MODEL-FREE *Q*-LEARNING OPTIMAL RESOURCE ALLOCATION IN UNCERTAIN COMMUNICATION NETWORKS

G.CETIN
Department of Electrical & Biomedical engineering
University of Nevada, Reno
Reno, USA

M. Sami Fadali
Department of Electrical & Biomedical engineering
University of Nevada, Reno
Reno, USA

H. Xu
Department of Electrical & Biomedical engineering
University of Nevada, Reno
Reno, USA

***Abstract*—This paper examines optimal resource allocation for uncertain communication networks with application to Cyber-Physical Systems (CPS). The aim of the paper is to test the hypothesis that Q-learning can provide an adaptive strategy that reduces or eliminates congestion in a real-time communication network by appropriate channel bandwidth allocation. To demonstrate this, we use a local linear state-space network model with congestion as the network state and bandwidth as the control input. A novel adaptive estimator is developed to approximate the ideal optimal cost in real-time and obtain the network control gain using the Bellman equation. Simulation results illustrate the effectiveness of the proposed Q-learning approach in eliminating network congestion. The closed-loop system is shown to be asymptotically stable and the estimated control inputs approaches the ideal optimal control signals asymptotically in the presence of model uncertainty.**

I. INTRODUCTION

Cyber-Physical Systems (CPS) is a multidisciplinary technology combining a physical process with embedded computation and networking technologies where embedded networked devices sense, monitor and control the physical world [15]. Merging these multidisciplinary, the design of CPS requires control, communication and computing innovations to meet the specific challenges, performance requirements, reliability, and complicated functionality of CPS [12], [2]. The research trends in CPS are categorized and summarized by [9] as energy control, secure control, transmission and management, control technique, system resource allocation, and model-based software design.

CPS includes different types of networks such as wired network, wireless network, Bluetooth, WLAN, GSM, etc. and distributed systems with limited resources. The allocation of these limited system resources, in particular computing resources and network bandwidth, are performance considerations for CPS [9]. Zhiwen and Hontago [16] proposed a bandwidth allocation strategy based on measurement error in a networked system with multiple control loops controlled by a remote controller. In this strategy, bandwidth is divided into three parts, namely; (i) guaranteed cost bandwidth, which guarantees the stability of all control loops, (ii) available bandwidth, which represents the available bandwidth for allocation to each channel, and (iii) non-real-time data bandwidth which denotes bandwidth consumption for transmitting non-real-time data. The purpose of their allocation strategy is to allocate additional bandwidth for each control loop in direct proportion to its measurement error. In [17], Zhiwen and Hontago modeled congestion and bandwidth as a linear time invariant system with congestion as state and network bandwidth allocation as control input. They used a Linear Quadratic Regulator (LQR) to eliminate congestion by dynamic bandwidth allocation.

Although the model of [17] can be used to provide an approximate model for local network behavior in a CPS, the model parameters are difficult to obtain in practice. At best, the model parameters can be approximately estimated and the model will include significant modeling errors. In addition, the dynamics of the network are more likely to include nonlinearities that cannot be represented by the simple linear model of [17]. Therefore, it is essential to develop a model-free real-time network resource allocation strategy that can effectively manage the network resource in the presence of network model uncertainty.

By combining approximate dynamic programming (ADP) techniques [1] with optimal control [6], a Q-learning based control was developed. Compared with conventional optimal control, the ADP-based control (i) can learn the optimal control in the forward-in-time manner instead of traditional optimal control which requires solving the Riccati equation backward in time, and (ii) does not require a model of the system dynamics. *Q*-learning is a popular model-free technique that can obtain an optimal control strategy assuming a local linear model and adapt the control as the operating conditions of the system change. Although local linear dynamics are used, the overall behavior of the system need not be linear and the method is applicable to nonlinear dynamics.

The main aim of this paper is to test the hypothesis that Q-learning can provide an optimal resource allocation strategy for network resources in CPS.

To test our hypothesis we use the simple state-space model of [17] and compare to the performance of LQR. LQR with the correct system model provides a reference case against which we test other strategies. We compare its response to that of controllers based on a perturbed model of the system, first using LQR then using Q-learning. We show that Q-learning can provide performance that approaches that of LQR with the correct model whereas LQR with a perturbed model fails to provide acceptable performance.

The remainder of the paper is organized as follows. Section II presents the background of network control for CPS and formulates the optimal network control problem. The simplified state-space network model used in this study is presented in Section III. In Section IV, the model-free *Q*-learning optimal network resource allocation is developed. Section V illustrates the effectiveness of the proposed scheme through numerical simulation and Section VI provides concluding remarks.

II. GPS AND NETWORK CONGESTION MODEL

This section provides background material on CPS then develops a network congestion model. We use the congestion as the controlled variable for the optimal network control problem.

A. Cyber-Physical Systems

As shown in Figure 1, CPS includes a physical plant with multiple actuators and sensors and a remote controller which communicates with the actuators and sensors through a shared communication network. More specifically, each sensor collects the relevant real-time data from the physical plant and transmits it to the remote controller to update the control signals. After receiving the real-time data, the controller computes the control signals and transmits them to the appropriate actuator. However, since multiple actuators and sensors transmit the real-time data through the shared communication network whose resources are limited, it is critical to develop an effective network control algorithm that can optimally allocate the network resources.

