible to be solved analytically [8]. To circumvent these deficiencies, new useful and effective optimal control schemes need to be developed.

ADP, which combines adaptive control and reinforcement learning (RL), has strong abilities of self-learning and self-adaptability. ADP algorithms can solve the optimal control problems by an online data-based procedure while the exact knowledge of the system dynamics is not required, that is, the solution of HJB equation can be obtained [9]. Recently, ADP algorithms have been applied to the optimal consensus control of MASs. By using the ADP algorithms, the schemes require the approximation of the value function and the control policy, called critic-actor structure. However, because the learning efficiency and convergence of ADP rely on the design of the critic, how to approximate the value functions is a key problem in the ADP algorithms [10].

NNs are often used in traditional critic design of MASs [11–17]. In [11], an iterative ADP method was proposed to solve a class of continuous-time (CT) non-linear two-person zero-sum differential games, where three NNs are used to construct the critic network and the optimal control pair. In [12], a cooperative policy iteration algorithm where NNs were used to approximate the value function, was adopted to obtain the adaptive learning solutions for multi-agent differential graphical games. The similar idea was found in [13] for solving the coupled integral RL HJB equations. In [14], based on the NNs-based critic design, a near-optimal control scheme was proposed to solve the nonzero-sum differential games of CT nonlinear systems. In [15], a value iteration algorithm was proposed to solve the multi-agent dynamic graphical games, two NNs were employed to approximate the value functions and the control policy. In [16], a novel RL-based cooperative tracking control scheme was proposed by using NNs for a class of multi-agent dynamic systems with disturbances and un-modeled dynamics on undirected graphs. In [17], a data-based ADP algorithm was presented to achieve the optimal consensus control of MASs, where critic NN and actor NN were employed. However, as a parametric method, NN-based value functions approximation methods often suffer from the following problems:

- The activation functions (basis functions) or the number of hidden layer of NNs are manually chosen, we can’t
judge whether the selected activation functions or the number of hidden layer are appropriate,

- Due to the local minimum in the NNs training process, how to improve the quality of the final policies is still an open problem [18].

Therefore, it motives us to circumvent the above disadvantages by using other method to approximate the value functions.

As non-parametric methods, kernel methods do not need to set the value function model structure in advance. While, kernel method can be used for feature representation, this characteristic helps improve the convergence speed and the near optimality of final control policies [19]. In previous researches, kernel methods have been adopted to approximate the value function in many RL and ADP approaches [20–22], and good performances are achieved.

Moreover, most results of the researches [11–16] require the complete or partial knowledge of the multi-agent system dynamics. Thus, it is difficult to implement those algorithms in real-time applications. However, there is substantive data in our world, and many state variables of the systems can be observed by different sensors. We can refer to data-driven methods to overcome the deficiency. A data-driven method relevant to ADHDP (it is a kind of ADP algorithm) was adopted in [17], but NNs were used to approximate the performance indices. It may suffer from the problems described above.

In this study, a kernel-based ADHDP approach is presented to achieve optimal consensus control for a class of discrete-time (DT) MAsSs without the knowledge of the system dynamics. Compared with [17], the action value functions known as the Q functions are used, and the kernel methods which rely on the historical data only are adopted to approximate the action value functions. The approach only requires the historical sample data of each agent and the reinforcement signal from the environment. Thus, model-free optimal consensus control is achieved. Furthermore, to overcome the difficulties that the structure of the kernel machines shows a linear increase with the scale of sample set increasing, linear approximately linear dependence (ALD) sparsification method [22] is adopted to collect sample data and sparsify the kernel machines.

The rest of the study is organized as follows. In section II, background on the synchronization control problem of multi-agent graphical game is provided. In section III, ADHDP algorithm using kernel method to achieve optimal consensus control is given. In section IV, simulations are carried out to verify the effectiveness of the kernel-based approach. In section V, conclusions and future work are given.

II. PRELIMINARIES

A. Algebraic graph theory

The directed graph $G = (\mathcal{V}, \varepsilon)$ is composed of a nonempty finite set of $N$ vertices $\mathcal{V} = \{v_1, \ldots, v_N\}$ and a set of edges $\varepsilon \subseteq \mathcal{V} \times \mathcal{V}$. The associated adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{N \times N}$, with $a_{ij} > 0$ if $(v_j, v_i) \in \varepsilon$, which means that the information flows from agent $j$ to agent $i$; otherwise, $a_{ij} = 0$. For $\forall i = 1, 2, \ldots, N$, $a_{ii} = 0$. The set of neighbors of node $v_i$ is $\mathcal{N}_i = \{v_j : (v_j, v_i) \in \varepsilon\}$. The in-degree matrix is expressed as $D = \text{diag}\{d_i\}$, where $d_i = \sum_{j \in \mathcal{N}_i} a_{ij}$ is the weighted in-degree of node $v_i$. The graph Laplacian matrix $\mathcal{L}$ is defined as $\mathcal{L} = D - A$. The matrix $\mathcal{L}$ has all row sums equal to zero.

