Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

# Quantized measurements in Q-learning based model-free optimal control

**Sini Tiistola*, Risto Ritala*, Matti Vilkko***

* *Faculty of Engineering Sciences, Tampere University, Tampere, Finland (e-mail: sini.tiistola@tuni.fi, risto.ritala@tuni.fi, matti.vilkko@tuni.fi)*

**Abstract:** Quantization noise is present in many real-time applications due to the resolution of analog-to-digital conversions. This can lead to error in policies that are learned by model-free Q-learning. A method for quantization error reduction for Q-learning algorithms is developed using the sample time and an exploration noise that is added to the control input. The method is illustrated with discrete-time policy and value iteration algorithms using both a simulated environment and a real-time physical system.

*Keywords:* Algebraic Riccati Equations, Discrete time, LQR control, Model-free, Optimal control, Quantized signals, Q-learning

## 1. INTRODUCTION

Q-learning is a reinforcement learning method, originally developed by Watkins (1989), that observes system responses to given actions and uses this data to learn an optimal policy. Recent studies have used Q-learning to learn the model-free optimal feedback control in simulated environments (e.g. Lewis et al., 2012; Rizvi and Lin, 2017). Q-learning is studied in both continuous time problems and discrete-time problems. Q-learning and recent literature is reviewed in more detail e.g. by Rizvi and Lin (2017) and Kiumarsi et al (2014). However, only few Q-learning studies use data from real-time applications (e.g. ten Hagen and Kröse, 2003; Radac and Precup, 2018). This study focuses on discrete-time systems and real-time Q-learning applications.

It is known, that the analog-to-digital conversion causes quantization error in the measurements (Bennett, 1948; Gray and Neuhoff, 1998). Multiple studies are conducted on model-based control and system identification with quantized measurements (Curry, 1970; Delchamps, 1990; Wang et al., 2010). Schoukens et al. (1988) and Roinila et al. (2010) among others, study excitation signals in identification applications. According to them, identification results could be improved with an excitation signal that has a large amplitude to yield larger signal-to-noise-ratio or by choosing the excitation within the frequency range of the system. However, quantization in model-free optimal control has not been widely studied yet. Zhao et al. (2015) have studied quantization in finite horizon optimization problem. They model the quantization error into the Q-learning algorithm and solve model-free the optimal control problem.

In this paper, the infinite horizon optimal control problem is solved using quantized control input and measurements. A new method for quantization error reduction is developed using only the exploration noise and the sample time. The rest of the paper is organized as follows. Q-learning for partially observable linear systems is reviewed in Section 2. The physical system studied, Quanser QUBE-Servo 2, is presented in Section 3. In Section 4, the connection between the quantization error, the exploration noise and the sample time is studied and a method for reducing the quantization error is developed. The method is illustrated both in a simulated and a real-time environment in Section 5. Conclusions are given in Section 6.

## 2. Q-LEARNING FOR PARTIALLY OBSERVABLE LINEAR SYSTEMS

Linear time-invariant system model is given by Franklin et al. (1998) as

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k + Du_k \end{cases}, \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$, and $y_k \in \mathbb{R}^{n_y}$ are the state, the control input and the output at time $k$ and $n_x$, $n_u$, and $n_y$ are the number of states, inputs and outputs. Matrices $A, B, C,$ and $D$ are the state transition, the input, the output, and the feedthrough matrices of appropriate dimensions. Lewis and Vamvoudakis (2011) and Rizvi and Lin (2017, 2019) denote a partially measurable state $x_k$ as

$$x_k = [M_u \quad M_y]\bar{x}_k. \tag{2}$$

The new state $\bar{x}_k$ is formed from a vectors $\bar{u}_k$ and $\bar{y}_k$ containing old controls and old measurements as

$$\begin{aligned} \bar{x}_k &= [\bar{u}_k^T \quad \bar{y}_k^T]^T \\ \bar{u}_k &= [\, u_{k-1} \quad u_{k-2} \quad \dots \quad u_{k-n}]^T. \\ \bar{y}_k &= [\, y_{k-1} \quad y_{k-2} \quad \dots \quad y_{k-n}]^T \end{aligned} \tag{3}$$

where $n \leq n_x$ is the observability index. Matrices $M_u$ and $M_y$ in (2) are defined using the observability, controllability and Toeplizt matrices $V_n$, $U_n$ and $T_n$ as

$$\begin{aligned} &M_y = A^n(V_n^T V_n)^{-1} V_n^T, \qquad M_u = U_n - M_y T_n \\ &V_n = [(CA^{n-1})^T \quad \cdots \quad (CA)^T \quad C^T]^T \\ &U_n = [B \quad AB \quad \cdots \quad A^{n-1}B] \end{aligned} \tag{4a}$$

$$T_n = \begin{bmatrix} 0 & CB & CAB & \cdots & CA^{n-2}B \\ 0 & 0 & CB & \cdots & CA^{n-3}B \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & CB \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4b)$$