In this paper, network congestion control is studied. Specifically, the scheduler utilizes the outputs of the controller to allocate the bandwidth of the communication network and select the sampling period of the control system [17], [7]. Since network bandwidth is defined as the capacity of transmission of information, higher network bandwidth capacity reduces congestion while lower bandwidth capacity increases congestion [17]. To test the hypothesis investigated in this paper, it is essential to use a suitable network congestion model. In this study, we adopt the state-space model of [17]. The state variables of the model are network congestion in each channel $x_j,\ j = 1, 2, \dots, n,$ where n is the number of channels. Note that the network congestion state is the input to the scheduler which is used to allocate the bandwidth of each channel. The network congestion model is developed in the next subsection.

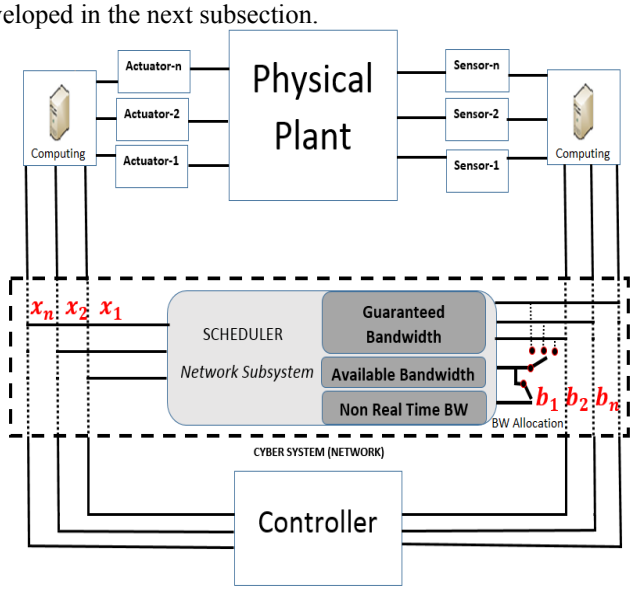

Figure 1 Schematic of a MIMO CPS[17],[7]

B. Network Congestion Model

According to [16], [17], dynamic bandwidth allocation is an effective means of minimizing network congestion. Therefore, bandwidth allocation can be considered as the control input of a network congestion model. However, due to practical network constraints and the possible need for extra bandwidth, bandwidth cannot be directly used in the network congestion model. According to [16] and [17], network bandwidth can be divided into three parts: guaranteed cost bandwidth B_{min}, available bandwidth B_a,and non-real time data bandwidth B_{other} . B_{min} Represents the minimum bandwidth consumption that guarantees the stability of all control loops. B_{other} denotes bandwidth consumption for transmitting non-real time data, B_a represents the available bandwidth which can be allocated and added into B_{min}[16]. In addition, the bandwidth used by all n control loops must be less than the total bandwidth capacity B_g i.e.

$B_g \geq B_{other} + B_{min} + B_a$

For each individual control loop, bandwidth consumption b_i can be defined as

$$b_i = \frac{m_i}{h_i}, i = 1,2, \dots, n \quad (1)$$

where m_i is the time spent to transmit data and h_i is the sampling period in the i^{th} control loop.

Next, based on (1), the total minimum bandwidth consumption [16] can be represented as

$$B_{min} = \sum_{i=1}^{n} \frac{m_i}{h_i} \quad (2)$$

To complete the formulation of the mathematical model of network congestion, we need the following definitions and assumption.

Definition 1. Primary Data/Primary Bandwidth
The primary data is the maximum amount of data transmitted through the network for a given bandwidth of each channel with no congestion. The primary bandwidth is the bandwidth which guarantees the stability of all control loops if there is no extra data transmitted [17].

Definition 2. Extra Data/Available Bandwidth
Data in excess of the primary data that will lead to network congestion/The available bandwidth which allocates B_{min} when extra data exists [17]. A certain quantity of data can be transmitted through the network for a given bandwidth allocation. However, additional data must be transmitted when corrective control action is required. In this case, the bandwidth capacity may be inadequate to transmit the additional data and the network becomes congested. The additional data is referred to as *extra data* in this paper.

Clearly, the amount of data transmitted is limited to an amount based on the primary bandwidth. However, different closed loop dynamics requires different primary bandwidths. Network congestion occurs when the primary bandwidth is insufficient to transmit the extra data [17].

Assumption: In the absence of network congestion, the designed controller stabilizes the network in the vicinity of its equilibrium at the zero state. In addition, the controller has no influence on the network characteristics [17].

As in [17], we represent network congestion with the linear time invariant system

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \quad (3)$$

where $\boldsymbol{x} = [x_1, x_2, \dots \dots \dots x_n]^T \in \mathcal{R}^n$ is the network congestion state and $\boldsymbol{u} = \left[u_1, u_2, , \dots, u_p\right]^T \in \mathcal{R}^p$ denotes the control signal, i.e. the bandwidth allocation for each individual loop.

According to the above assumption, when extra data is transmitted in a channel, network congestion arises. Similar to [17], let $d_i(t)$ denote the extra data sent over the i^{th} network channel, the congestion is proportional to the extra data with proportionality constant b_i.