Definition 1 [23]. A (directed) tree is a connected digraph where every node except one, called the root, has in-degree equal to one. A graph is said to have a spanning tree if a subset of the edges forms a directed tree.

We assume the digraph has no repeated edges and no self-loops. A digraph is said to be strongly connected, if there is a directed path from node $v_i$ to node $v_j$, for all distinct nodes $v_i, v_j \in \mathcal{V}$, and a digraph has a spanning tree if it is strongly connected. In this study, we focus on the strongly connected communication digraph with fixed topology.

B. Synchronization and tracking error dynamics

Consider the communication graph $G = (\mathcal{V}, \varepsilon)$ having $N$ agents, each with dynamics given by

$$x_i(k+1) = Ax_i(k) + B_i u_i(k), \quad i = 1, 2, \ldots, N$$

where $x_i(k) \in \mathbb{R}^n$ is the state vector of agent $i$, and $u_i(k) \in \mathbb{R}^{m_i}$ is the control input vector of agent $i$.

A leader agent $v_0$ has command generator dynamics $x_0(k) \in \mathbb{R}^n$ given by

$$x_0(k+1) = Ax_0(k).$$

Here $A \in \mathbb{R}^{n \times n}$ and $B_i \in \mathbb{R}^{n \times m_i}$, are considered to be unknown.

Remark 1: Note that the trajectory generator $A$ may not be stable. If $A$ is stable, then the command trajectory is naturally convergent, which will lose the meaning of consensus.

The objective of synchronization control design problem is to design the distributed control protocol $u_i(k)$ for $\forall i$ to make all the agents achieve synchronization with the leader using the information from the agent itself and its neighbor agents, that is $\lim_{k \to \infty} \|x_i(k) - x_0(k)\| = 0, \forall i$.

To study the consensus problem on diagraphs, the local neighborhood tracking error [24] is defined as

$$e_i(k) = \sum_{j \in \mathcal{N}_i} a_{ij} (x_j(k) - x_i(k)) + g_i (x_0(k) - x_i(k)), \quad (3)$$

where $g_i \geq 0$ is the pinning gain, $g_i > 0$ if agent $i$ is coupled to the leader; otherwise, $g_i = 0$.

From (3), the overall tracking error vector for diagraph $G$ is converted to

$$e(k) = ((\mathcal{L} + G) \otimes I_n) x_0(k) - ((\mathcal{L} + G) \otimes I_n) x(k), \quad (4)$$

where $e(k) = [e_1^T(k) \ e_2^T(k) \cdots e_N^T(k)]^T \in \mathbb{R}^{Nn}$, $G = \text{diag}\{g_i\} \in \mathbb{R}^{N \times N}$ is a diagonal matrix of pinning gains, $\otimes$ denotes the Kronecher product, $x_0(k) = [x_0^T(k) \ x_0^T(k) \cdots x_0^T(k)]^T \in \mathbb{R}^{Nn}$, and $x(k) = [x_1^T(k) \ x_2^T(k) \cdots x_N^T(k)]^T \in \mathbb{R}^{Nn}$.

The global disagreement vector is

$$\xi(k) = x(k) - x_0(k). \quad (5)$$
Then, (4) can be rewritten as
\[ e(k) = -((L + G) \odot I_n) \xi(k). \tag{6} \]
Equation (6) shows the relationship between the overall tracking error vector and the global disagreement vector. The following Lemma shows that if the communication digraph contains a spanning tree, the disagreement vector converges to zero when the tracking error converges to zero.

**Remark 2**: The matrix \((L + G)\) is non-singular if the graph has a spanning tree and \(g_i \neq 0\) for a leader agent [24].

**Lemma 1** [24]. Let \((L + G)\) be non-singular, then the synchronization error is bounded by
\[ \|\xi(k)\| \leq \|e(k)\|/\lambda_{\text{min}}(L + G), \tag{7} \]
where \(\lambda_{\text{min}}(L + G)\) is the minimum singular value of \((L + G)\).

