Identification and Optimal Control of Large-Scale Systems Using Selective Decentralization

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 selective decentralization in improving the control performance for unknown large-scale systems using model-based approaches. In selective decentralization, we explore all of the possible communication policies among subsystems and show that with the appropriate switching among the resulting multiple identification models (with corresponding communication policies), such selective decentralization significantly outperforms a centralized identification model when the system is weakly interconnected, and performs at least equivalent to the centralized model when the system is strongly interconnected. To derive the sub-optimal control, our control design include two phases. First, we apply system identification to train the approximation model for the unknown system. Second, we find the suboptimal solution of the Halminton-Jacobi-Bellman (HJB) equation to derive the suboptimal control. In linear systems, the HJB equation transforms to the well-solved Riccati equation with closed-form solution. In nonlinear systems, we discretize the approximation model in order to acquire the control unit by using dynamic programming methods for the resulting Markov Decision Process (MDP). We compare the performance among the selective decentralization, the complete decentralization and the centralization in our two-phase control design. Our results show that selective decentralization outperforms the complete decentralization and the centralization approaches when the systems are completely decoupled or strongly interconnected.**

Keywords— decentralized control, nonlinear control, linear control, Markov Decision Process, Dynamic Programming

I. INTRODUCTION

To tackle major issues in in large scale adaptive control design, decentralization has been one of the major topics in control system since 1970s. Decentralization employs the domain-knowledge to decouple the entire system's state variables into subsystems, and applies the control algorithms on each subsystem. With decentralization, the control algorithms operate on less number of state variables; therefore, the control algorithms are less susceptible to uncertain system parameters [1]. In addition, decentralization may adapt to the structure change in the system, which the centralized system often fails to do [2]. In addition, decentralization with smaller state space may allow the control algorithm to overcome the curse of dimensionality and improve the convergence speed. Most of the decentralization methods focus on solving linear system [2, 3], in which the centralized and decentralized system could be uniformly represented in matrix form; therefore it is feasible to observe the estimated system parameter. Theory for the stability and controllability of the decentralized linear system has been found in [4, 5]. Application of decentralized control could be found in many large-scale system such as power networks, urban traffic networks or ecology systems [6]. Although decentralization is a promising approach for large-scale adaptive control, this type of approach is likely to suffer from instability because of interconnection among subsystems regardless of the interconnection strength [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 control algorithm. In partial communication, each subsystem is responsible to select the other subsystems to communicate with, depending on the subsystem's state variable and other circumstance [7]. In multi-model-switching, the entire system has K policies to allow the subsystems to communicate, and the system has a central communicator who is responsible to switch the communication policy depending on certain circumstances [8-10]. However, the question of how the subsystems should communicate in the system is still open, because the number of communication policies grows following the Bell's number, which is more than exponential [11]. Other practical questions in decentralization are: how to create and justify the subsystems, and how fast the control algorithms converge.

Besides the decentralization problem, solving the Halminton-Jacobi-Bellman equation is the key in the control system theory. For the linear system, the HJB equation becomes the well-known Ricatti equation with complete solution [12]. However, in most of the real-world cases, the system is nonlinear where the close-form solution for HJB equation is very difficult to find. Therefore, researchers have been focusing on approximation methods to tackle nonlinear HJB equation problem such as [13-16]. Generally, these efforts focus on the nonlinear feedback-linearization system, in which the close-form solution for the approximation of HJB equation has been found [17]. Theoretically, the HJB equation could be solved

with dynamic programming [18]. In principle, the dynamic programming method updates the state-utility function (sometimes called the cost function) and the state-control function (sometimes called the policy function) iteratively [19, 20]. Therefore, a simple idea is to discretize the nonlinear system to convert it into a Markov-decision-process (MDP) and solve it by the policy iteration algorithm [21]. Discretization of nonlinear control system has been studied in [22, 23].

In this paper, we propose the selective decentralization method to control the completely unknown system in two-phase approach: system identification and control. In addition, we create a simple discrete-MDP approach to tackle the nonlinear control in the most general form. From our knowledge, the approach using decentralized method with system identification to unknown system control is relatively unexplored. Supposing that the sub-components of the unknown system is given by domain knowledge, we also compare the control performance of our selective decentralization method with the completely decentralized method and the centralized method.

