Continuous State Space Q-Learning for control of Nonlinear Systems

ten Hagen, S.H.G.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 3

LQR using Q-Learning

3.1 Introduction

In this chapter we will present a theoretic framework that enables us to analyze the use of RL in problem domains with continuous state and action spaces. The first theoretical analysis and proof of convergence of RL applied to such problems can be found in [82]. It shows the convergence to the correct value function for a Linear Quadratic Regularization task, where the weights of a carefully chosen function approximator were adjusted to minimize the temporal difference error. Based on the same idea the convergence was proven for other RL approaches, including Q-learning [44].

In [16][19][20][17], a policy iteration based Q-learning approach was introduced to solve a LQR task. This was based on a Recursive Least Squares (RLS) estimation of a quadratic Q-function. These approaches do not use the model of the linear system and can be applied if the system is unknown. If data is generated with sufficient exploration, the convergence to the optimal linear feedback can be proven.

Both [17] and [44] indicate that the practical applicability of the results are limited by the absence of noise in the analysis. A scalar example in [17] shows that the noise introduces a bias in the estimation, making that proofs of convergence no longer hold. We are interested in the use of RL on real practical problem domains with continuous state and action spaces. This means that we have to include the noise in our analysis.

We are also interested in how well the RL approach performs compared to alternative solution methods. We can solve the LQR task with an unknown system using an indirect approach, where data is used to estimate the parameters of the system. Then these estimated parameters are used to compute the optimal feedback. Because we want to compare the results it is important to replace the RLS by a batch linear least squares estimation. This has the advantage that no initial parameters have to be specified and so the resulting solution only depends on the data and the solution method. The result is that both solution methods are off-line optimization methods, because first all data is generated and then the new feedback is computed.

According to the convergence proofs, sufficient exploration is required to find the op-
optimal solution. This means that random actions have to be applied to the system. In a practical control task this is not desirable, so we need an indication of the minimal amount of exploration that is sufficient. This means that our analysis has to show how the performance of the two solution methods depend on the amount of exploration used to generate the data.

In the next section we will specify the Linear Quadratic Regularization task where the linear system is assumed to be unknown. We then present the two solution methods and give an overview on how to compare the performance of these two methods. Section 3.3 will focus on the influence of the exploration on the comparison. We will show that the noise determines the amount of exploration required for a guaranteed convergence. Also we will show that this amount of exploration differs for the two solution methods. The experimental confirmation of the results will be given in section 3.4, followed by the discussion and conclusion in section 3.5 and 3.6.

3.2 LQR with an Unknown System

In this section we will describe the LQR task and show how to obtain the optimal feedback when everything is known. We will then present a direct and an indirect solution method for the situation where the parameters of the system are unknown. Also we will define a performance measure and give an overview of the comparison of the two solution methods.

3.2.1 Linear Quadratic Regulation

In the Linear Quadratic Regulation (LQR) framework, the system is linear and the direct cost is quadratic. Let a linear time invariant discrete time system be given by:

\[ x_{k+1} = Ax_k + Bu_k + v_k, \]

(3.1)

where \( x_k \in \mathbb{R}^{n_x} \) the state, \( u_k \in \mathbb{R}^{n_u} \) the control action and \( v_k \in \mathbb{R}^{n_x} \) the system noise at time step \( k \). All elements of system noise \( \mathbf{v} \) are assumed to be \( N(0, \sigma_v^2) \) distributed and white. Matrix \( A \in \mathbb{R}^{n_x \times n_x} \) and \( B \in \mathbb{R}^{n_x \times n_u} \) are the parameters of the system.

The direct cost \( r \) is a quadratic function of the state and the control action at time \( k \):

\[ r_k = x_k^T S x_k + u_k^T R u_k, \]

(3.2)

where \( S \in \mathbb{R}^{n_x \times n_x} \) and \( R \in \mathbb{R}^{n_u \times n_u} \) are the design choices. The objective is to find the mapping from state to control action \( (\mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}) \) that minimizes the total costs \( J \), which is given by:

\[ J = \sum_{k=0}^{\infty} r_k. \]

(3.3)

The value of \( J \) is finite if (3.2) approaches zero fast enough. This is the case when (3.1) is controlled using (1.3) and the closed loop is stable. The total costs \( J \) becomes infinite if the closed loop is unstable. It is possible to include a discount factor \( \gamma < 1 \) in (3.3),
but then the total costs can be finite for an unstable system. We will use (3.3) without a
discount factor (or \( \gamma = 1 \)), so that a finite \( J \) always implies that the system is stable.