Combining with (1), (2), (3), the dynamics of local neighborhood tracking error becomes
\[ e(k+1) = f_i(e_i(k), u_i(k), u_{ij}(k)) = A e_i(k) - (d_i + g_i) B_i u_i(k) + \sum_{j \in N_i} a_{ij} B_j u_{j}(k), \tag{8} \]
where \(u_{ij}(k)\) is the control input \(\{u_j : j \in N_i\}\) from the neighbors of agent \(i\).

From (8), it can be seen that the dynamics of local neighborhood tracking error are interacting dynamical systems driven by the control input of agent \(i\) and its neighbors. To achieve the optimal consensus control for the multi-agent system, our objective turns to optimality subject to (8).

**C. Optimal consensus control via Nash equilibrium**

To define the dynamic graphical game, the local performance indexes are defined as follows
\[
J_i(e_i(k), u_i(k), u_{ij}(k)) = \sum_{t=k}^{\infty} \gamma^{t-k} r_i(e_i(t), u_i(t), u_{ij}(t)) \tag{9}
\]
where \(r_i(e_i(k), u_i(k), u_{ij}(k)) = e_i^T(k) Q_i e_i(k) + u_i^T(k) R_i u_i(k) + \sum_{j \in N_i} u_{ij}(k)^T R_{ij} u_{ij}(k).\) All weighting matrices are time-invariant and satisfy \(Q_{ii} > 0, R_{ii} > 0, R_{ij} \geq 0, \gamma \in (0, 1)\) is the discount factor. It can be seen that the local performance indexes of agent \(i\) depend on its states and control input and control inputs of its neighbors.

The problem of achieving optimal consensus control converts to how to obtain the distributed control to minimize (9) subjected to (8).

**Definition 2** (Admissible Control) [25]. Control policies \(u_i, \forall i\) are defined as admissible if \(u_i\) stabilize systems (8), but also guarantee the performance indices (9) finite.

Under the admissible control policies, the local value functions can be written as
\[
V_i(e_i(k)) = \sum_{t=k}^{\infty} \gamma^{t-k} r_i(e_i(t), u_i(t), u_{ij}(t)) = r_i(e_i(k), u_i(k), u_{ij}(k) + \gamma V_i(e_i(k+1))). \tag{10}
\]

According to Bellman’s principle of optimality, the optimal local value functions are time-invariant and satisfy the coupled discrete time HJB (DTHJB) equation
\[
V_i^\ast(e_i(k)) = \min_{u_i(k)} \left( r_i(e_i(k) + u_i(k) + u_{ij}(k)) + \gamma V_i^\ast(e_i(k+1)) \right) \tag{11}
\]
Then the local optimal coordination control policies \(u_i^\ast(k)\) is derived by applying \(\frac{\partial V_i^\ast(e_i(k))}{\partial u_i(k)} = 0\). Combining with (8), we get
\[
u_i^\ast(k) = -\frac{\gamma}{2} (d_i + g_i) R_i^{-1} B_i^T \frac{\partial V_i^\ast(e_i(k+1))}{\partial e_i(k+1)}. \tag{12}
\]

**Remark 3**: From (12), it shows that the local optimal coordination control policies can be calculated only when the control coefficient matrices of each agent are known. However, the dynamics of MASs is always unknown or uncertain. Therefore, a new approach, the ADHPD algorithm using kernel-based action value function approximation will be introduced to achieve model-free optimal consensus control.

**Definition 3** (Global Nash Equilibrium) [26]. An \(N\)-tuple of control policies \(\{u_1^\ast, u_2^\ast, \ldots, u_N^\ast\}\) is referred to as a global Nash equilibrium solution for an \(N\)-player game, if there exists
\[
J_i^\ast \triangleq J_i(u_1^\ast, u_2^\ast, \ldots, u_i^\ast, \ldots, u_N^\ast) \leq J_i(u_1^\ast, u_2^\ast, \ldots, u_i^\ast, \ldots, u_N^\ast), (u_i \neq u_i^\ast). \tag{13}
\]
The \(N\)-tuple of the local performance values \(\{J_1^\ast, J_2^\ast, \ldots, J_N^\ast\}\) is known as a Nash equilibrium of the \(N\)-player game.

For the strongly connected communication digraph with fixed topology of multi-agent system, the local performance indexed \(J_i^\ast(e_i(k), u_i^\ast(k), u_{ij}(k))\) can be treated as the local optimal value function \(V_i^\ast(e_i(k))\) for agent \(i\), and \((u_i^\ast(k), u_{ij}^\ast(k))\) are in Nash Equilibrium [15].