II. Problem Statements

A. Decentralized control in linear system

For linear system, we study the discrete-time invariant linear-quadratic-regulator (LQR) unknown system:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (1)$$

where $\mathbf{x} \in \Re^n$ is the state vector, $\mathbf{u} \in \Re^n$ is the control and $\mathbf{A} \in \Re^{n \times n}$ is the unknown state-transitional matrix. The main objective is to find the sequence of control $\mathbf{u}$ to minimize the infinite performance function

$$J = \sum_{t=0}^{\infty} (\mathbf{x}(t)\mathbf{Q}\mathbf{x}(t)^T + \mathbf{u}(t)\mathbf{R}\mathbf{u}(t)^T) \quad (2)$$

where $\mathbf{Q}$ and $\mathbf{R}$ are $\Re^{n \times n}$ semi positive-definite matrices. Since $\mathbf{A}$ is unknown, the intermediate objective is to find the approximation matrix $\widehat{\mathbf{A}} \in \Re^{n \times n}$ for $\mathbf{A}$ in order to apply existing control algorithms such that with the predicted state vector:

$$\hat{\mathbf{x}}(t+1) = \widehat{\mathbf{A}}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (3)$$

the identification error

$$e(t) = |\mathbf{x}(t) - \hat{\mathbf{x}}(t)|^2 \quad (4)$$

approaches 0 as $t \to \infty$.. In decentralized control, let k be the number of subsystems in (1) with dimension n_1, n_2, … n_k such that $\sum_{i=1}^{k} n_i = n$. Decentralized identification computes $\widehat{\mathbf{A}}$ as the block-diagonal matrix

$$\widehat{\mathbf{A}} = \begin{bmatrix} \widehat{\mathbf{A}_1} & & & \\ & \widehat{\mathbf{A}_2} & & \\ & & \ddots & \\ & & & \widehat{\mathbf{A}_k} \end{bmatrix} \quad (5)$$

B. Decentralized control in nonlinear system

For nonlinear system, we study the discrete-time invariant nonlinear system in arbitrary form:

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t)) \quad (6)$$

where $\mathbf{x} \in \Re^n$ is the state vector, $\mathbf{u} \in \Re^m$ is the control input and $f \in \Re^{n+m} \to \Re^n$ is the completely unknown nonlinear state transition function. To ease the formulation for decentralization, here we assume that each state variable corresponds to one control unit, which means $m = n$. Similar to the linear system, the main objective is to find the sequence of control $\mathbf{u}$ to minimize the infinite performance function defined in equation (2). The intermediate goal is to find the approximated nonlinear function $\hat{f}$ from a set of basis nonlinear functions (i.e. polynomial, neural network) such that with the predicted state vector

$$\hat{\mathbf{x}}(t+1) = \hat{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (7)$$

the identification error $e(t)$ approaches 0 as $t \to \infty$. Similar to the linear system, let k be the number of subsystems in (6). Decentralized identification models $\hat{f}$ as follow

$$\hat{\mathbf{x}}(t+1) = \begin{bmatrix} \hat{\mathbf{x}}_1(t+1) \\ \hat{\mathbf{x}}_2(t+1) \\ \vdots \\ \hat{\mathbf{x}}_k(t+1) \end{bmatrix} = \hat{f}(\mathbf{x}(t), \mathbf{u}(t)) = \begin{bmatrix} \hat{f}_1(\mathbf{x}_1(t), \mathbf{u}_1(t)) \\ \hat{f}_2(\mathbf{x}_2(t), \mathbf{u}_2(t)) \\ \vdots \\ \hat{f}_k(\mathbf{x}_k(t), \mathbf{u}_k(t)) \end{bmatrix} \quad (8)$$

C. Statement of the Decentralized Identification Problem

More precisely, let w dynamical system (Σ) be described by the equation:

$$\Sigma: \boldsymbol{x}(t+1) = f[\boldsymbol{x}(t), u(t), \theta] \quad (9)$$

where $x \in \mathbb{R}^N$ where N is a large number. It is assumed in this problem that the input $u(t)$ is known. θ is an unknown parameter vector in $\mathbb{R}^M$ where the dimension M is large. The objective is to estimate θ using measurements of the overall system.