Using (1), the congestion state equation can be represented similar to [17] as

$$\boldsymbol{x_c(t)} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \Lambda \boldsymbol{d(t)} = \Lambda \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} \quad (4)$$

where Λ is the matrix of channel bandwidths

$\ddot{\mathrm{E}} = diag\{b_1, \dots, b_n\} = \mathrm{diag}\{m_1/h_1, \dots, m_n/h_n\}$

Network congestion with model uncertainty can be represented as

$$\Lambda \dot{\boldsymbol{d}}\,(\boldsymbol{t}) = (A + \Delta_A)\Lambda \boldsymbol{d(t)} + (B + \Delta_B)\boldsymbol{u} \quad (5)$$

where $\Delta_A \; and \; \Delta_B$ are constant perturbation matrices.

The network congestion state is $\boldsymbol{x_c(t)} = \Lambda \boldsymbol{d(t)}$ and (5) can be rewritten as

$$\dot{\boldsymbol{x}}_{\boldsymbol{c}}(\boldsymbol{t}) = (A + \Delta_A)\boldsymbol{x_c(t)} + (B + \Delta_B)\boldsymbol{u} \quad (6)$$

The sensor and actuator are typically part of a digital control loop and a discrete-time model of the congestion model is required. The discrete-time network congestion model is

$$\boldsymbol{x_{c,k+1}} = A_c \boldsymbol{x_{c,k}} + B_c \boldsymbol{u_{c,k}}, \quad (7)$$

where

$$A_c = e^{(A+\Delta_A)T_S}, B_c = \int_0^{T_S} e^{(A+\Delta_A)(T_S-\tau)} B d\tau$$

with sampling period T_S.

Using the network congestion model (7), the optimal network control, i.e. optimal network bandwidth allocation, is studied in the next section.

III. MODEL FREE Q-LEARNING BASED OPTIMAL NETWORK CONGESTION CONTROL

In this section, the optimal network congestion control problem is studied. Since network congestion control is modeled as a discrete-time linear time-invariant system (7), we first review ideal optimal control for discrete-time linear systems. We then present a Q-learning based optimal design that can learn the optimal network congestion control even when the congestion model is unknown. The Q-learning approach applicable to network congestion control where modeling errors can be large.

A. Ideal Infinite Horizon Optimal Network control

$$\boldsymbol{x}_{d,k+1} = A_d \boldsymbol{x}_{d,k} + B_d \boldsymbol{u}_{d,k} \quad (8)$$

where $\boldsymbol{x}_{d,k}, \boldsymbol{u}_{d,k}$, are the system state and control input, respectively, and $A_{d,k} \in \mathcal{R}^{n\times n}$, $B_d \in \mathcal{R}^{n\times m}$ denote the system matrices. Using standard optimal control theory [6], we minimize the cost function

$$V^*(\boldsymbol{x}_{d,k}) = \min_{\boldsymbol{u}_{d,k}} \sum_{l=k}^{\infty} r(\boldsymbol{x}_{d,l}, \boldsymbol{u}_{d,l}) = \min_{\boldsymbol{u}_{d,k}} \sum_{l=k}^{\infty} (\boldsymbol{x}_{d,l}^{\boldsymbol{T}} Q_d \boldsymbol{x}_{d,l} + \boldsymbol{u}_{d,l}^{\boldsymbol{T}} R_d \boldsymbol{u}_{d,l}) \quad (9)$$

The cost-to-go is defined as

$r(\boldsymbol{x}_{d,k}, \boldsymbol{u}_{d,k}) = \boldsymbol{x}_{d,k}^{\boldsymbol{T}} Q_d \boldsymbol{x}_{d,k} + \boldsymbol{u}_{d,k}^{\boldsymbol{T}} R_d \boldsymbol{u}_{d,k}$, where Q_d is a symmetric positive semi definite matrix and R_d, S_N are symmetric positive definite matrices.

Using dynamic programming (DP), we write the Bellman equation in discrete-time as

$$0 = \min_{\boldsymbol{u}_{d,k}} r(\boldsymbol{x}_{d,k}, \boldsymbol{u}_{d,k}) + V^*(\boldsymbol{x}_{d,k+1}) - V^*(\boldsymbol{x}_{d,k}) \quad (10)$$

Assuming that the minimum on the right hand side of equation (10) exists and is unique, then the optimal control can be derived as [6]

$$\boldsymbol{u}_{d,k}^* = -\frac{1}{2} R_d^{-1} B_{d,k} \frac{\partial V^*(\boldsymbol{x}_{d,k+1})}{\partial \boldsymbol{x}_{d,k+1}} \quad (11)$$

Substituting the optimal control input (11) into the Bellman equation (10), gives the discrete-time (DT) Hamilton-Jacobi-Bellman (HJB) equation

$$0 = \boldsymbol{x}_{d,k}^{\boldsymbol{T}} Q_d \boldsymbol{x}_{d,k} + \frac{1}{4} \frac{\partial V^{*T}(\boldsymbol{x}_{d,k+1})}{\partial \boldsymbol{x}_{d,k+1}} B_d^T R_d^{-1} B_d \frac{\partial V^*(\boldsymbol{x}_{d,k+1})}{\partial \boldsymbol{x}_{d,k+1}} + V^*(\boldsymbol{x}_{d,k+1}) - V^*(\boldsymbol{x}_{d,k}) \quad (12)$$