The optimal control action \( u^* \) is a linear function of the state:

\[
    u^*_k = L^* x_k \quad \text{with} \quad L^* = -(B^T K^* B + R)^{-1} B^T K^* A
\]

(3.4)

where \( K^* \in \mathbb{R}^{n_x \times n_x} \) is the unique symmetric positive definite solution to the Discrete
Algebraic Riccati Equation (DARE):

\[
    K^* = A^T (K^* - K^* B (B^T K^* B + R)^{-1} B^T K^*) A + S.
\]

(3.5)

This solution exists if: \( (A, B) \) is controllable, \( (A, S^{\frac{1}{2}}) \) is observable, \( S \geq 0 \) (positive semi-
definite) and \( R > 0 \) (positive definite) [11]. Only with perfect knowledge about \( A, B, S \) and \( R \) can equation (3.5) be solved. This restricts the practical applicability of LQR
because in practice perfect knowledge about \( A, B \) is not available.

### 3.2.2 System Identification

In indirect adaptive control the parameters of the system have to be estimated, and we will refer to this as the System Identification (SI) approach.\(^1\) This is our first method to solve
the LQR problem with unknown \( A \) and \( B \). The estimations are based on measurements
generated by controlling the system using:

\[
    u_k = L x_k + e_k
\]

(3.6)

where \( L \) is the existing feedback\(^2\) and \( e_k \in \mathbb{R}^{n_u} \) represents the excitation (or exploration)
noise. The main difference between \( e \) and \( v \) is that \( v \) is an unknown property of the system,
while \( e \) is a random process that is added on purpose to the control action. All elements
of \( e \) are chosen to be \( \mathcal{N}(0, \sigma^2_e) \) distributed and white. Although the value of \( e \) is always
known, the two methods presented in this chapter will not use this knowledge.

Controlling the system for \( N \) time steps results in a set \( \{x_k\}_{k=0}^N \) with:

\[
    x_k = D_0 + \sum_{i=0}^{k} D^{k-i-1} (B e_i + v_i)
\]

(3.7)

where \( D = A + B L \) represents the closed loop. The sets \( \{x_k\}_{k=0}^N \) and \( \{u_k\}_{k=0}^N \) (computed
with (3.6)) form a data set that depends on the parameters of the system, the feedback, the
initial state and both noise sequences. To estimate the parameters of the system, rewrite
(3.1) to:

\[
    x_{k+1}^T = \begin{bmatrix} x_k^T & u_k^T \end{bmatrix} \begin{bmatrix} A^T & B^T \end{bmatrix} + v_k
\]

(3.8)

\(^1\)Actually closed-loop identification or identification for control would be more appropriate, but system
identification stresses the main difference with the reinforcement learning method more.

\(^2\)Note that this controller already has the form of (3.4), only the value of \( L \) is not optimal.
So for the total data set
\[
Y_{SI} = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix} = \begin{bmatrix} x_0^T \\ \vdots \\ x_{N-1}^T \end{bmatrix} \begin{bmatrix} A^T \\ B^T \end{bmatrix} + \begin{bmatrix} v_0^T \\ \vdots \\ v_{N-1}^T \end{bmatrix} = X_{SI} \theta_{SI} + V_{SI}
\]  
(3.9)

should hold. Since \( V_{SI} \) is not known, only a least squares estimate of \( \theta_{SI} \) can be given:
\[
\hat{\theta}_{SI} = (X_{SI}^T X_{SI})^{-1} X_{SI}^T Y_{SI}.
\]  
(3.10)

The estimated parameters of the system, \( \hat{A} \) and \( \hat{B} \), can be derived from \( \hat{\theta}_{SI} \). When \( \hat{A} \) and \( \hat{B} \) are used, the solution of (3.5) will be \( \hat{K} \). Then a feedback \( \hat{L}_{SI} \) can be computed using (3.4). This feedback \( \hat{L}_{SI} \) is the resulting approximation of \( L^* \) by the SI approach.

### 3.2.3 The Q-function

Reinforcement learning is our second method for solving the LQR task with unknown \( A \) and \( B \). This means that the costs in (3.2) will be regarded as the reinforcements. As explained in chapter 2, the main idea behind RL is to approximate the future costs and find a feedback that minimizes these costs. In LQR the solution of the DARE (3.5) can be used to express the future costs as a function of the state when the optimal feedback is used [11]:
\[
V^*(x) = \sum_{i=k}^{\infty} r_i = x_k^T K^* x_k,
\]  
(3.11)

with \( V^* : \mathbb{R}^{n_x} \rightarrow \mathbb{R} \). The feedback that minimizes (3.11) is given by (3.4), which requires knowledge about \( A \) and \( B \). So, it is not very useful to estimate the parameters \( K^* \) of (3.11). Q-Learning is more appropriate, since it does not require knowledge about the system to obtain the feedback.