### *2.1 Optimal model-based control*

Originating from the Bellman's optimality principle (Bellman, 1957), the optimal control for system (1) is the control that minimizes the Hamilton-Jacobi-Bellman (HJB) equation. It is given by Sutton et al. (2018) and Lewis et al. (2012) as

$$V^*(x_k) = \min_{u_k}\big(r(x_k, u_k) + V^*(x_{k+1})\big), \quad (5)$$

where and the one-step cost $r(x_k, u_k)$ is given as

$$r(x_k, u_k) = {x_k}^T Q x_k + {u_k}^T R u_k = {y_k}^T Q_y y_k + {u_k}^T R u_k, \quad (6)$$

where $Q, Q_y$ and $R$ are the state, the output and the control weighting matrices of appropriate dimensions and $Q = C^T Q_y C$ and $u_k = h(x_k)$ is the control policy.

Assuming the system (1) is controllable, Franklin et al. (1998) solve the optimal control policy as

$$u_k = h(x_k) = K^* x_k, \quad (7)$$

where $K^* \in \mathbb{R}^{n_u \times n_x}$ is the optimal control gain matrix given as

$$K^* = -(R + B^T X B)^{-1} B^T X A, \quad (8)$$

and $X$ is the algebraic Riccati equation solution

$$X = A^T X A - A^T X B (R + B^T X B)^{-1} B^T X A + Q. \quad (9)$$

### *2.2 Q-learning for partially observable linear systems*

According to Watkins (1989) the optimal Q-function can be defined for any optimal control problem as

$$Q^*(x_k, u_k) = r(x_k, u_k) + V^*(x_{k+1}). \quad (10)$$

Inserting (10) into (5) shows that the optimal policy also minimizes the optimal Q-function. The resulting equation can be inserted back into (10). It yields the Q-Bellman optimality equation

$$Q^*(x_k, u_k) = r(x_k, u_k) + \min_{u_k}\big(Q^*(x_{k+1}, u_{k+1})\big). \quad (11)$$

The Q-function for linear systems is solved by Lewis and Vamvoudakis (2011) and Rizvi and Lin (2017, 2019) as

$$Q(\bar{x}_k, u_k) = {\bar{z}_k}^T T \bar{z}_k = r(y_k, u_k) + {\bar{z}_{k+1}}^T T \bar{z}_{k+1}, \quad (12)$$

$$W^T \phi(\bar{z}_k) = r(y_k, u_k) + W^T \phi(\bar{z}_{k+1}), \quad (13)$$

where $r(y_k, u_k)$ is given in (6) and $\bar{z}_k$ is formed with the new state $\bar{x}_k$ in (3) and with the current policy $u_k$ as

$$\bar{z}_k = \begin{bmatrix} \bar{x}_k \\ u_k \end{bmatrix} \in \mathbb{R}^{n_{\bar{z}}} \text{ and } n_{\bar{z}} = n(n_u + n_y) + n_u. \quad (14)$$

The symmetric kernel matrix $T$ in (12) is defined as

$$T = \begin{bmatrix} M_u^T & 0 \\ M_y^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A^T X A + Q & A^T X B \\ B^T X A & B^T X B + R \end{bmatrix} \begin{bmatrix} M_u & M_y & 0 \\ 0 & 0 & I \end{bmatrix}$$
$$= \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1n_{\bar{z}}} \\ t_{21} & t_{22} & & t_{2n_{\bar{z}}} \\ \vdots & & \ddots & \vdots \\ t_{n_{\bar{z}}1} & t_{n_{\bar{z}}2} & \cdots & t_{n_{\bar{z}}n_{\bar{z}}} \end{bmatrix} = \begin{bmatrix} T_{\bar{u}\bar{u}} & T_{\bar{u}\bar{y}} & T_{\bar{u}u} \\ T_{\bar{y}\bar{u}} & T_{\bar{y}\bar{y}} & T_{\bar{y}u} \\ T_{u\bar{u}} & T_{u\bar{y}} & T_{uu} \end{bmatrix}. \quad (15)$$

where $T_{uu} \in \mathbb{R}^{n_u x n_u}, T_{\bar{u}\bar{u}}, T_{\bar{u}\bar{y}}, T_{\bar{y}\bar{y}} \in \mathbb{R}^{nxn}, T_{\bar{u}u} \in \mathbb{R}^{nxn_u}$ and $T_{\bar{y}u} \in \mathbb{R}^{nxn_y}$ are matrix elements of $T$. The matrix $T$ is expressed in (13) in a vector form using its scalar elements as

$$W = \big[t_{11}, 2t_{12}, \cdots, 2t_{1n_{\bar{z}}}, t_{22}, \cdots, 2t_{2n_z}, \cdots, t_{n_{\bar{z}}n_{\bar{z}}}\big]^T. \quad (16)$$