**III. KERNEL-BASED OPTIMAL COORDINATION CONTROL**

The value functions defined by (10) give an evaluation for each state only. The \(Q\)-functions give an evaluation for each state-action pair, known as the action value functions. Given local coordination control policies, the local action value functions of agent \(i\) are defined as
\[
Q_i(e_i(k), u_i(k), u_{ij}(k)) = r_i(e_i(k), u_i(k), u_{ij}(k)) + \gamma Q_i(e_i(k+1), u_i(k+1), u_{ij}(k+1)). \tag{14}
\]
Better local coordination control policies \(u_i^\ast(k)\) can be obtained directly through minimizing (14). That is
\[
 u_i^\ast(k) = \arg \min_{u_i(k) \in A_i} Q_i(e_i(k), u_i(k), u_{ij}(k)), \tag{15}
\]
where \(A_i\) is the action space of agent \(i\).

For deriving the optimal control policies, policy evaluation (14) and policy improvement (15) repeat until no improvements of the control policies are observed. In this section, our intention is to develop an ADP algorithm on DT MASs using kernel method to approximate the local action value functions.
A. Framework
To implement the ADP algorithm, the critic-actor structure is employed. The critic is to approximate the local action value functions. Combining the actor and the critic, the state-action pairs \((e_i, u_i, u_{(j)})\) and \((e_{im}, u_{im}, u_{im(j)})\), respectively. \(\alpha_{im}\) are the weights, and \((e_{im}, u_{im}, u_{im(j)})\) are the selected state-action pairs in the sample data using ALD analysis, that is, choosing the samples from trajectories generated from a Markov decision process. \(k_f(\cdot, \cdot)\) is the kernel function.

B. Design of the critic
Combining (14) with (16), it is observed that

\[
\hat{Q}_i(s_i) - \gamma \hat{Q}_i(s_{i+1}) = r_i(e_i(k), u_i(k), u_{(j)}(k)),
\]

where \(\alpha_{im}\) is the solution to (17). To obtain \(\alpha_{im}\) iteratively, as studied in [21], the update rule of the critic is obtained

\[
w_i(k + 1) = \delta I / (\mu + \bar{f} d),
\]

\[
\bar{\alpha}_i(k + 1) = \bar{\alpha}_i(k) + w_i(k + 1) \left( r_i(k) - \bar{f} \bar{\alpha}_i(k) \right),
\]

\[
P_i(k + 1) = \frac{1}{\mu} \left[ P_i(k) - \delta \bar{f} P_i(k) / (\mu + \bar{f} d) \right],
\]

where \(P_i(k)\) is the step size of the critic, \(P_i(0) = \delta I\), \(\delta\) is a positive number, \(I\) is the identity matrix, \(\mu \in (0, 1)\) is the forgetting factor, \(\bar{f} = P_i(k) \tilde{h}_f(s_i(k)), \tilde{h}_f(s_i(k)) = (k_f(s_{i1}, s_i(k)), k_f(s_{i2}, s_i(k)), \ldots, k_f(s_{iM}, s_i(k)))^T\).

C. Design of the actor
The actor uses multilayer perceptron to approximate the local control policies

\[
\hat{u}_i(k) = f(e_i(k), \beta_i(k)),
\]

where \(\beta_i\) are the weights, \(f(\cdot)\) are the activation functions. The learning objective of the actor is to minimize the output of the critic. Therefore, the following objective function is used to realize the learning control objective

\[
E_{a_i} = \frac{1}{2} Q_i^2(s_i(k)).
\]

The policy gradient learning rule in the actor is designed as

\[
\Delta \beta_i(k) = \frac{\partial E_{a_i}}{\partial \beta_i(k)} = Q_i(s_i(k)) \frac{\partial Q_i(s_i(k))}{\partial u_i(k)} \frac{\partial u_i(k)}{\partial \beta_i(k)}
\]

In this study, Gaussian kernel functions are used. Thus,

\[
\hat{Q}_i(s_i) = \sum_{m=1}^{M} \alpha_{im} k_f(s_i, s_{im}) = \sum_{m=1}^{M} \alpha_{im} e^{-\frac{\|s_i - s_{im}\|^2}{\sigma^2}},
\]

where \(\sigma\) is the length-scale parameter. Combining (21) and (22), the actor learning update rule is obtained

\[
\beta_i(k + 1) = \beta_i(k) - \eta_i(k) \Delta \beta_i(k) = \beta_i(k) - \eta_i(k) \hat{Q}_i(s_i(k))
\]

\[
\times \frac{\partial u_i(k)}{\partial \beta_i(k)} \sum_{m=1}^{M} \alpha_{im} \left( u_i(k) - u_{im} \right) e^{-\frac{(\gamma s_i(k) - s_{im})^2}{\sigma^2}},
\]

where \(\eta_i(k)\) is the step size of the actor.