For linear systems, the value function (9) can be formulated as a quadratic function of the state vector and can be expressed as [6]

$V^*(\boldsymbol{x}_{d,k})=\boldsymbol{x}_{d,k}^{\boldsymbol{T}} P \boldsymbol{x}_{d,k}$ (13)

with $P_k, \forall k = 0, 1, \ldots, N$, positive definite kernel matrices. Substituting (13) into (2), the DT HJB reduces to the well-known Algbra Riccati equation (ARE)

$0 = A_d^T [P - PB_d (B_d^T PB_d + R_d)^{-1} B_d^T P] A_d + Q_d - P$ (14)

Moreover, the optimal control input can be represented in terms of the ARE solution P as

$\boldsymbol{u}_k^* = -(B_d PB_d + R_d)^{-1} B_d^T PA_d \boldsymbol{x}_{d,k}$ (15)

When the system matrices are uncertain, the ARE solution cannot be found. Additionally, generating the control input in a forward-in-time manner has significant practical advantages for hardware implementation over traditional optimal control. For the network congestion model, the system matrices are uncertain due to the complex network environment. To overcome these challenges, a model-free Q-learning algorithm is adopted.

B. Q-learning Based Optimal Congestion Control

The main objective of Q-learning is to use the system states and inputs to learn the optimal control in the presence of model uncertainties. Since the inputs and states of the real-time network congestion system include the effects of system uncertainties, it is possible to use them to accomplish this goal. The proposed Q-learning design is also more practical because it uses a forward-in-time optimization strategy. Next, the details of Q-learning are given.

Using the DT HJB gives the Q-function

$$Q^*(\boldsymbol{x}_k, \boldsymbol{u}_k) = r(\boldsymbol{x}_k, \boldsymbol{u}_k) + V^*(\boldsymbol{x}_{k+1}) = [\boldsymbol{x}_k^T \; \boldsymbol{u}_k^T] H [\boldsymbol{x}_k \; \boldsymbol{u}_k]^T$$
$$= \boldsymbol{x}_k^T Q \boldsymbol{x}_k + \boldsymbol{u}_k^T R \boldsymbol{u}_k + \boldsymbol{x}_{k+1}^T P \boldsymbol{x}_{k+1} \quad (16)$$

with $r(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{x}_k^T Q \boldsymbol{x}_k + \boldsymbol{u}_k^T R \boldsymbol{u}_k$.

When $\boldsymbol{u}_k$ is the optimal control, the optimal value function is $V^*(\boldsymbol{x}_k) = Q^*(\boldsymbol{x}_k, \boldsymbol{u}_k)$

Equation (16) can be written as

$Q^*(\boldsymbol{x}_k, \boldsymbol{u}_k) = r(\boldsymbol{x}_k, \boldsymbol{u}_k) + Q^*(\boldsymbol{x}_{k+1})$

Using the definition of $r(\boldsymbol{x}_k, \boldsymbol{u}_k)$ and the network congestion system model (7), we rewrite (15) in terms a matrix H as

$$[\boldsymbol{x}_k^T \; \boldsymbol{u}_k^T] H \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{bmatrix} = [\boldsymbol{x}_k^T \; \boldsymbol{u}_k^T] \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{bmatrix} +$$
$$[\boldsymbol{x}_k^T \; \boldsymbol{u}_k^T] \begin{bmatrix} A^T \\ B^T \end{bmatrix} P \begin{bmatrix} A^T \\ B^T \end{bmatrix}^T \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{u}_k \end{bmatrix} \quad (17)$$

with

$$H = \begin{bmatrix} H_{xx} & H_{xy} \\ H_{yx} & H_{yy} \end{bmatrix} = \begin{bmatrix} A^T PA + R & A^T PB \\ B^T PA & B^T PB + I \end{bmatrix} \quad (18)$$

Recalling the ideal optimal control given in (15), the optimal policy can also be represented in terms of H as $\boldsymbol{u}_k = L\boldsymbol{x}_k = -H_{yy}^{-1} H_{yx} \boldsymbol{x}_k$. Similar to [1], the relationship between P and H can be formulated as

$$P = [I \; L^T] H \begin{bmatrix} I \\ L^T \end{bmatrix} \quad (19)$$

Substituting (19) into (17), we obtain the matrix

$$H = G + \begin{bmatrix} A & LA \\ B & LB \end{bmatrix} H \begin{bmatrix} A & B \\ LA & LB \end{bmatrix} \quad (20)$$

where

$G = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$.

Note that the Q-function and H matrix are critical for solving the optimal network congestion control problem. However, they cannot be obtained directly from (17). To learn the ideal Q-function and the associated H matrix, the following update policy [1] is used

$$Q_{i+1}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{x}_k^T Q_i \boldsymbol{x}_k + \boldsymbol{u}_i^T(\boldsymbol{x}_k) \boldsymbol{u}_i(\boldsymbol{x}_k) +$$
$$[\boldsymbol{x}_{k+1}^T \; \boldsymbol{u}_i^T(\boldsymbol{x}_{k+1})] H_i \begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{u}_i(\boldsymbol{x}_{k+1}) \end{bmatrix} \quad (21)$$

where Q_i and H_i are estimated Q-function and H matrix at iteration i. L_i is also updated based on the updated matrix H_i ($L_i = -H_{yy,i}^{-1} H_{yx,i}$).

