Selectively Decentralized Q-Learning

Thanh Nguyen
Dept. of Computer and Information Science
Indiana University - Purdue University - Indianapolis
Indiana, United States
thamnguy@imail.iu.edu

Snehasis Mukhopadhyay
Dept. of Computer and Information Science
Indiana University - Purdue University - Indianapolis
Indiana, United States
smukhopa@cs.iupui.edu

Abstract— **In this paper, we explore the capability of selectively decentralized Q-learning approach in learning how to optimally stabilize control systems, as compared to the centralized approach. We focus on problems in which the systems are completely unknown except the possible domain knowledge that allow us to decentralize into subsystems. In selective decentralization, we explore all of the possible communication policies among subsystems and use the cumulative gained Q-value as the metric to decide which decentralization scheme should be used for controlling. The results show that the selectively decentralized approach not only stabilizes the system faster but also shows superior converging speed on gained Q-value in different systems with different interconnection strength. In addition, the selectively decentralized converging time does not seem to grow exponentially with the system dimensionality. Practically, this fact implies that the selectively decentralized Q-learning could be used as an alternative approach in large-scale unknown control system, where in theory, the Hamilton-Jacobi-Bellman-equation approach is difficult to derive the close-form solution.**

Keywords—selective decentralization, Q-learning, control system

I. Introduction

To tackle major issues in large scale systems learning, decentralization has been one of the major topics in learning and adaptive control. Decentralization employs the possible domain-knowledge to decouple the entire system's state variables into subsystems, assigns a learning agent for each subsystem and applies the learning algorithms on each subsystem. Compared to centralization, decentralization is less susceptible to uncertain system parameters [1], may adapt to the structure change in the system, [2], overcomes the curse of dimensionality and may improve the converging speed. Although decentralization is a promising approach, it may suffer from instability because of the interconnection among subsystems [1, 3]. To overcome the stability issue, two major solutions: partial communication and multi-model-switching, have been proposed to integrate subsystem interaction into the learning algorithms. In partial communication, each subsystem is responsible to select the other subsystems to communicate with, depending on the subsystem's state variable and other circumstances [4]. In multi-model-switching, given a number of known communication schemes, a central coordinator is responsible to switch the communication scheme depending on certain circumstances [5-7].

Decentralization has been widely applied in Q-learning [8], which is one of the most well-known model-free-techniques for learning in unknown environment [9-13], especially when the systems are naturally distributed [14, 15]. In Q-learning, the learning agent maintains the optimal values for all state-action entries in its Q-table. In each state, the learning agent chooses the action by the highest Q-table entry for the state. After each visit, the learning agent updates the former state-action Q-value by the new state's reward and highest Q-value. Most of the decentralized Q-learning approaches adapt the partial communication idea: each subsystem manages its own communication and updates its own Q-table. Although decentralized Q-learning has been well-established, there are still two open questions in this approach. First, how well does decentralized Q-learning tackle the slow rate of converging weakness in Q-learning [16]? Second, how do we apply multi-model-switching in decentralized Q-learning, which has not been thoroughly explored?

In this paper, from the multi-model-switching idea, we propose the selective decentralization approach in Q-learning. We apply selective decentralization in several learning and control problems in which the systems are unknown and the learning agents aim to optimally stabilize the systems using Q-learning. The learning agents know the possible decentralization schemes among them, which is bounded by the Bell's number theory [17]. In our learning and control problems, we design the state-reward function with linearity property such that the central state-reward value is equivalent to the sum of all sub-state rewards. From this argument, we choose the cumulative gained Q-value to decide the best communication scheme. In addition, we provide comparison on converging speed between selectively decentralized Q-learning and centralized Q-learning, which is often absent in many other decentralized Q-learning literatures. Since, the relationship between the decentralized Q-table and the centralized Q-table is difficult for theoretical analysis for this question, most of our results only serve as confirmation studies.

II. Method

A. Problem statement

In this paper, we are interested in the systems in the general form

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t)) \quad (1)$$

This is the author's manuscript of the article published in final edited form as:

Nguyen, T., & Mukhopadhyay, S. (2017). Selectively decentralized Q-learning. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 328–333). https://doi.org/10.1109/SMC.2017.8122624

where $\mathbf{x} \in \Re^n$ is the joint state vector, $\mathbf{u} \in \Re^m$ is the joint action (also called control) vector, $f: \Re^{n+m} \to \Re^n$ is a general nonlinear unknown function. We assume that f has the stable equilibrium point $f(\mathbf{0}, \mathbf{0}) = \mathbf{0}$. The main objective is to learn the sequence of action units $\mathbf{u}(t)$ to stabilize $\mathbf{x}$

$$\mathbf{x}(t) \to \mathbf{0}, \mathbf{u}(t) \to \mathbf{0} \text{ as } t \to \infty \quad (2)$$