D. Convergence analysis
The kernel-based methods can be treated as linear function approximators. According to the results in [27], when linear function approximators are used in critic-actor algorithms, these algorithms can be proved to converge if the update in the critic is faster than the actor. By making use of the update rule (18) in the critic and appropriately choosing the step sizes \(\eta_i(k)\) of the actor, the critic learning process can be treated as a faster recursion than the actor. Thus, convergence of the proposed ADP method using kernel-based local action value functions approximation can be guaranteed.

IV. SIMULATION RESULTS
In this section, simulations are carried out to demonstrate the effectiveness of the kernel-based approach. Consider the diagraph with four agents studied in [15], as Fig. 1 shows. The system models are given as

\[
\begin{align*}
x_0(k + 1) &= Ax_0(k), \\
x_1(k + 1) &= Ax_1(k) + B_i u_i(k),
\end{align*}
\]

where

\[
A = \begin{bmatrix} 0.995 & 0.09983 \\ -0.09983 & 0.995 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0.2047 \\ 0.08984 \end{bmatrix},
\]

\[
B_2 = \begin{bmatrix} 0.2147 \\ 0.2895 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 0.2097 \\ 0.1897 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix}.
\]

The pining gain \(g_1 = g_2 = g_3 = 0, g_4 = 1\), and the edge weights \(a_{12} = 0.8, a_{14} = 0.7, a_{23} = 0.6, a_{31} = 0.8\). The weighting matrices are given as \(Q_{11} = Q_{22} = Q_{33} = Q_{44} = I_{2 \times 2}, R_{11} = R_{22} = R_{33} = R_{44} = 1, \ R_{13} = R_{21} = R_{32} = R_{41} = 0, \ R_{12} = R_{14} = R_{23} = R_{31} = 1.\)

80 samples of each agent \(i\) are collected using ALD to construct the sample set with threshold \(\tau = 0.001\). The forgetting factor \(\mu = 1\), positive number \(\sigma = 0.6\), and the discount factor \(\gamma = 0.95\). The learning rates of the actor is \(\eta_i = 0.01, \forall i\), and the length-scale parameter of the kernel function is \(\sigma = 1.8\).
The initial state of the systems are chosen randomly in $(0, 1)$. The local action value functions of the initial four states and actions during the learning process are shown in Fig. 2. It can be seen that, all four local action value functions are convergent. The neighborhood tracking error dynamics and the dynamics of the four agents are shown in Fig. 3 and Fig. 4, respectively. The phase plane plots of four agents are shown in Fig. 5. It is shown that the four agents achieve synchronization to the leader.

V. CONCLUSION

Optimal consensus control of MASs not only considers synchronization but also minimizes the energy cost. In this study, an ADHDP algorithm using kernel method to approximate the local action value functions has been presented. By using the algorithm, model-free optimal consensus control for DT MASs is achieved. The main contributions of this study are as follows:

- The value functions approximation is the key part of ADP, parametric methods such as NN-based methods are often used for approximation. In this study, as non-parametric methods, kernel methods are used for approximation, and pre-determined value function model structure design is avoided.
- Local action value functions ($Q$ functions) are defined in this study. The use of local $Q$ functions achieves deriving the optimal coordination control policies model-freely.
- Using kernel methods to approximate the local $Q$ functions, an ADHDP algorithm is presented to achieve the optimal consensus control of MASs without knowing the system dynamics.

Currently, the kernel function in this study is manually selected, which has a certain degree of subjectivity. It may be helpful to adjust the kernel function model by changing the hyper-parameters of the kernel function. Thus, more accurate value functions approximation will be achieved. Future work will focus on discussing the feasibility of updating the value function and the kernel function model simultaneously, and...
ACKNOWLEDGMENT

The authors would like to thank the support by National Natural Science Foundation of China under Grant 61473316 and Grant 61210011, the Hubei Provincial Natural Science Foundation of China under Grant 2015CFA010 and the 111 project under Grant B17040.

REFERENCES