In the problem of interest to us, the system is assumed to consist of r subsystems of low dimension which are interconnected. If the state vectors of the subsystems $\Sigma_1, \Sigma_2, \dots, \Sigma_r$ are respectively $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_r$, it is assumed that each subsystem can be described by the difference equation

$$\Sigma_i: \boldsymbol{x}_i(k+1) = f_i[\boldsymbol{x}_i(k), u_i(k), \theta_i] + \sigma_i g[\boldsymbol{z}_i(k)] \quad (10)$$

where the parameter σ_i is assumed to be small, and $[\boldsymbol{x}_i, \boldsymbol{z}_i] = \boldsymbol{x}^T$(i.e., the elements of $\boldsymbol{z}_i$ are state variables not contained in $\boldsymbol{x}_i$).

A decentralized model can be set up as:

$$\hat{\boldsymbol{x}}_i(t+1) = \hat{f}_i[\boldsymbol{x}_i(t), \boldsymbol{z}_i(t), u(t), \theta(t)] \quad (11)$$

At this stage, the knowledge that each subsystem has about the components of z that affect it, becomes important. We distinguish between two distinct cases:

Every subsystem knows all the state variables that affect its outputs (known decentralization structure).

Every subsystem Σ_i knows the small set of variables in $\boldsymbol{z}_i$ that might affect its outputs, but does not know exactly which variables do affect them (unknown decentralization structure).

In the former case, Σ_i uses a single model set up with the correct decentralization structure, but in the latter case it uses multiple models corresponding to different possible

decentralization structures, and switches between them. The former is referred to as strict decentralization, and the latter as selective decentralization.

Selective decentralization policy: The number of possible decentralization structures for k subsystems is B_k (the kth Bell's number), which grows super-exponentially. We set up a separate identification model for each such decentralization structure and adaptively switch among the models implementing the different decentralization policies to determine the best model. It is clear that a large of computation is carried out by the multiple identification models. However, since computation in many applications is cheap as compared to speed of convergence, we justify the increase in computational load by the significant improvement in performance resulting from such a selectively decentralized approach.

III. Decentralized system identification

A. Decentralized linear system identification

The theory for linear time-invariant system identification has been well-studied. The gradient decent is one of the most robust methods as shown in [24]. The gradient decent minimizes the second norm of the identification error

$$J = e(t)^2 = \left(\mathbf{x}(t) - \hat{\mathbf{A}}\mathbf{x}(t-1) - \mathbf{u}(t-1)\right)^T \ldots \left(\mathbf{x}(t) - \hat{\mathbf{A}}\mathbf{x}(t-1) - \mathbf{u}(t-1)\right) \quad (12)$$

Taking the derivative of J respect to $\hat{\mathbf{A}}$ yields

$$\frac{\partial J}{\partial \hat{\mathbf{A}}} = -\mathbf{x}(t-1)\left(\mathbf{x}(t) - \hat{\mathbf{A}}\mathbf{x}(t-1) - \mathbf{u}(t-1)\right)^T \quad (13)$$

From the Newton-Raphson method, $\hat{\mathbf{A}}$ could be iteratively updated as

$$\hat{\mathbf{A}}(t) = \hat{\mathbf{A}}(t-1) - \alpha \mathbf{H}(J, \hat{\mathbf{A}}) \frac{\partial J}{\partial \hat{\mathbf{A}}} \quad (14)$$

where $\mathbf{H}(J, \hat{\mathbf{A}})$ is the Hessian matrix of J on $\hat{\mathbf{A}}$ and α is the learning rate. A reasonable approximation of $\mathbf{H}(J, \hat{\mathbf{A}})$ is

$$\mathbf{H}(J, \hat{\mathbf{A}}) = \mathbf{x}(t-1)\mathbf{x}(t-1)^T + \mathbf{I} \quad (15)$$

where $\mathbf{I}$ is the identity matrix. Substituting (15) and (13) to (14) we have

$$\hat{\mathbf{A}}(t) = \hat{\mathbf{A}}(t-1) - \alpha \frac{\left(\mathbf{x}(t) - \hat{\mathbf{A}}(t-1)\mathbf{x}(t-1) - \mathbf{u}(t-1)\right)\mathbf{x}(t-1)^T}{1 + \mathbf{x}(t-1)^T\mathbf{x}(t-1)} = \hat{\mathbf{A}}(t-1) - \alpha \frac{(\mathbf{x}(t) - \hat{\mathbf{x}}(t))\mathbf{x}(t-1)^T}{1 + \mathbf{x}(t-1)^T\mathbf{x}(t-1)} \quad (16)$$