In order to apply Q-learning, we need to discretize the system (1). Therefore, to simplify the discretization, we assume that the system (1) has the following properties:

- Each dimension of $\mathbf{x}$ is symmetrically bounded by $[-\chi, \chi]$, where $\chi > 0$ is a known boundary for $\mathbf{x}$.
- Each dimension of $\mathbf{u}$ is symmetrically bounded by $[-\mu, \mu]$, where $\mu > 0$ is a known boundary for $\mathbf{u}$.

To apply selectively decentralized Q-learning, we restate the following assumptions for system (1), as showed in [18]. First, the system could be decoupled in to K subsystems, where each subsystem could be assigned to an independent learning agent. Each subsystem knows which components of $\mathbf{x}$ and $\mathbf{u}$ belonging to it. Second, since f is unknown, each subsystem k does not know the relationship between the current sub-state $\mathbf{x}_k(t)$ and previous sub-state/sub-action $[\mathbf{x}_k(t\text{-}1), \mathbf{u}_k(t\text{-}1)]$. Each subsystem does not know the interconnection among itself and the other subsystems.

We proposed the selective decentralization idea in [18] for model-based system control. Briefly, the key theme in selective decentralization is the joint structures in which the subsets of agents fully cooperate to learn the optimal actions. The number of possible decentralized schemes for k subsystems is B_k (the k^{th} Bell's number), which grows super-exponentially. We design a central coordinator unit to decide which decentralization structure could provide the best learning performance. In each scheme, there are $L \leq K$ groups such that each group contains one or more subsystems/agents communicating to execute Q-learning. In a group, inside agents do not communicate with any outside agents.

B. System discretization and reward function

Let M be the number of intervals in each dimension of $\mathbf{x}$ and $\mathbf{u}$ for which we uniformly divide the dimension into small grids. Therefore, the entire state space is divided into M^n small hyper cubes with edge $\theta_x = 2\chi/M$. The control space is divided into M^m small hyper cubes with edge $\theta_u = 2\mu/M$. All points inside a hyper cube are discretely represented by the center of the hyper cube. Points on the border between two hyper cubes are represented by the center of the 'left' hypercube. Mathematically, the discretization process is described by the following formulas

$$\mathbf{x}[i] \to \theta_x + \chi/M \;\; \forall i \in [1, n] \; and \; \mathbf{x}[i] \in [\theta_x, \theta_x + 2\,\chi/M) \quad (3)$$
$$\mathbf{u}[i] \to \theta_u + \mu/M \;\; \forall i \in [1, m] \; and \; \mathbf{u}[i] \in [\theta_u, \theta_u + 2\,\mu/M) \quad (4)$$

where $\theta_x \in \{-\chi, -\chi + 2\chi/M, -\chi + 4\chi/M, ..., \chi - 2\chi/M\}$ and $\theta_u \in \{-\mu, -\mu + 2\mu/M, -\mu + 4\mu/M, ..., \mu - 2\mu/M\}$, which are the 'left' boundaries in the hyper cubes. We denote $\mathbf{x}_{\text{dis}}$ and $\mathbf{u}_{\text{dis}}$ as the discrete space and control vector of $\mathbf{x}$ and $\mathbf{u}$, correspondingly.

With the discretization process in (3) and (4), it is easy to see that when M is odd, the zero vector $\mathbf{0}$ is one of the discrete space/control vectors. Given this condition, we define the state reward function $q(\mathbf{x})$ as

$$q(\mathbf{x}) = \sum_{i=1}^{n} q(i), \text{where } q(i) = \begin{cases} -\mathbf{x}_{\text{dis}}(i)^2 \text{ if } \mathbf{x}_{\text{dis}}(i) \neq 0 \\ r \text{ if } \mathbf{x}_{\text{dis}}(i) = 0 \end{cases} \quad (5)$$

where $r > 0$ is a small bonus factor when the discrete $\mathbf{x}_{\text{dis}}$ is $\mathbf{0}$, or $\mathbf{x}$ is within the hypercube containing the equilibrium point. Since our main objective is to stabilize (1), with reward function (5), the learning problem aims to maximize

$$J(\mathbf{x}) = \sum_{t=0}^{\infty} \gamma^t q\big(\mathbf{x}(t)\big) \quad (6)$$

where $0 < \gamma < 1$ is the discount factor. The learning problem (3-6) is similar to a classical exploration problem in [19], where there is only one terminated state with positive reward and all of the other states show negative reward. It is important to note that the choice of M and $q(\mathbf{x})$ could be flexible. The necessary condition is that the discrete reward should be higher for the states which are closer to the stable point.