Specifically, the Q-function can be written as the adaptive estimator [1]

$\hat{Q}(\bar{z}, \boldsymbol{h_i}) = z^T H_i z = \boldsymbol{h_i^T} \bar{z}$, $z \in n + m_1 + m_2 = q$ (22)

where $z = [x^T \; u^T]^T$, and

$\bar{z} = (z_1^2, \ldots, z_1 z_q, z_2^2, z_2 z_3 \ldots\ldots, z_{q-1} z_q, z_q^2)$ is the Kronecker product quadratic polynomial basis function. $h = v(H)$ is the desired parameters, $v(.)$ is a vector function that gives a $q(q+1)/2 \times 1$ vector from a $q \times q$ matrix [1]. This exploits the symmetry of the Q-function so that only the upper triangle terms of the matrix is used in the calculations.

Substituting (22) into (21) we have

$\boldsymbol{h_{i+1}^T} \bar{z}(\boldsymbol{x_k}) = d(z_k(\boldsymbol{x_k}), \boldsymbol{h_i})$ (23)

The desired target function is defined as

$d(z_k(\boldsymbol{x_k}), H_i) = \boldsymbol{x_k^T} Q \boldsymbol{x_k} + \widehat{\boldsymbol{u_i}}(\boldsymbol{x_k})^T \widehat{\boldsymbol{u_i}}(\boldsymbol{x_k}) + Q(\boldsymbol{x_{k+1}}, \boldsymbol{u_{k+1}})$ (24)

$\boldsymbol{h_{i+1}}$ can be found by minimizing the square error between the target value function in (23) and (24) [1]. Using least squares, the H matrix can be updated as

$$\boldsymbol{h_{i+1}} = \arg\min_{h_{i+1}} \left\{ \int_\Omega \left| \boldsymbol{h_{i+1}^T} \bar{z}(\boldsymbol{x_k}) - d(z_k(\boldsymbol{x_k}), \boldsymbol{h_i}) \right|^2 d\boldsymbol{x}_k \right\} \quad (25)$$

$$\boldsymbol{h_{i+1}} = \arg\min_{h_{i+1}} \left\{ \int_\Omega |\bar{z}(\boldsymbol{x_k}) \bar{z}(\boldsymbol{x_k})^T dx - d(z_k(\boldsymbol{x_k}), \boldsymbol{h_i})|^2 d\boldsymbol{x}_k \right\} \quad (26)$$

$$\boldsymbol{h_{i+1}} = \left(\int_\Omega \bar{z}(\boldsymbol{x_k}) \bar{z}(\boldsymbol{x_k})^T dx \right)^{-1} \int_\Omega d(\bar{z}(\boldsymbol{x_k})^T, \boldsymbol{h_i}) \, d\boldsymbol{x} \quad (27)$$

$$z(\boldsymbol{x_k}) = \left[\boldsymbol{x_k^T} (\widehat{\boldsymbol{u}}_{\boldsymbol{i}}(\boldsymbol{x_k})^{\boldsymbol{T}}) \right]^T = \left(\boldsymbol{x_k^T} [I \; L_i^T]^T \right)^T \quad (28)$$

$\int_\Omega \bar{z}(\boldsymbol{x_k}) \bar{z}(\boldsymbol{x_k})^T d\boldsymbol{x_k}$ is not invertible, so a very small zero mean noise term (n_{1k}) can be added to make it invertible [1].

Thus, $\widehat{\boldsymbol{u}}_{\boldsymbol{ei}}(\boldsymbol{x_k}) = L_i \boldsymbol{x_k} + \boldsymbol{n_{1k}}$ and equation (28) becomes

$$z_k = \begin{bmatrix} \boldsymbol{x_k} \\ \widehat{\boldsymbol{u}}_{\boldsymbol{ei}}(\boldsymbol{x_k}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x_k} \\ L\boldsymbol{x_k} \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{n_{1k}} \end{bmatrix}$$

Using a sufficient number of points, we have an over determined system whose minimum least squares solution is [1]

$\boldsymbol{h_{i+1}} = (ZZ^T)^{-1} Z\boldsymbol{Y}$

where

$Z = [\bar{z}(\boldsymbol{x_{k-N-1}}), \bar{z}(\boldsymbol{x_{k-N-2}}), \ldots\ldots, \bar{z}(\boldsymbol{x_{k-1}})]$

$$\boldsymbol{Y} = [\boldsymbol{d}(\bar{\boldsymbol{z}}(\boldsymbol{x}_{\boldsymbol{k-N-1}}), \boldsymbol{h}_{\boldsymbol{i}})\ \boldsymbol{d}(\bar{\boldsymbol{z}}(\boldsymbol{x}_{\boldsymbol{k-N-2}}), \boldsymbol{h}_{\boldsymbol{i}})\ \dots\ \boldsymbol{d}(\bar{\boldsymbol{z}}(\boldsymbol{x}_{\boldsymbol{k-1}}), \boldsymbol{h}_{\boldsymbol{i}})]^T$$