The quadratic basis vector $\phi(\bar{z}_k) \in \mathbb{R}^{(n_{\bar{z}}(n_{\bar{z}}+1)/2)}$ is given as

$$\phi(\bar{z}_k) = \big[\bar{z}_1^2, \bar{z}_1\bar{z}_2, \cdots, \bar{z}_1\bar{z}_{n_{\bar{z}}}, {\bar{z}_2}^2, \bar{z}_2\bar{z}_3, \cdots \bar{z}_2\bar{z}_{n_{\bar{z}}}, \cdots, \bar{z}_{n_{\bar{z}}}^2\big]^T, \quad (17)$$

where $\bar{z}_m$ is the $m^{\text{th}}$ element of $\bar{z}_k$.

Rizvi and Lin (2017, 2019) solve the model-free discrete-time Linear Quadratic Regulator (LQR) problem by identifying the kernel matrix $T$ in (15) from data using (12) or (13). The optimal policy that minimizes (12) is derived as

$$u_k = -(T_{uu})^{-1}[T_{u\bar{u}} \quad T_{u\bar{y}}]\, \bar{x}_k. \quad (18)$$

This is equal to inserting (2) and (8) into (7).

### *2.3 Policy and value iteration to solve the LQR problem*

Policy and value iteration (PI and VI) algorithms run policy and value updates until the optimal policy is found. They are initialized at $j = 0$. The initial policy is chosen randomly, but it must be stabilizing for PI. Here, the value update step updates the weight matrix $W$. Lewis et al. (2012) denote the weight matrix $W$ in (16) as $\widehat{W}_{j+1}$so that

$$\widehat{W}_{j+1}^T \varphi_k = \mu_k. \quad (19)$$

where the data and the regression vector $\mu_k$ and $\varphi_k$ are given for PI as

$$\begin{matrix} \varphi_k = \phi(\bar{z}_k) - \phi(\bar{z}_{k+1}) \\ \mu_k = r(y_k, u_k) \end{matrix}. \quad (20)$$

and for VI using the old weight matrix $\widehat{W}_j$ as

$$\begin{matrix} \varphi_k = \phi(\bar{z}_k) \\ \mu_k = r(y_k, u_k) + \widehat{W}_j^T \phi(\bar{z}_{k+1}) \end{matrix}. \quad (21)$$

The weight $\widehat{W}_{j+1}$ is updated with recursive least squares (RLS) and the policy update step updates the policy $u_{j+1,k}$ with (18). The updated control policy is applied in the system during learning with an added exploration noise $\epsilon_k$. Lewis et al. (2012), among others, add it to the control input to ensure the persistence of excitation (PE) condition and the convergence of the kernel matrix $\widehat{T}_{j+1}$. The value and policy updates are repeated until the weight $\widehat{W}_{j+1}$ convergences so that $\|\widehat{W}_{j+1} -$

$\|\hat{W}_j\| \leq \varepsilon_j$, where $\varepsilon_j$ is a small constant. Lewis and Vamvoudakis (2011) use a discounting factor $\gamma$ in (5) to reduce the noise bias effects, but Rizvi and Lin (2017) prove that choosing $\gamma < 1$ does not guarantee the closed-loop system stability so therefore discounting is not used here.

### *2.4 Recursive least squares value update*

Franklin, et al (1998) define the one-step RLS update as

$$\begin{aligned} L_{i+1} &= \lambda^{-1}P_i\varphi_k(a^{-1} + \lambda^{-1}\varphi_k^T P_i \varphi_k)^{-1} \\ \hat{W}_{j+1,i+1} &= \hat{W}_{j+1,i} + L_{i+1}(\mu_k - \varphi_k^T \hat{W}_{j+1,i}), \\ P_{i+1} &= \lambda^{-1}(I - L_{i+1}\varphi_k^T)P_i \end{aligned} \quad (22)$$

where $L_{i+1}$, $P_{i+1}$ and $\hat{W}_{j+1,i+1}$ are the update, the covariance and the weight matrices at index $i+1$ and $\lambda$ is a RLS discounting factor. For regular RLS $\lambda = 1$ and $a = 1$.