B. Identification of nonlinear time-invariant systems

In this work, we use the three-layer feedforward neural network as the basis function to approximate f as $\hat{f}$ in nonlinear system identification. Neural networks have been known for their capability to approximate a large and general class of nonlinear functions over compact domains. Theoretical foundation and application of neural network as such universal function approximators in control systems can be found in [13, 25, 26]. We use the backpropagation training/learning algorithm for neural networks [27], in which $\{\mathbf{x}(t-1), \mathbf{u}(t-1)\}$ are presented at the input layer and $\hat{\mathbf{x}}(t)$ is computed at the output layer of the neural network identification model.

IV. Control designs

A. Control in linear system

In the linear system (1), the control vector u could be solved by finding the solution of the Riccati equation [28].

$$\mathbf{A}^\mathbf{T}\mathbf{P}\mathbf{A} - \mathbf{P} - \mathbf{A}^T\mathbf{P}\mathbf{B}(\mathbf{B}^T\mathbf{P}\mathbf{B} + \mathbf{R})^{-1}\mathbf{B}^T\mathbf{P}\mathbf{A} + \mathbf{Q} = 0 \quad (17)$$

The solution $\mathbf{P}$ could be solve by DARE algorithm implemented by Arnold et al [29]. At each iteration, by replacing $\mathbf{A}$ by the approximator $\hat{\mathbf{A}}(t)$ and solution $\mathbf{P}(t)$, the control vector $\mathbf{u}(t)$ is computed by

$$\mathbf{u}(t) = -(\mathbf{R} + \mathbf{B}^T\mathbf{P}(t)\mathbf{B})^{-1}\mathbf{B}^T\mathbf{P}(t)\hat{\mathbf{A}}(t)\mathbf{x}(t) \quad (18)$$

B. Nonlinear control system

We design the control mechanism with two phases: control iteration and identification. In the identification phase, we train the neural networks according to [21] to acquire the function approximators $\hat{f}$ from using $<\mathbf{x}(t), \mathbf{u}(t)>$ as the input tuples and $\mathbf{x}(t+1)$ as the outputs. After training $\hat{f}$, we use $\hat{f}$ to update the state-action transitional matrix for the MDP. We apply the Monte Carlo simulation to convert from $\hat{f}$ to the MDP [30]. In the control phase, from the MDP, we use policy iteration algorithm to compute the estimated optimal policy [21]. Here, the window size parameter M decides how frequently we call the identification phase. In other words, M decides the number of $<\mathbf{x}(t), \mathbf{u}(t), \mathbf{x}(t+1)>$ tuples to train $\hat{f}$.

In this design, discretization is the most important step to ensure the performance of the overall control system. In this work, we apply the ad-hoc method to discretize $\mathbf{x}$ and $\mathbf{u}$ based on the distribution of $\mathbf{x}(t+1)$ on $<\mathbf{x}(t), \mathbf{u}(t)>$. Different systems have different discretization schemes, as shown in each case study in the later sections.

C. Selective decentralization in control system

We propose a selective decentralization method to tackle the decentralized identification problem in systems with unknown decentralization structures. Here, we assume that the components of the system are known by the domain knowledge, i.e. combustion engine, brake, wheels of a car; however, the interconnection among these components is unknown. Below is the description of selective decentralization method using a simplified example.