C. Selectively decentralized Q-learning

First, we rewrite (5) and (6) for subsystem as follow. Let n_1, n_2, $\ldots n_K$ be the dimensionality of the K subsystems. Certainly, $n_1 + n_2 + \ldots + n_K = n$. Let $\{i_1\}$, $\{i_2\}$, $\ldots$ $\{i_K\}$ be the set of indexes of $\mathbf{x}$ and $\mathbf{u}$ belonging to these subsystems. In subsystem k, we denote $\mathbf{x}\{i_K\}$ and $\mathbf{u}\{i_K\}$ as the sub-state and sub-action vectors. Thus, (5) becomes

$$q\big(\mathbf{x}(\{i_k\})\big) = \sum_{\forall i \in \{i_k\}} q(i),$$
$$\text{where } q(i) = \begin{cases} -\mathbf{x}_{\text{dis}}(i)^2 \text{ if } \mathbf{x}_{\text{dis}}(i) \neq 0 \\ r \text{ if } \mathbf{x}_{\text{dis}}(i) = 0 \end{cases} \quad (7)$$

In each subsystem, at each iteration, the Q-learning is executed according to [19]

$$Q[\mathbf{x}_{\text{dis}}\{i_k\}(t-1), \mathbf{u}_{\text{dis}}\{i_k\}(t-1)] = (1-\alpha) Q[\mathbf{x}_{\text{dis}}\{i_k\}(t-1), \mathbf{u}_{\text{dis}}\{i_k\}(t-1)] + \alpha\left(q\big(\mathbf{x}_{\text{dis}}\{i_k\}(t-1)\big) + \gamma \max_{\mathbf{u'}_{\text{dis}}\{i_k\}} Q[\mathbf{x}_{\text{dis}}\{i_k\}(t), \mathbf{u'}_{\text{dis}}\{i_k\}]\right) \quad (8)$$

where $Q[\mathbf{x}_{\text{dis}}\{i_k\}, \mathbf{u}_{\text{dis}}\{i_k\}]$ denotes the Q table in subsystem k and $0 < \alpha < 1$ is the learning rate.

Suppose that the decentralization scheme b partitions the entire system into L disjoint components $c_1, c_2, \ldots, c_L$ with dimensionality $n_1, n_2, \ldots, n_L$. Each component contains one or more subsystems. For any component c_l, let $\{I_l\} = \cup \{i_k\}$ be the union of indexes from all subsystems k belonging to c_l. In this component, the Q-learning is executed according to (7) and (8) with index set $\{I_l\}$.

Since the number of possible decentralization schemes in a K-subsystem is B_K [17], the main question in selective decentralization is which scheme b is the 'best'. In this work, we select the scheme b returning the highest cumulative gained Q value, which is

$$\Omega(b) = \sum_{l=1}^{L} \alpha \begin{pmatrix} q\big(\mathbf{x}_{\text{dis}}\{I_l\}(t-1)\big) + \\ \gamma \max_{\mathbf{u'}_{\text{dis}}\{I_l\}} Q[\mathbf{x}_{\text{dis}}\{I_l\}(t), \mathbf{u'}_{\text{dis}}\{I_l\}] - \\ Q[\mathbf{x}_{\text{dis}}\{I_l\}(t-1), \mathbf{u}_{\text{dis}}\{I_l\}(t-1)] \end{pmatrix} \quad (9)$$

Let w be the window index covering the time from $t = (w\text{-}1)\,\Omega + 1$ to $t = w\Omega$. In this window, we choose the same decentralization scheme to decide the optimal action $\mathbf{u'}_{\text{dis}}$ for $\max_{\mathbf{u'}_{\text{dis}}} Q[\mathbf{x}_{\text{dis}}(t), \mathbf{u'}_{\text{dis}}]$. Larger window size implies less scheme switching. Pseudo code of procedure ***QLearning_Window*** shows more details on how we execute selectively decentralized Q-learning in each window.