For the value update, the index $i$ is set as $i = 0$ and the weight $\hat{W}_{j+1,i}$ is set as the previous weight $\hat{W}_j$ and the covariance matrix $P_0$ is set as $P_0 = \delta I$, where $\delta$ is a large scalar (Franklin et al., 1998). Each update step $i$ the weight $\hat{W}_{j+1,i+1}$ is updated with (22) using new measurements. The regression $\varphi_k$ and data vector $\mu_k$ are formed from the measured data using (20) or (21) and (3), (14) and $u_{k+1}$is calculated with (18) using the current policy. Updates stop when the weight $\hat{W}_{j+1}$ has converged so that $\|\hat{W}_{j+1,i+1} - \hat{W}_{j+1,i}\| \leq \varepsilon_i$, where $\varepsilon_i$ is a small constant. (Rizvi and Lin, 2017, 2019).

## 3. THE QUANSER QUBE-SERVO 2 SYSTEM

Apkarian et al. (2016) derive a continuous-time system model for the Quanser QUBE-Servo 2 experiment (Fig. 1). The state $x$ is chosen as $x = [\theta \quad \omega]^T$, where $\theta$ is the angular position of the disk load $(rad)$ and $\omega = \dot{\theta}$ is the angular velocity $(rad/s)$. With voltage as the input $u$ and angular position $\theta$ as the output $y$, the marginally stable system model with numerical parameters is

$$\begin{cases} \dot{x} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -10.0485 \end{bmatrix} x + \begin{bmatrix} 0 \\ 239.2509 \end{bmatrix} u \\ y = [1 \quad 0]x \end{cases}. \quad (23)$$

Block diagram in Fig. 1 shows the system connected to the computer. The data acquisition device (DAQ) works as an interface and it communicates via MATLAB. The DAQ uses an encoder and a decoder to process data. The angular position $\theta$ is measured as encoder counts using a chosen sample time $dt$ $(s)$. The quadrature decoder generates 2048 counts per revolution (Apkarian et al, 2016). Therefore, one count corresponds to $2\pi$ /2048 $rad \approx 0.0031\ rad$. This resolution leads to uniformly quantized output. Xu et al. (2015) and Zhao et al. (2015) define the quantized output $y_k^q$ and control $u_{qk}^q$ as

$$\begin{aligned} y_k^q &= q(y_k) = \Delta_\theta \cdot (\lfloor y_k/\Delta_\theta \rfloor + 1/2) \\ u_{qk}^q &= q(u_{qk}) = \Delta_u \cdot (\lfloor u_{qk}/\Delta_u \rfloor + 1/2)' \end{aligned} \quad (24)$$

where $\lfloor y_k/\Delta_\theta \rfloor$ and $\lfloor u_{qk}/\Delta_V \rfloor$ are floor functions of $y_k/\Delta_\theta$ and $u_{qk}/\Delta_u$ and the output quantization interval $\Delta_\theta$ is

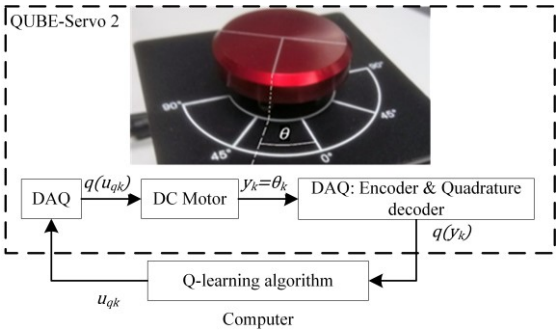


Fig. 1. Block diagram of the Quanser QUBE-Servo 2 system

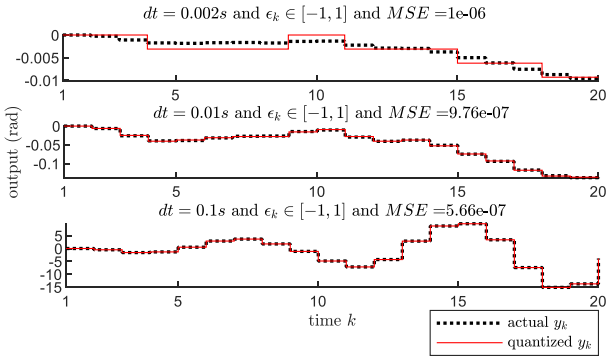


Fig. 2. Sample time and quantization

$\Delta_\theta = 0.0031\ rad$. For simulations, the input quantization interval $\Delta_u$ was chosen as $\Delta_u = 0.0001\ V$. Quantized control $u_{qk}$ in (24) and Fig.1 is computed with (18) using a quantized state $\bar{x}_{qk} = [\bar{u}_{qk}^T \quad \bar{y}_{qk}^T]^T$, where $\bar{u}_{qk}$ and $\bar{y}_{qk}$ are given as

$$\begin{aligned} \bar{u}_{qk} &= [\, u_{q(k-1)} \quad u_{q(k-2)} \quad \dots \quad u_{q(k-n)}]^T \\ \bar{y}_{qk} &= [\, y_{k-1}^q \quad y_{k-2}^q \quad \dots \quad y_{k-n}^q]^T \end{aligned}. \quad (25)$$

## 4. REDUCING THE QUANTIZATION ERROR

According to Xu et al. (2015) and Zhao et al. (2015), the quantization in (24) causes error in the Bellman equation in (12). A dither signal could be added between the system and the analog-to-digital converter to reduce the quantization error (Schuchman, 1964; Widrow and Kollar 2008). Here, the exploration noise is a dither signal, but it is only possible to insert it into the control input. A method to reduce the quantization error with the exploration noise is developed.