For the system of k component, the number of possible decentralization schemes is $B(k)$ (the k^{th} Bell number [11]), or the number of possible partition for the set of size k. $B(k)$ schemes cover all possible number of subsystems from 1 to k. For example, with $k = 3$, we have $B(k) = 5$ possible decentralization schemes: {{1, 2, 3}}, {{1, 2}, {3}}, {{1, 3}, {2}}, {{1}, {2, 3}} and {{1}, {2}, {3}}, in which each scheme

has 3, 2, 2, 2 and 1 subsystem(s), correspondingly. A subsystem only uses its state and control variable to compute its own approximator. For example, with scheme {{1, 2}, {3}}, we have the format $\widehat{\mathbf{A}} = \begin{bmatrix} \widehat{\mathbf{A}}_{1,2} & \\ & \widehat{\mathbf{A}}_{13} \end{bmatrix}$ for linear system and $\hat{f} = \begin{bmatrix} \hat{f}_{1,2} \\ \hat{f}_3 \end{bmatrix}$. In this example, $\widehat{\mathbf{A}}_{1,2}$ and $\hat{f}_{1,2}$ are computed only using $\mathbf{x}_1$,$\mathbf{x}_2$, $\mathbf{u}_1$ and $\mathbf{u}_2$ according to formula (11) and backpropagation training algorithm, meanwhile $\widehat{\mathbf{A}}_3$ and $\hat{f}_3$ are computed only using $\mathbf{x}_3$ and $\mathbf{u}_3$.

Let Ω be the time-window size and w be the window index. Then the window w covers the discrete time index from $t = (w\text{-}1)\ \Omega + 1$ to $t = w\Omega$. Let $E(w)$ be the window-identification error at window w, which is the average of $e(t)$ from $t = (w\text{-}1)\ \Omega + 1$ to $t = w\Omega$. Let ε and γ be two small numbers for thresholding. The pseudo code for selective decentralization is as follow

initialize b: the best decentralization scheme

for w from 1 to the maximum window index

calculate **u** ***using*** b using Riccati equation for linear system and discrete-MDP for nonlinear system

Train approximator and compute $E(w)$ for $B(k)$ decentralization schemes

Select the decentralization scheme with the lowest $E(w)$ as b

if w > 1 and ($E(w) < \varepsilon$ or $|E(w) - E(w\text{-}1)| / |E(w)| < \gamma$)

return final identification error $E(w)$ and convergence time $w\Omega$.

(stop the identification process)

end if

end for

V. SIMULATION CASE STUDIES

A. Linear system

In this simulation, we setup a system of 6 state and 6 control variables with $k = 3$ for equation (1). The unknown transitional block matrix **A** is setup with real components {{1,2}, {3, 4}{5, 6}}as follow

$$\mathbf{A} = \begin{bmatrix} 0.7 & 0.3 & & & & \sigma \\ 0.2 & 0.8 & & & & \\ & & 0.23 & 0.77 & & \\ & & 0.4 & 0.6 & & \\ & & & & 0.5 & 0.5 \\ \sigma & & & & 0.35 & 0.65 \end{bmatrix} \quad (17)$$

where the non-block entries of **A** are a random number between 0 and σ. The control variables $\mathbf{u}_t$ and initial state variable $\mathbf{x}_1$ are set randomly between -1 and 1. As shown in (17), σ decides the interconnection strength among system components. We call σ

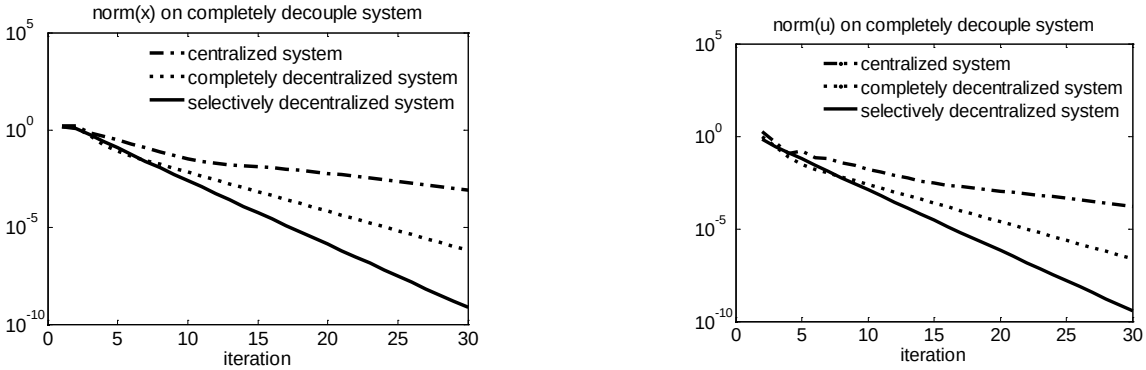

Fig 1. Comparison of control performance among the centralized system, the completely decentralized system and the selectively decentralized system when the systems are linear and completely decouple.