Procedure *QLearning_window* (w)
Persistent input: Q tables in all $B(K)$ decentralization schemes
S: array to store the cumulative gained Q value
b: best decentralization scheme
if $w = 1$
Initialize all Q tables as 0 in all decentralization schemes
Choose a random decentralization scheme as b
Reset S to 0
for t from $(w\text{-}1)\,\Omega + 1$ to $w\Omega$
// use b to compute the action
for all components l in b
Compute $\mathbf{u}_{\text{dis}}\{I_l\}(t)$ as
$$\max_{\mathbf{u'}_{\text{dis}}\{I_l\}} Q[\mathbf{x}_{\text{dis}}\{I_l\}(t), \mathbf{u'}_{\text{dis}}\{I_l\}] \quad (11)$$
Assembly $\mathbf{u}_{\text{dis}}(t)$ from all $\mathbf{u}_{\text{dis}}\{I_l\}(t)$
// update the cumulative Q-value gained
for all decentralization schemes β
S[β] = S[β] + $\Omega(\beta)$
Update Q tables according to (8), with $\mathbf{u}_{\text{dis}}(t)$
Choose b as $\operatorname{argmax}_{\beta} S[\beta]$

In (10), if there are multiple $\mathbf{u'}_{\text{dis}}\{I_l\}$ returning the same optimal Q value for $\mathbf{x}_{\text{dis}}\{I_l\}(t)$, we randomly select one instance. We choose the cumulative gained Q-value as the choice of decentralization scheme because the state-reward function (5), (7) satisfy the first linearity assumption: *For any decentralization b separating system (1) into L components such that these component are completely disjoint, the sum of components' state-rewards is equal to the centralized state-reward.*

$$q(\mathbf{x}) = \sum_{l=1}^{L} q\big(\mathbf{x}(\{I_l\})\big) \quad (12)$$

III. TOY EXAMPLE RESULTS

A. Converging speed of selectively decentralized Q-learning

We perform experiments on several toy examples from the same class of system to show the superior converging speed of selectively decentralized Q-learning, compared to centralized Q-learning – which is defined in (5) and (6). In these examples, we examine the convergence from two points of view: the closeness of $\mathbf{x}(t)$ toward 0 and the magnitude of cumulative Q-value increase.

The systems used in these examples are in the format

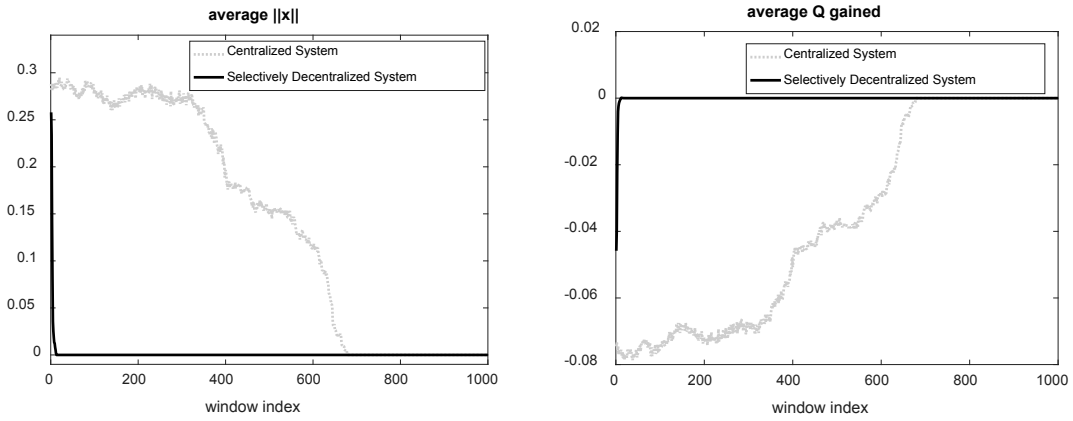

Fig.1. Comparison between centralized and selectively decentralized Q-learning in completely decoupled 3-subsystem.

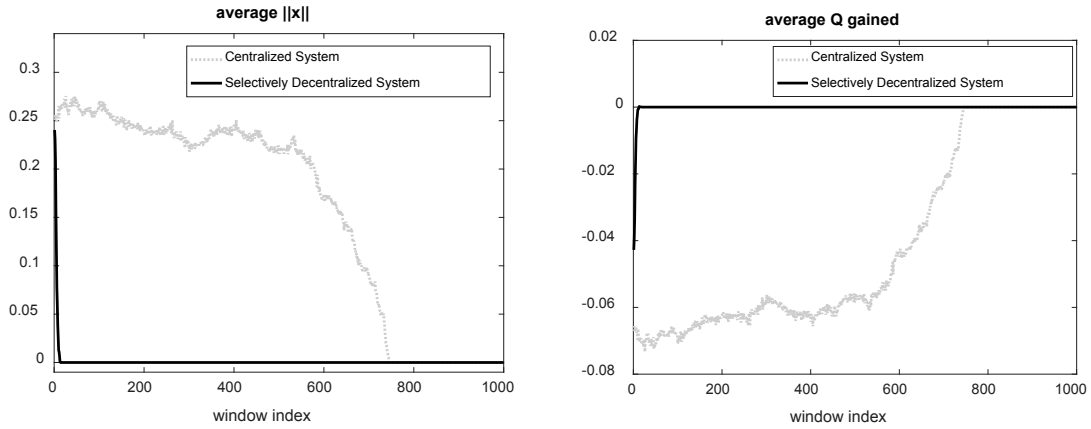

Fig.2. Comparison between centralized and selectively decentralized Q-learning in strongly coupled ($\sigma = 0.5$) 3-subsystem.