### *4.1 Quantization, sample time and exploration noise*

The sample time, exploration noise and their connection to the quantization noise is studied. Each test starts from the same initial position and the data is collected for 20 time steps. The control gain is chosen as $K = -[0.5 \quad 0.5 \quad 0.5 \quad 0.5]$ and the input saturation is set as $[-7\ V, 7V]$. The exploration noise is added to the control input and it is an uniform random noise between $[-A_\epsilon\ V, A_\epsilon\ V]$, where $A_\epsilon$ is the exploration noise amplitude. Fig. 2. shows 3 tests, where $A_\epsilon = 1$ and the sample time $dt$ is $0.002\ s, 0.01\ s$ or $0.1\ s$.

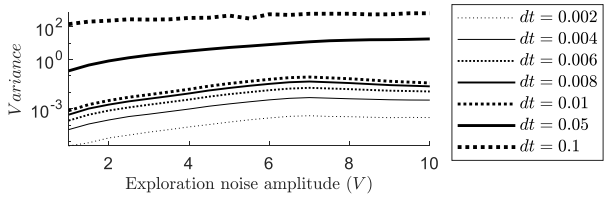


Fig. 3. Quantized output variance as a function of the exploration noise amplitude with different sample times

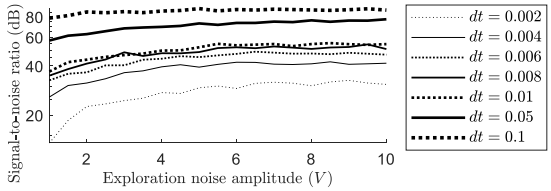


Fig. 4. SNR with different sample times and noise amplitudes

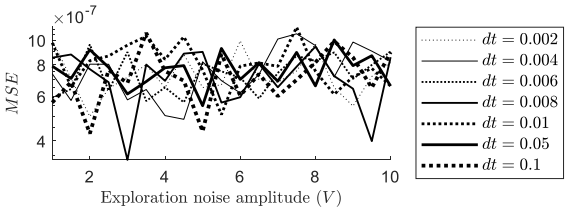


Fig. 5. MSE with different sample times and noise amplitudes

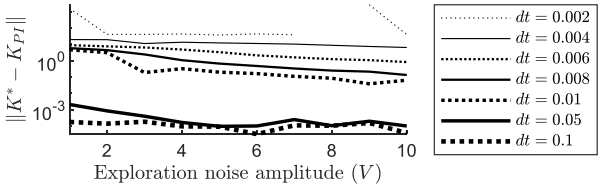


Fig. 6. Norm of the error between the optimal and the learned gain $K^*$ and $K_{VI}$ with different exploration noise amplitudes and sample times

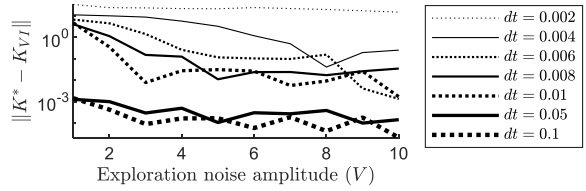


Fig. 7. Norm of the error between the optimal and the learned gain $K^*$ and $K_{PI}$ (NaN values with $\mathrm{dt} = 0.002$ s)

According to Gray and Neuhoff (1998), the quantization interval $\Delta$ is ordinarily small when the number of quantization levels is large. Quantization levels are the values the quantized output can have and here the number of quantization levels is infinite. However, the number of quantization levels in Fig. 2 are 4, 44 and 7910 from top to bottom respectively. It is seen (see Fig. 2 top) that the value of the quantized output might not change between samples if the amount of quantization levels is small, or, put differently, the quantization interval $\Delta_\theta$ is large compared to values the actual output can have. Widrow and Kollar (2008) mention that the quantization noise can be assumed white, but small sample times demand smaller quantization intervals for this assumption to be valid.

The variance of the quantized output is computed using different sample times and exploration noise amplitudes and the results are shown in Fig. 3. The figure shows that the variance is increased when the exploration noise amplitude and the sample time increase. The number of quantization levels increases when the variance increases, since the output can obtain values from a larger range. This is also seen in Fig. 2.