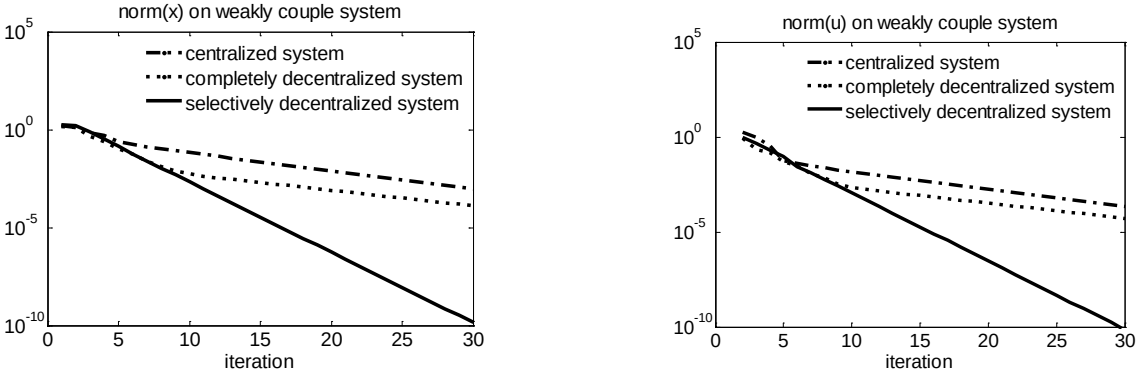

Fig 2. Comparison of control performance among the centralized system, the completely decentralized system and the selectively decentralized system when the systems are linear and weakly couple.

coupling parameter. We setup the completely decouple system by setting $\sigma = 0$ and the strongly couple system by $\sigma = 0.1$. To avoid numerical overflowing, we normalize **A** into a Markov matrix before identification in (1). We set **B**, **Q** and **R** as the identity matrix. For identification, we set $\alpha = 1$. At the starting point, we randomly set **x**(0) between -1 and 1.

In Figures 1 and 2, we observe that the selectively decentralized system shows better control performance than the completely decentralized system and the centralized system. In these figures, we draw the y-axis in log scale due to the numerical characteristics of **x** and **u**. We use norm(x) and norm(u) to denote the second-norm of **x** and **u**, correspondingly. Clearly, after 15 iterations, both **x** and **u** in the completely decentralized system converge to 0 significantly faster than they do in the completely decentralized system and the centralized system. At the first few iterations, the selectively decentralized system shows slightly poorer control performance. This may due to the complexity of the selectively decentralized system in identifying unknown **A**. In the other hands, as the systems are more couple, we see that the performance gap between the decentralized systems and the centralize system is less.

B. Nonlinear system

In this example, we choose the system

$$f(\mathbf{x}(t), \mathbf{u}(t)) = sigmoid(\mathbf{A}\mathbf{x}(t) + \mathbf{u}(t)) - 0.5 \quad (18)$$

where **x**, **u** $\in R^6$, matrix **A** is defined in equation (17) and the *sigmoid* function is defined as

$$sigmoid(\mathbf{x}) = \begin{bmatrix} 1/(1+e^{-x_1}) \\ \vdots \\ 1/(1+e^{-x_6}) \end{bmatrix} \quad (19)$$

Here, we assume that the boundary of **x** an **u** is known as $-1 \le x_i, u_i \le 1 \ \forall i \in [1,6]$.