Finally the target equation (24) becomes

$$d(z_k(\boldsymbol{x}_{\boldsymbol{k}}), H_i) = \boldsymbol{x}_{\boldsymbol{k}}^T Q \boldsymbol{x}_{\boldsymbol{k}} + \widehat{\boldsymbol{u}_{\boldsymbol{ei}}}(\boldsymbol{x}_{\boldsymbol{k}})^T \widehat{\boldsymbol{u}_{\boldsymbol{ei}}} + Q(\boldsymbol{x}_{\boldsymbol{k+1}}, \widehat{\boldsymbol{u}}_{\boldsymbol{i}}(\boldsymbol{x}_{\boldsymbol{k+1}}))$$

IV. RESULTS AND DISCUSSION

We adopt the discrete network congestion model [17]

$$\Lambda \boldsymbol{d}(k+1) = A\Lambda \boldsymbol{d}(k) + B\boldsymbol{u}(k)$$

where $\Lambda = \text{diag}\{b_1, \dots, b_n\} = diag\left\{\frac{m_1}{h_1}, \dots, \frac{m_n}{h_n}\right\}$. Without loss of generality, we let the matrix Λ be the identity matrix. Then the discrete LTI system becomes

$$\boldsymbol{d}(k+1) = A\boldsymbol{d}(k) + B\boldsymbol{u}(k)$$

where $\boldsymbol{d}(k)$ is the congestion state.

We discretize the model of [17] with $T_s = 0.01$s and obtain the matrices

$$A = \begin{bmatrix} 1.0001 & 0.0202 & 0.0098 \\ 0.0100 & 1.0003 & 0.0098 \\ 0.0002 & 0.0388 & 0.9420 \end{bmatrix},\quad B = \begin{bmatrix} 0.0004 \\ 0.0301 \\ 0.0103 \end{bmatrix}$$

$R = 1, Q = I$

with initial state $d_0 = [0.2,1,0.5]$. Then the LQR gain K can be calculated using MATLAB as [6]

$K = [\ 1.3180 \quad 1.8695 \quad 0.4619]$.

In practice, the correct model is unknown and the model parameters can only be obtained approximately. We assume the matrices obtained from experimental data describing the local behavior of the network be

$$A = \begin{bmatrix} 1.035 & 0.029 & 0.017 \\ 0.0198 & 1.058 & 0.00158 \\ 0.0003657 & 0.0446 & 0.9546 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0024 \\ 0.0490 \\ 0.0390 \end{bmatrix}$$

Because any mathematical model of network congestion is approximate, LQR will not be able to eliminate the congestion in the presence of modeling errors. Q-learning is an appropriate strategy for optimizing the network because, unlike LQR, it does not require an exact mathematical model. All it requires is knowledge of the matrix H of (18). Thus, Q-learning is able to eliminate the congestion in the presence of model uncertainties.

We present simulation results that demonstrate the effectiveness of Q-learning in eliminating congestion in the presence of modeling uncertainties. The time response for LQR both with the true and the experimentally evaluated network congestion are shown together with the response for Q-learning in Figure 2-4. The figures show that for all three state variables Q-learning eliminates the congestion in each states in spite of the modeling errors while LQR fails to do so.Figure 5-7 show the absolute error in following the ideal LQR response for state variables x_1, x_2, and x_3, respectively. The Q-Learning response converges to the ideal LQR response after about 3 s while LQR with the perturbed model exhibits a large error and fails to converge. The root mean square error for LQR is 1.2437, while that for Q-learning is only 0.0941. Thus, Q-Learning results in the elimination of network congestion while LQR fails to do so in the presence of modeling errors. After eliminating the network congestion, each loop uses its primary bandwidth that is selected based on its characteristics [17].