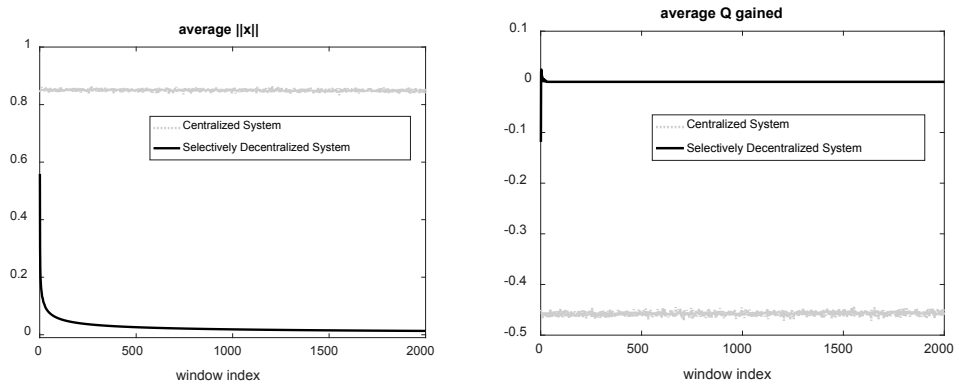

Fig.3. Comparison between centralized and selectively decentralized Q-learning in completely decoupled 6-subsystem.

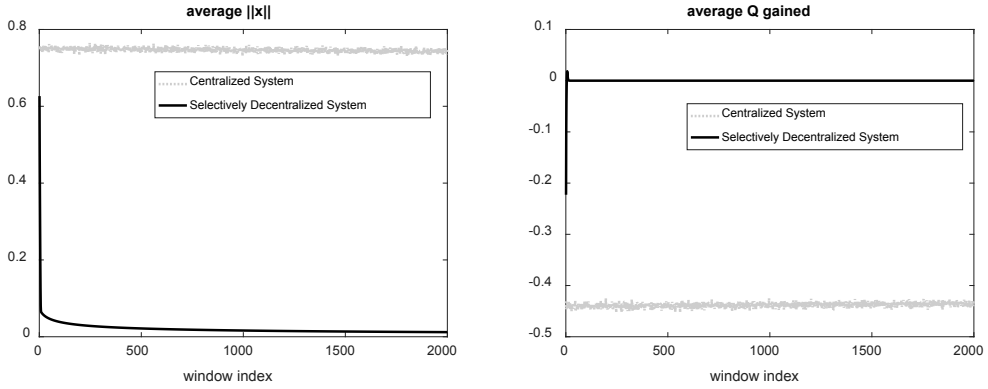

Fig.4. Comparison between centralized and selectively decentralized Q-learning in strongly coupled (σ = 0.5) 6-subsystem.

$$\mathbf{x}(t) = \sin(\mathbf{A}\mathbf{x}(t\text{-}1) + \mathbf{u}(t\text{-}1)) \tag{13}$$

where $\mathbf{A}$ are $n \times n$ random Markov matrices such that all diagonal entries share the same value. $\mathbf{x}$ and $\mathbf{u}$ have the same dimensionality for the ease of decentralization. Each subsystem corresponds to one state/action dimension. The vector sin function is defined from each dimension as

$$\sin(\mathbf{x}) = \begin{bmatrix} \sin(\mathbf{x}_1) \\ \sin(\mathbf{x}_2) \\ \vdots \\ \sin(\mathbf{x}_n) \end{bmatrix} \tag{14}$$

We define the coupling parameter σ as

$$\sigma = \frac{\sum_{i \neq j} \mathbf{A}_{ij}}{\sum \mathbf{A}_{ij}} \quad \forall i, j \in [1, n] \tag{15}$$

In other words, σ is the ratio between the sum of non-diagonal entries in $\mathbf{A}$ and the sum of all entries in $\mathbf{A}$. With $\sigma = 0$, $\mathbf{A}$ becomes the identity matrix or the systems are completely decoupled. The systems are more couple when σ increases. For state and action variables, $\chi = \mu = 0.5$ and initial state vectors $\mathbf{x}(0)$ are uniformly random numbers. For discretization (3-4), we choose $M = 5$. Therefore, $\theta_x = \theta_u = 0.2$. For Q-learning parameters (5-8), we choose $r = 0.01$, $\alpha = 0.1$ and $\gamma = 0.9$. We test system (13) with number of subsystems $n = 3$, 4, 5 and 6 and window size $w = 50$. For each choice of n, we repeat the experiments 100 times and report the average value due to the randomness of $\mathbf{A}$ and $\mathbf{x}(0)$.