For system approximation, we use a three-layer neural network with 30 hidden units and backpropagation to train the neural network for $\hat{f}$. We call the identification phase after every 50 iterations in the control phase. At the starting point, we set **x**(0) as a vector of random numbers between -1 and 1. Similar to the linear system case study, we setup the completely decouple system by setting $\sigma = 0$ and the strongly couple system by $\sigma = 0.1$

In the MDP-discretization method, we discretize **x** and **u** into $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ as follow

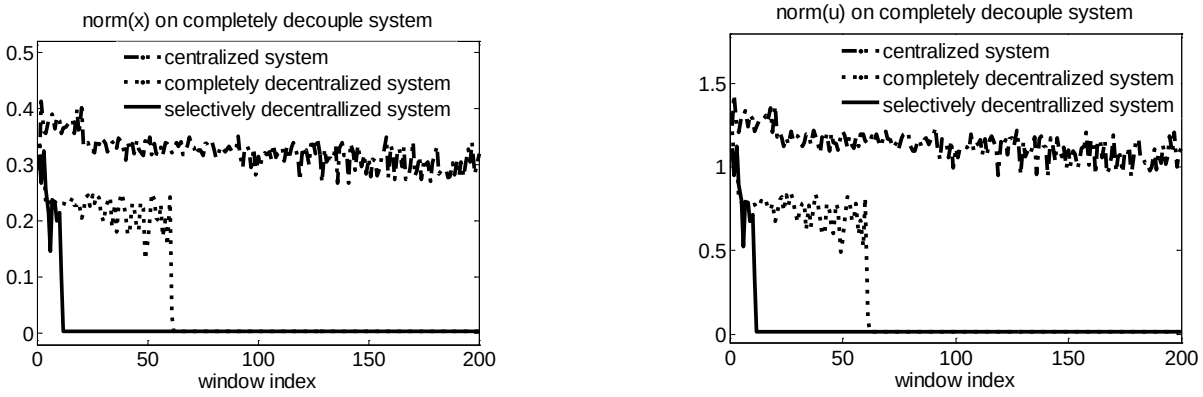

Fig 3. Comparison of control performance among the centralized system, the completely decentralized system and the selectively decentralized system when the systems are nonlinear and completely decouple.

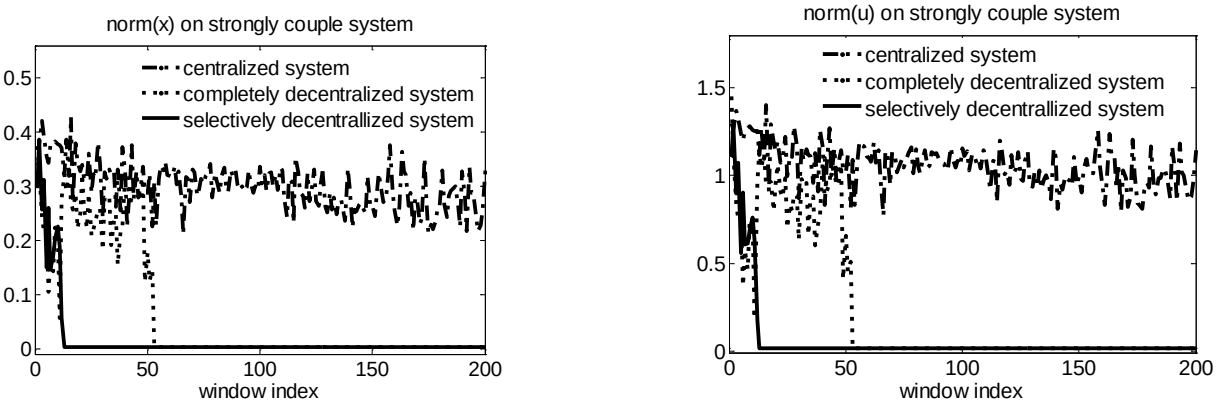

Fig 4. Comparison of control performance among the centralized system, the completely decentralized system and the selectively decentralized system when the systems are nonlinear and strongly couple.

$$\bar{x}_i = \begin{cases} -1 \text{ if } x_i < -0.01 \\ 0 \text{ if } -0.01 \le x_i < 0.01 \\ 1 \text{ if } x_i > 0.01 \end{cases} \quad (20)$$

and

$$\bar{u}_i = \begin{cases} -1 \text{ if } u_i < -0.01 \\ 0 \text{ if } -0.01 \le u_i < 0.01 \\ 1 \text{ if } u_i > 0.01 \end{cases} \quad (21)$$

We choose the discretization thresholds [-0.01, 0.01] in formula (23) due to the natural distribution of $\mathbf{x}(t+1)$ when $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are uniformly generated from -1 to 1. As can be seen in figure 5, $\mathbf{x}(t+1)$ follows the bell-shape distribution centered at 0. Therefore, with discretization thresholds [-0.01, 0.01], the solution of the MDP for system (21) ensures that the boundary of the control performance (2) is $12\times(0.01)^2 = 0.0012$ when the input and control are 6-dimmensional vector and **R**, **Q** are identity matrices.