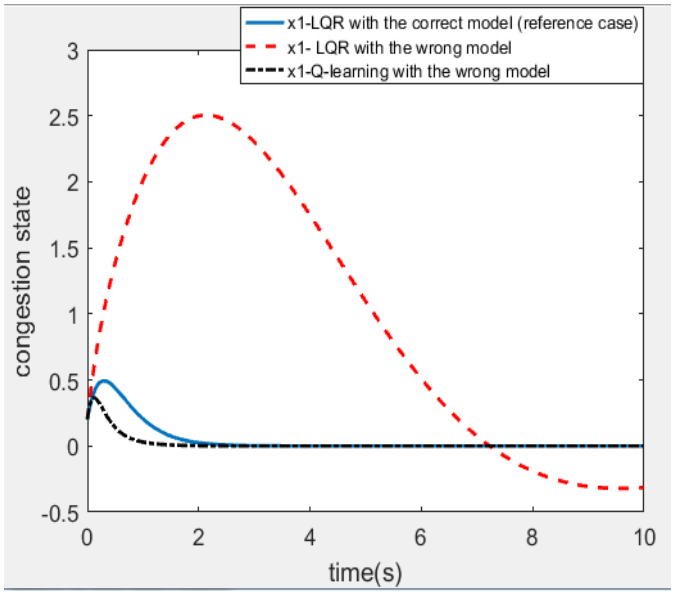

Figure 2 Congestion state $\boldsymbol{x_1}$ of LQR and Q-learning

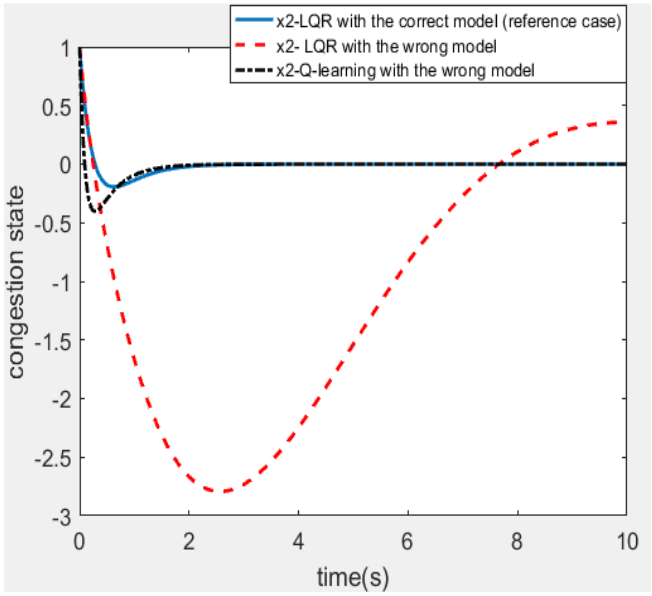

Figure 3 Congestion state $\boldsymbol{x_2}$ of LQR and Q-learning

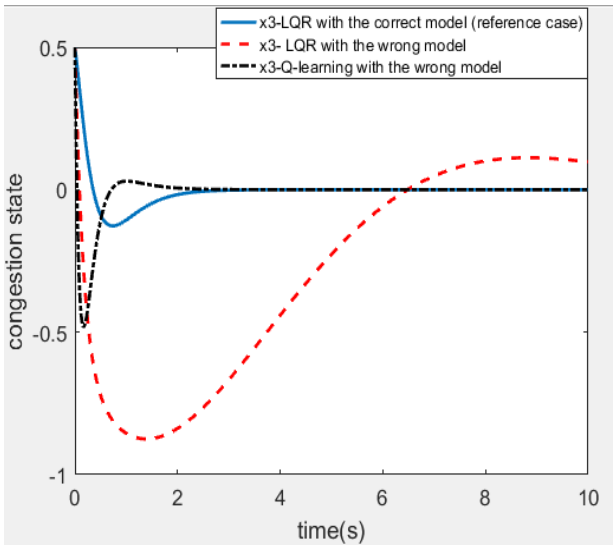

Figure 4 Congestion state x_3 of LQR and Q-learning

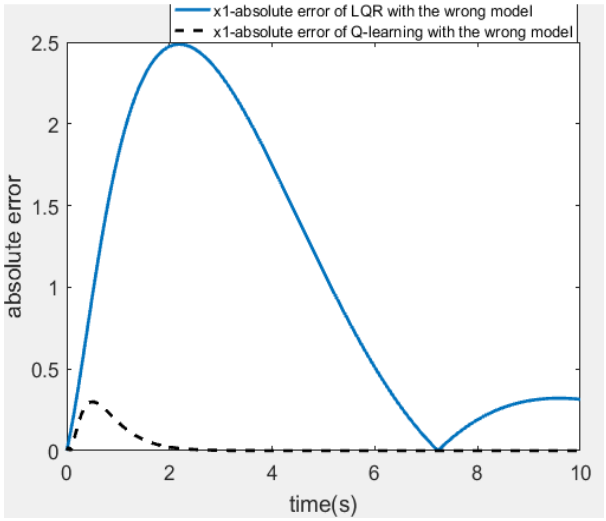

Figure 5 Absolute errors in x_1 for LQR and Q-Learning

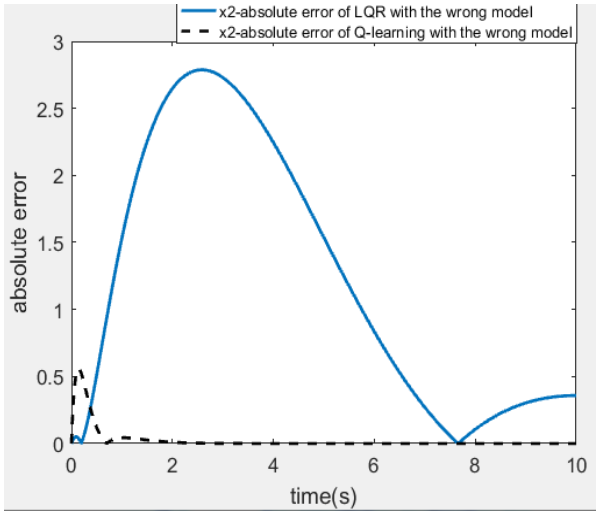

Figure 6 Absolute errors in x_2 for LQR and Q-Learning

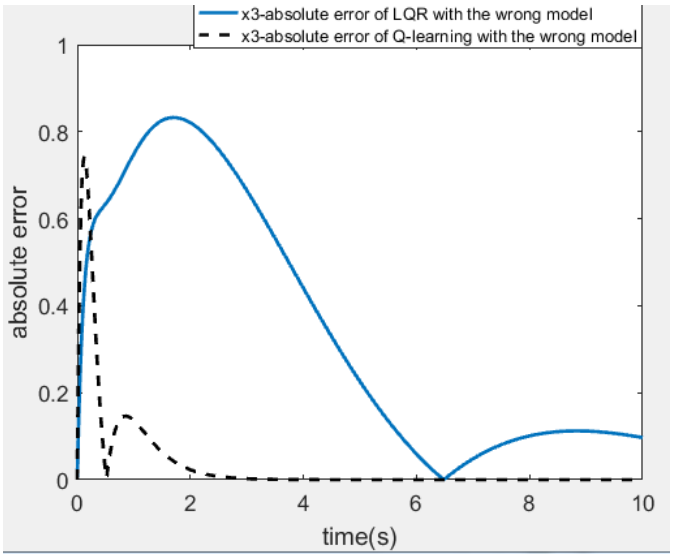

Figure 7 Absolute errors in x_3 for LQR and Q-Learning

V. CONCLUSION

This paper verifies the validity of the hypothesis that Q-learning can provide optimal resource allocation for CPS. To test the hypothesis, we used a simple state-space model to represent local system dynamics and LQR as the reference optimal behavior. We perturbed the nominal model and simulated it, first with LQR then with Q-learning. Our results show that Q-learning provides performance that approaches that of the ideal LQR control whereas LQR control with modeling errors cannot. In addition, Q-learning is a model-free approach and does not require knowledge of the mathematical model of the system; the latter is very difficult to obtain for CPS. In addition, Q-learning is feasible for more complicated system dynamics including nonlinear and time-varying behavior. Future work will examine Q-learning with a NS3 model of a communication network in a CPS.

REFERENCES

[1] A. Al-Tamimi, F. L. Lewis and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control." *AUTOMATICA, 473-481,* 2007.

[2] V. Gunes, S. Peter, T. Givargis and F. Vahid, "A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems", *KSII Transactions On Internet And Information Systems*,Vol. 8, NO. 12, December 2014.

[3] Y. Halevi and A. Ray, "integrated communication and control systems: Part I - Analysis," *J. Dynamic Systems, Measurement, and Control,* Vol. 120, PP. 367-373, 1988.

[4] S. Haykin, Neural Networks and Learning Machines, 3rd Ed., *Pearson, Prentice-Hall, Upper Saddle River*, NJ, 2009.

[5] K. Lakshmanan, D. de Niz , R. Rajkumar and G. Moreno, "Resource Allocation in Distributed Mixed-Critically Cyber-