Figures 1-4 highlight two significant advantages of selectively decentralized Q-learning, compared to centralized Q-learning. First, selectively decentralized Q-learning converges faster in both completely decoupled systems and strongly coupled systems. This fact suggests that selectively decentralized technique could be applied to many systems with wide-range of interconnection. Second, the converging time of selectively decentralized Q-learning grows much slower than the convergence time in centralized Q-learning. Due to the lack of space, we only draw the result when $n = 3$ and $n = 6$ to highlight the change in system dimensionality. As showed in figures 3 and 4, the centralized Q-learning does not converge within 100,000 iterations (or 2000 windows). Figures 5-6 show more details on how selectively decentralized Q-learning converges within the first few tens of windows.

B. Switching among decentralization schemes

In Figure 7, we repeat all experiments in the previous section with $w = 1$ (the most frequent switching scenario) to show that selectively decentralized Q-learning will stop switching the 'best' decentralization scheme. Here, in order to compare with figure 5, we draw the average number of scheme switches for every 50 iterations/windows (to recall, in the previous section, we set $w = 50$). Comparing figures 5 and 6, in most of the cases, we observe that the point when number of scheme switches drop to 0 is earlier than the point when the selectively decentralized Q-learning converges. This result may suggest that selectively decentralized Q-learning may learn the optimal communication policy during the optimal stabilization process.

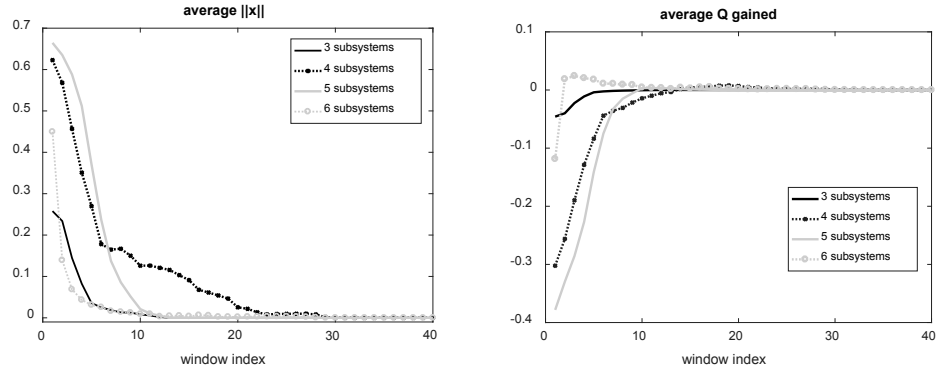

Fig. 5. Convergence of selectively decentralized Q-learning in the first few of tens windows when the systems are completely decoupled.

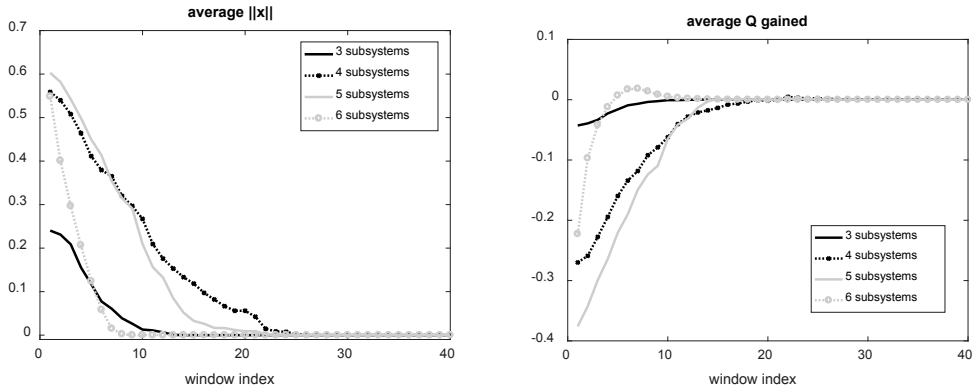

Fig.6. Convergence of selectively decentralized Q-learning in the first few of tens windows when the systems are strongly coupled (σ= 0.5).