Gray and Nehoff (1998), and Widrow and Kollar (2008) give the signal-to-quantization-noise ratio (SNR) as

$$SNR = 10\log_{10}(\mathrm{var}(Y)/E[(q(Y) - Y)^2]) \qquad (26)$$

with Y as the output signal and $q(Y)$ the quantized signal. The signal-to-quantization-noise ratio is computed for the different datasets (Fig. 4). The ratio increases when the noise amplitude and sample time increase. This means that the quantization noise can be reduced in comparison to the actual signal when the sample time and exploration noise amplitude are increased.

Gray and Neuhoff (1998) define the mean-squared error (MSE) for a small quantization interval $\Delta$ approximately as $\Delta^2/12$. For the given system output it should be approximately $(0.0031\,rad)^2/12 = 8.0083 \cdot 10^{-7}\,rad$. Fig 4 shows the MSEs between the quantized and actual outputs of the different datasets. In fact, all of the MSEs are close to the computed value. Even though changing the sample time and the exploration noise reduces the quantization noise in comparison to the actual signal (as was seen in Fig. 3 and Fig. 4), the absolute size of the quantization interval $\Delta_\theta$ and the absolute size of the quantization error is not reduced.

### *4.2 Choosing the sample time and exploration noise*

The initial gain and the kernel matrix are chosen stabilizing for both PI and VI. To assure the stability, they are not selected randomly here, but instead a small gain is computed using (2), (7) and (8) with the discretized model (23) and $R = 1$ and $Q = \begin{bmatrix} 5 & 0 \\ 0 & 0 \end{bmatrix}$, if $dt \geq 0.01$ or $Q = \begin{bmatrix} 0.1 & 0 \\ 0 & 0 \end{bmatrix}$, if $dt < 0.01$. PI and VI learning algorithms are initialized with $Q_y = 1$ and $R = 1$.

A simulator with input and output quantization and input saturation is used with different exploration noise amplitudes and sample times. The norm of the error between the final learned gain $K_{PI}$ or $K_{VI}$ and the optimal gain $K^*$ is computed for each sample time and noise amplitude (Fig. 6 and Fig. 7). The error becomes smaller with larger noise amplitudes and sample times in this application. This result is expected due to results in Fig. 3 and Fig. 4 as the quantization error becomes proportionally smaller compared to the signal.

While in the simulated environment the performance can be improved by increasing the sample time and noise amplitude, in the real-time system control should be within $\pm$ 10 $V$. The time constant $\tau$ of the system is approximately $\tau = 0.13\ s$ and the system developers use a sample time $dt = 0.002\ s$ (Apkarian et al., 2016). Applying larger noise amplitudes long term could also harm the system. The exploration noise is chosen as an uniform random noise $\epsilon_k \in [-5\ V, 5V]$ and the sample time as $dt = 0.01\ s$. This choice is supported by the earlier results, but it is also chosen so that it is physically possible to use it in the real-time system.

## 5. CONTROL RESULTS WITH THE CHOSEN EXPLORATION NOISE AND SAMPLE TIME

PI and VI are used in a simulated model without disturbances (original simulator) and in a simulated model with added input and output quantization and other disturbances (modified simulator). The exploration noise, the sample time and the initial stabilizing gain $\hat{K}_0$ were chosen in Section 4.2. Convergence limits $\varepsilon_i$ and $\varepsilon_j$ were chosen as $\varepsilon_i = \varepsilon_j = 10^{-7}$ for the original simulator and $\varepsilon_i = \varepsilon_j = 10^{-5}$ for the modified simulator. PI and VI are initialized with $Q_y = 1$ and $R = 1$.

The results with the original and modified simulator are shown in Fig.8. It shows the control gain $\hat{K}_j$ at each time $k$. The gain $\hat{K}_j$ is the gain at iteration $j$ defined as $\hat{K}_j = -(T_{uu,j})^{-1}[T_{u\bar{u},j} \quad T_{u\bar{y},j}]$. Table 1 lists the converged gains $\hat{K}_\infty$. The first row of the table, the reference gain, is computed using (2) and (8) in (7) with $Q = C^T Q_y C$. Near optimal control is learned during every run, so the success rate for both algorithms in a simulated environment is 100 %.

Then, Gaussian random noise was added before the quantization in the modified simulator to model the measurement noise and other disturbances. Three different variances were tested (Fig. 9) and only the largest one lead to unstable learning. Value iteration tolerates the noise more than policy iteration in this application.