Physical Systems", *Internat. Conf. Distributed Computing Systems*, Genoa, Italy, June 2010.

[6] F. L. Lewis, D. Vrabie, and V. L. Syrmos, Optimal control, 3rd edition, *Wiley, Hoboken*, NJ, 2012.

[7] J. K. Mendiratta, M. Brindha,' Networked Control System – A Survey', *Internat. J. Modern Education and Computer Science*, Vol. 6, 42-48, 2013.

[8] C. Peng, D. Yue, Z. Gu ,F. Xia, "Sampling period scheduling of networked control systems with multiple-control loops", *Mathematics and Computers in Simulation,* Vol. 79, No. 5, PP 1502-1511, 2009.

[9] J. Shi, J. Wan, H. Yan and H. Suo, "A Survey of Cyber-Physical Systems" . In *Proc. of the Int. Conf. on Wireless Communications and Signal Processing*, Nanjing, China, November 9-11, 2011.

[10] M. Umer Tariq, J. Florence, and M. Wolf, "Design Specifications of Cyber-Physical Systems towards a Domain-Specific Modelling Language Based on Simulink, Eclipse Modelling Framework and Giotto", *7th Internat. Workshop Model-Based Architecting & Construction Embedded Systems,* Valencia, Spain, Sep 2014.

[11] M. Velasco, J .M. Fuertes, C. Lin, P. Marti, S. Brandt "A Control Approach to Bandwidth Management in Networked Control Systems.", *Technical report*, UCSC-CRL-04-10

[12] J. Wan , H. Yan , H. Suo and F. Li, "Advances in Cyber-Physical Systems Research" , *KSII Trans. Internet and Information Systems* VOL. 5, NO. 11, November 2011.

[13] F .Y. Wang and D. Liu, Networked Control Systems: Theory and Applications, SPRINGER, 2008.

[14] B.Y. Xia, M. Fu and G.P. Liu, Analysis and Synthesis of Networked Control Systems*, Springer,* 2011.

[15] F. Xia , L. Ma , J. Dong and Y. Sun, " Network QoS Management in Cyber-Physical Systems", *Internat. Conf. Embedded Software & Systems Symposia*, Chengdu, Sichuan, China, July, 2008.

[16] W. Zhiwen and S. Hongtao , "A bandwidth allocation strategy based on the proportion of measurement error in networked control system", *in Proceedings of the 3rd IEEE International Conference Digital Manufacturing & Automation,* Changcha, Hunan, China, pp. 9–12, Dec 2010.

[17] W. Zhiwen and S. Hongtao, "Control and optimization of network in networked control system" *Mathematical Problems Engineering,* Vol. 2014, art.ID 237372, 2014.