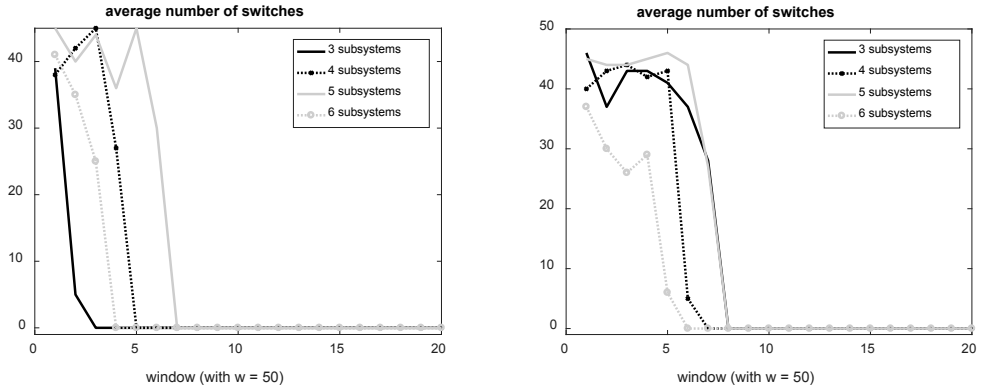

Fig. 7. Switching in selectively decentralized Q-learning. Left: completely decoupled systems. Right: strongly coupled (σ= 0.5) systems.

IV. CONCLUSIONS

In this paper, we show the superior converging performance of selectively decentralized Q-learning, compared to centralized Q-learning, in several control problems. The results suggest that decentralized Q-learning could be practically applied as an alternative approach for unknown system control problems. Real-world application of Q-learning, such as robotics and networking, could be found in [9-15]. In these problems, the conventional Hamilton-Jacobi-Bellman equation approach may not provide the close-form solution in general [20-23].

Since the theoretical analysis of Q-learning, decentralized and distributed Q-learning mostly focuses on the existence of the optimal Q-value and the guarantee of reaching the optimal Q-value [8, 9, 24], we lack the theoretical explanation for the drastic superior converging speed of decentralized Q-learning. In this section, we try to explain the superior performance of selectively decentralized Q-learning from two points of view. First, as stated in the foundation of Q-learning [8], the convergence of Q-learning assumes that all of the state-action entries in the Q-table are visited infinitely. Therefore, in order to converge to the optimal Q, the Q-learning systems are supposed to spend time to explore the Q-table. In figures 1 and 2 where we show the convergence of centralized Q-learning, there are long periods when $\|\mathbf{x}\|$ and accumulate Q-gained change slowly. These periods may correspond to the exploration phases. Because the number of states, actions, and state-action entries

grow exponentially with system dimensionality, decentralized Q-learning into smaller dimension may also improve the convergence exponentially due to exponentially less search space. Second, selectively decentralized Q-learning proposes more search options than centralized Q-learning, which is another factor to improve the converging speed. In centralized Q-learning, a newly visited state has no prior information to estimate its Q-table entries. With the same state, in selectively decentralized Q-learning, the components of the state have higher chance to be visited by the subsystem learner (in different centralized states), which may reduce the effort to compute the optimal Q-value.

There are two major open questions in this paper. First, although selectively decentralized Q-learning may reduce exponential convergence measured by the number of data points/iterations, the number of decentralization schemes also grows exponentially with the dimensionality. Therefore, in practice, more refined techniques are needed to reduce the search of decentralization schemes. At this point, we believe that selectively decentralized Q-learning is practically useful because the best decentralization schemes stop switching after a few of tens windows (Figure 7). Second, we choose best decentralization scheme by the sum of subsystems' gained Q-values only because of the linearity in state-reward function, which is the main driver for Q-value update. However, there is no theoretical basis to support whether or not the different sum of subsystem gained Q-value in different decentralization scheme is comparable. There may exist more solid options for choosing the best decentralization scheme than cumulative gained Q-value.

Another exploration this paper should take is the impact of Q-learning parameters, such as γ and α in equation (9), on the overall learning performance. However, due to the limitation of space, we decide not to present this point. The parameters used in this paper is selected similar to well-known examples in [19]. Different experiments may require different choices of parameters.

V. ACKNOWLEDGEMENT

The research presented in this paper was supported by a National Science Foundation grant (No. ECCS-1407925).