In Figures 3 and 4 we observe that the selectively decentralized system shows better control performance than the completely decentralized system and the centralized system in the major of cases.. In these figures, we observe that the selectively decentralized systems converge **x** and **u** to 0 after less than 20 iterations, which is significantly faster than the completely decentralized systems do.

VI. Conclusions

In this paper, we show that selective decentralization can improve the control performance in both linear and nonlinear systems with several levels of interconnection among subsystems. In addition, we show that the discrete-MDP technique could solve the nonlinear control problem in general form. Practically, this fact is important to apply selective decentralization in adaptive and learning systems, such as in reinforcement learning, in which the system is required to perform not only precisely but also quickly, even when the data set and the time window of operation are limited. However, the discrete-MDP technique in this work is limited to case studies. The discretization thresholds need the distribution of the next state assuming that the current state and control vectors are uniformly distributed.

Acknowledgment

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] Wang, S.-H., and Davison, E.J.: 'On the stabilization of decentralized control systems', Automatic Control, IEEE Transactions on, 1973, 18, (5), pp. 473-478

[5] Corfmat, J.-P., and Morse, A.S.: 'Decentralized control of linear multivariable systems', Automatica, 1976, 12, (5), pp. 479-495

[6] Sandell, N.R., Varaiya, P., Athans, M., and Safonov, M.G.: 'Survey of decentralized control methods for large scale systems', Automatic Control, IEEE Transactions on, 1978, 23, (2), pp. 108-128

[7] 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

[8] 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

[9] 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

[10] 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)

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

[12] Bellon, J.: 'Riccati Equations in Optimal Control Theory', 2008

[13] 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

[14] 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

[15] 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

[16] 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

[17] Lewis, F.L., and Syrmos, V.L.: 'Optimal control' (John Wiley & Sons, 1995. 1995)

[18] Chui, C.K., and Chen, G.: 'Linear Systems and optimal control' (Springer Science & Business Media, 2012. 2012)

[19] Al-Tamimi, A., Lewis, F.L., and Abu-Khalaf, M.: 'Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof', Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2008, 38, (4), pp. 943-949

[20] Bradtke, S.J., Ydstie, B.E., and Barto, A.G.: 'Adaptive linear quadratic control using policy iteration'. Proc. American Control Conference

[21] Russell;, S., and Norvig, P.: 'Artificial Intelligence A Modern Approach' (Prentice Hall, 2010, 3rd edn. 2010)

[22] Munos, R., and Moore, A.W.: 'Variable resolution discretization for high-accuracy solutions of optimal control problems', Robotics Institute, 1999, pp. 256

[23] Munos, R., and Moore, A.: 'Variable resolution discretization in optimal control', Machine learning, 2002, 49, (2-3), pp. 291-323

[24] Keesman, K.J.: 'System Identification: an Introduction' (Springer-Verlag, 2011. 2011)

[25] Funahashi, K.-I.: 'On the approximate realization of continuous mappings by neural networks', Neural networks, 1989, 2, (3), pp. 183-192

[26] Miller, W.T., Werbos, P.J., and Sutton, R.S.: 'Neural networks for control' (MIT press, 1995. 1995)

[27] Rumelhart, D.E., Hinton, G.E., and Williams, R.J.: 'Learning representations by back-propagating errors', Cognitive modeling, 1988, 5, (3), pp. 1

[28] Lancaster, P., and Rodman, L.: 'Algebraic riccati equations' (Clarendon press, 1995. 1995)

[29] Arnold III, W.F., and Laub, A.J.: 'Generalized eigenproblem algorithms and software for algebraic Riccati equations', Proceedings of the IEEE, 1984, 72, (12), pp. 1746-1754

[30] Bishop, C.M.: 'Pattern Recognition', Machine Learning, 2006, pp. 537-54