Results on how the real-time system worked with the chosen exploration noise and sample time during 200 s run are given in Fig. 10 and Fig. 11. It is seen, that while the learning algorithms were stable in the simulated environment each run, in the real-time environment they can become unstable. Learning in the real-time environment was deemed successful if it remained stable within 200 s, while it might have not reached the optimal value. The real-time learning was run 30 times for both policy and value iteration. Success rate within these runs for policy iteration was $5/30 \approx 17$ % and for value

**Table 1. Learned gains when simulator is used**

| Algorithm | Output feedback gain $\hat{K}_\infty$ | | | |
|---|---|---|---|---|
| LQR ref. | $[-0.1395$ | $-0.0706$ | $-6.6362$ | $5.7039]$ |
| PI RLS orig. | $[-0.1395$ | $-0.0706$ | $-6.6377$ | $5.7054]$ |
| PI RLS mod. | $[-0.1358$ | $-0.0683$ | $-6.4770$ | $5.5655]$ |
| VI RLS orig. | $[-0.1395$ | $-0.0706$ | $-6.6382$ | $5.7058]$ |
| VI RLS mod | $[-0.1387$ | $-0.0708$ | $-6.6064$ | $5.6820]$ |

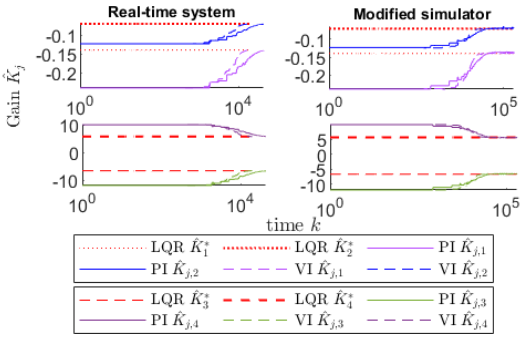


Fig. 8. Learned gain when no disturbances (left) and when quantization and saturation are present (right)

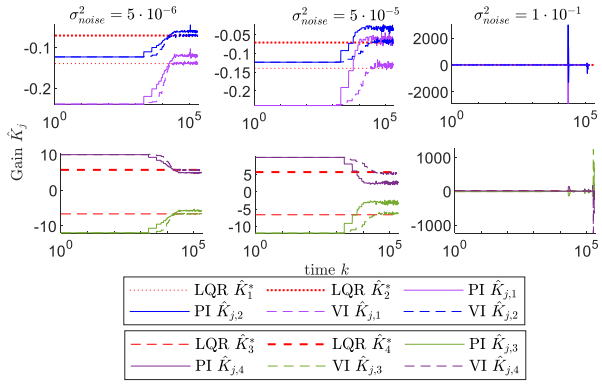


Fig. 9. Disturbances added to the system.

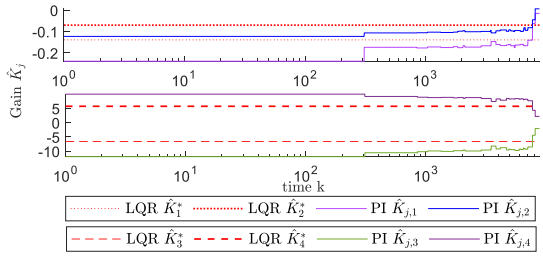


Fig. 10. Unsuccessful real-time learning

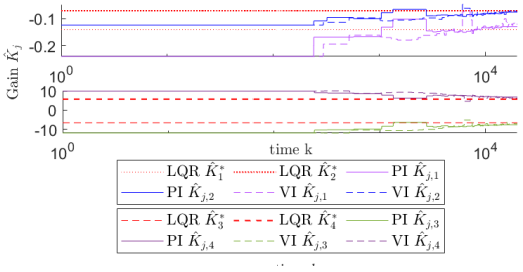


Fig. 11. Successful real-time learning

iteration $10/30 \approx 33$ %. One explanation for the difference between these percentages is that PI needs a stabilizing policy, but the disturbances in the real-time environment cause error in the policies, making the system marginally stable or unstable.

In this case, the frequency of the exploration noise could be designed to be in the frequency range of the system. It would increase the signal-to-quantization-noise ratio as shown in some identification studies earlier (e.g. Roinila et al, 2010). Besides this, it is important to develop algorithms that are adaptive and reactive so that changes in the environment are also considered during control. Vincent and Sun (2012) define a reactive system as a system that can sense the environment and react to the changes by an adaptive control algorithm.

## 6. CONCLUSIONS

Larger exploration noise amplitudes and sample times lead to larger variance in the quantized output and increase the signal-to-quantization-noise ratio and therefore reduce the effects of the quantization noise in the Q-learning algorithms. The new method was proven to work in the simulated environment, but it was not reliable in the real-time environment as it would still lead to instability on some of the test runs.

In small-scale real-time applications the exploration noise amplitude can be increased, if the larger amplitude does not cause danger or damage. However, the control voltage is often constrained and the large variance in the control input can damage the system. Similarly, larger sample time can make the Q-learning algorithm react to changes slower as each control action is implemented after one time step. Therefore, future research must find new ways to reduce the quantization error, e.g. by studying the frequency domain characteristics of the exploration noise.