REFERENCES

[1] Ioannou, P.A.: 'Decentralized adaptive control of interconnected systems', Automatic Control, IEEE Transactions on, 1986, 31, (4), pp. 291-298

[2] Shi, L., and Singh, S.K.: 'Decentralized adaptive controller design for large-scale systems with higher order interconnections', Automatic Control, IEEE Transactions on, 1992, 37, (8), pp. 1106-1118

[3] Gavel, D.T., and Siljak, D.: 'Decentralized adaptive control: structural conditions for stability', Automatic Control, IEEE Transactions on, 1989, 34, (4), pp. 413-426

[4] Narendra, K., Oleng, N., and Mukhopadhyay, S.: 'Decentralised adaptive control with partial communication', IEE Proceedings-Control Theory and Applications, 2006, 153, (5), pp. 546-555

[5] Han, Z., and Narendra, K.S.: 'New concepts in adaptive control using multiple models', Automatic Control, IEEE Transactions on, 2012, 57, (1), pp. 78-89

[6] Narendra, K.S., and Balakrishnan, J.: 'Improving transient response of adaptive control systems using multiple models and switching', Automatic Control, IEEE Transactions on, 1994, 39, (9), pp. 1861-1866

[7] Narendra, K.S., and Mukhopadhyay, S.: 'To communicate or not to communicate: A decision-theoretic approach to decentralized adaptive control'. Proc. American Control Conference (ACC), 20102010 pp. Pages

[8] Watkins, C.J., and Dayan, P.: 'Q-learning', Machine learning, 1992, 8, (3-4), pp. 279-292

[9] Arslan, G., and Yuksel, S.: 'Decentralized Q-Learning for Stochastic Teams and Games', IEEE Transactions on Automatic Control, 2016

[10] Liu, K., and Zhao, Q.: 'Distributed learning in multi-armed bandit with multiple players', IEEE Transactions on Signal Processing, 2010, 58, (11), pp. 5667-5681

[11] Matignon, L., Laurent, G.J., and Le Fort-Piat, N.: 'Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams', in Intelligent Robots and Systems, 2007, IEEE/RSJ International Conference on., pp. 64-69

[12] Lauer, M., and Riedmiller, M.: 'An algorithm for distributed reinforcement learning in cooperative multi-agent systems', In Proc. of the Seventeenth International Conference on Machine Learning. 2000.

[13] Kiumarsi, B., Lewis, F.L., Modares, H., Karimpour, A., and Naghibi-Sistani, M.-B.: 'Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics', Automatica, 2014, 50, (4), pp. 1167-1175

[14] Galindo-Serrano, A., and Giupponi, L.: 'Distributed Q-learning for aggregated interference control in cognitive radio networks', IEEE Transactions on Vehicular Technology, 2010, 59, (4), pp. 1823-1834

[15] Morozs, N., Clarke, T., Grace, D., and Zhao, Q.: 'Distributed Q-learning based dynamic spectrum management in cognitive cellular systems: Choosing the right learning rate', in Computers and Communication (ISCC), 2014 IEEE Symposium on, pp. 1-6

[16] Narayanan, V., and Jagannathan, S.: 'Distributed adaptive optimal regulation of uncertain large-scale interconnected systems using hybrid Q-learning approach', IET Control Theory & Applications, 2016, 10, (12), pp. 1448-1457

[17] Rota, G.-C.: 'The number of partitions of a set', The American Mathematical Monthly, 1964, 71, (5), pp. 498-504

[18] Nguyen, T., and Mukhopadhyay, S.: 'Identification and Optimal Control of Large-scale System Using Selective Decentralization'. Proc. IEEE International Conference on Systems. Man and Cybernetics, Budapest2016 pp. Pages

[19] Russell, S., and Norvig, P.: 'Artificial Intelligence: A Modern Approach, Third Edition' (Pearson, 2010. 2010)

[20] Abu-Khalaf, M., and Lewis, F.L.: 'Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach', Automatica, 2005, 41, (5), pp. 779-791

[21] Saridis, G.N., and Lee, C.-S.G.: 'An approximation theory of optimal control for trainable manipulators', Systems, Man and Cybernetics, IEEE Transactions on, 1979, 9, (3), pp. 152-159

[22] Beard, R.W., Saridis, G.N., and Wen, J.T.: 'Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation', Automatica, 1997, 33, (12), pp. 2159-2177

[23] Huang, C.-S., Wang, S., and Teo, K.: 'Solving Hamilton—Jacobi—Bellman equations by a modified method of characteristics', Nonlinear Analysis: Theory, Methods & Applications, 2000, 40, (1), pp. 279-293

[24] Sastry, P., Phansalkar, V., and Thathachar, M.: 'Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information', IEEE Transactions on systems, man, and cybernetics, 1994, 24, (5), pp. 769-777