## ACKNOWLEDGEMENTS

This study was written as a part of MIDAS project at Tampere University. The results are partly based on the results of the M.Sc. thesis of the author Sini Tiistola.

## REFERENCES

Apkarian, J., Lévis, M. and Martin, P. (2016). *Student Workbook: QUBE-Servo 2 Experiment for MATLAB®/ Simulink® Users*. Available from: https://www.quanser.com/courseware-resources/ (Accessed 1 March 2019).

Bellman, R. E. (1957). *Dynamic programming.* Princeton University Press.

Bennett, W. R. (1948). Spectra of quantized signals. *The Bell System Technical Journal,* vol. 27, no. 3, pp. 446-472.

Curry, R. K. (1970). *Estimation and Control with Quantized Measurements*, 116 pp. The MIT Press

Delchamps, D. F. (1990). Stabilizing a linear system with quantized state feedback. *IEEE Transactions on Automatic Control*, vol. 35, no. 8, pp. 916-924.

Franklin, G. F., Powell, J. D. and Workman, M. L. (1998). *Digital Control of Dynamic Systems*, 3rd edition, pp. 1-742. Addison-Wesley.

Gray, R. M. and Neuhoff, D. L. (1998). Quantization, *IEEE Transactions on Information Theory,* vol. 44, no. 6, pp. 2344-2346.

ten Hagen, S. and Kröse, B. (2003). Neural Q-learning. *Neural Computing & Applications,* vol. 12, no. 2, pp. 81-88.

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

Lewis, F. L. and Vamvoudakis, K. G. (2011). Reinforcement Learning for Partially Observable Dynamic Processes: Adaptive Dynamic Programming Using Measured Output Data. *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 41, no. 1, pp. 14-25.

Lewis, F. L., Vrabie, D. and Vamvoudakis, K. G. (2012). Reinforcement learning and Feedback Control. *IEEE Control Systems*, vol. 32, no. 6, pp. 76-105.

Radac, M. and Precup, R. (2018). Data-driven model-free slip control of anti-lock braking systems using reinforcement Q-learning. *Neurocomputing,* vol. 275, pp. 317-329.

Rizvi, S. A. A., and Lin, Z. (2017). Output Feedback Reinforcement Q-Learning Control for the Discrete-Time Linear Quadratic Regulator. *IEEE 56th Annual Conference on Decision and Control*, pp. 1311-1316

Rizvi, S. A. A., and Lin, Z. (2019). An iterative Q-learning scheme for the global stabilization of discrete-time linear systems subject to actuator saturation. *International Journal of Robust and Nonlinear Control*, vol. 29, no. 9, pp. 2660-2672

Roinila, T., Vilkko, M. and Suntio, T. (2010), Frequency-Response Measurement of Switched-Mode Power Supplies in the Presence of Nonlinear Distortions*, IEEE Transactions on Power Electronics*, vol. 25, no. 8, pp. 2179-2187

Schoukens, J., Pintelon, R., van der Ouderaa, E. and Renneboog, J. (1988). Survey of excitation signals for FFT based signal analyzers, *IEEE Transactions on Instrumentation and Measurement*, vol. 37, no. 3, pp. 342-352.

Schuchman, L. (1964). Dither Signals and Their Effect on Quantization Noise. *IEEE Transactions on Communication Technology,* vol. 12, no. 4, pp. 162-165

Sutton, R. S., Barto, A. G. and Bach, F. (2018). *Reinforcement learning: an introduction,* 2nd edition, 526 pp. The MIT Press.

Vincent, I. and Sun Q. (2012) A combined reactive and reinforcement learning controller for an autonomous tracked vehicle. *Robotics and Autonomous Systems*. vol. 60, no. 4, pp. 599–608

Widrow, B. and Kollar, I. (2008). *Quantization noise: roundoff error in digital computation, signal processing, control, and communications.* Cambridge University Press, Cambridge, New York. pp. 560-561

Wang, L. Y., Yin, G. G., Zhang, J. and Zhao, Y. (2010). *System Identification with Quantized Observations,* Birkhäuser.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*, Ph.D. dissertation, University of Cambridge.

Xu, H., Zhao, Q. and Jagannathan, S. (2015). Finite-Horizon Near-Optimal Output Feedback Neural Network Control of Quantized Nonlinear Discrete-Time Systems With Input Constraint. *IEEE Transactions on Neural Networks and Learning Systems,* vol. 26, no. 8, pp. 1776-1788

Zhao, Q., Xu, H., and Jagannathan, S. (2015). Optimal control of uncertain quantized linear discrete-time systems. *International Journal of Adaptive Control and Signal Processing*, vol. 29, no. 3, pp. 325-345.