In Q-Learning the feedback is derived from the Q-function, which represents the future costs as a function of the state and action. So \( Q : \mathbb{R}^{n_x \times n_u} \rightarrow \mathbb{R} \) is the function to approximate based on the measurements. If we know what function we have to approximate, then we only have to estimate the parameters. According to (2.14)
\[
V^*(x) = \min_u Q^*(x, u) = Q^*(x, u^*)
\]  
(3.12)

should hold. It can be shown [19][44] that \( Q^*(x_k, u^*_k) \) is given by:
\[
Q^*(x_k, u^*_k) = \sum_{i=k}^{\infty} r_i = [x_k^T \quad u_k^T] \begin{bmatrix} S + A^T K^* A & A^T K^* B \\ B^T K^* A & R + B^T K^* B \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}
\]  
(3.13)
\[
= [x_k^T \quad u_k^T] \begin{bmatrix} H_{xx}^* & H_{xu}^* \\ H_{ux}^* & H_{uu}^* \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} = \phi_k^T H^* \phi_k^*.
\]  
(3.14)

\footnote{In optimization tasks other than LQR, the computation of the future costs may be intractable so that an approximation of the future costs may be useful.}
The vector \( \phi_k^T = [x_k^T \quad u_k^T] \) is the concatenation of the state and optimal control action and the matrix \( H^* \) contains the parameters of the optimal Q-function. This shows that the optimal Q-function for the LQR task is a quadratic function of the state and action.

The Q-function in (3.14) can be used to compute the optimal control action without the use of the system model. According to (2.15),

\[
    u_k^* = \arg\min_u Q^*(x_k, u)
\]

(3.15)

should be computed for all states to get the optimal feedback function. The Q-function in (3.14) is a quadratic function and \( H^* \) is a symmetric positive definite matrix. So this function can easily be minimized by setting the derivative to the control action to zero:

\[
    \nabla_{u_k} Q^*(x_k, u_k^*) = 2H_{ux}^* x_k + 2H_{uu}^* u_k^* = 0,
\]

resulting in:

\[
    u_k^* = -(H_{uu}^*)^{-1} H_{ux}^* x_k = L^* x_k \quad \text{with} \quad L^* = -(H_{uu}^*)^{-1} H_{ux}^*.
\]

(3.16)

With the \( H_{ux}^* = B^T K^* A \) and \( H_{uu}^* = R + B^T K^* B \) in (3.14), this result is identical to (3.4).

It is not the optimal Q-function that is being approximated, but the function representing the future costs. This is the function \( Q^L(x, u) = \phi^T H^L \phi \) (with \( \phi^T = [x \quad u]^T \))\(^4\), because all measurements are generated using some feedback \( L \). The \( H^L \) is symmetric and positive definite so that \( L' = -(H_{uu}^L)^{-1} H_{ux}^L \)\(^5\) is the feedback that minimizes \( Q^L \). The \( L' \) does not have to be the optimal feedback but it will have lower future costs than \( L \). Estimating the parameters of \( Q^L \) forms a policy evaluation step and computing \( L' \) forms a policy improvement step. This means that this Q-Learning approach is based on policy iteration. If \( L' \) is not good enough, the whole procedure can be repeated by generating measurements using \( L' \). If the parameters of \( Q^L \) are always estimated correctly, the sequence of new values of \( L' \) forms a contraction towards the optimal solution \( L^* \) [44]. This means we only have to verify the correctness of one policy improvement step. Then the convergence to the optimal solution follows from induction.

### 3.2.4 Q-Learning

In (2.16) the update rule for Q-Learning is given. It is based on repeatedly restarting the system and generating new data in each run. The update also has a learning rate that has to decrease according to (2.19) and (2.20). This makes it impossible to compare the result with that of the SI approach. We therefore have to change the Q-learning algorithm such that it uses one single data set and does not use a learning rate.

The parameters \( H^L \) of the Q-function should be estimated in the same way as the parameters of the system in paragraph 3.2.2. So the same data set with \( \{x_k\}_k^N \) and \( \{u_k\}_k^N \) is used. Q-Learning also uses scalar reinforcements. These are the direct costs

\[^4\text{If there is also noise: } Q^L(x, u) = \phi^T H^L \phi + v^T K^L v\]

\[^5\text{The } ' \text{ indicates the feedback that is optimal according to the Q-function, so } L' \text{ optimizes } Q^L. \text{ This is}\
\text{equivalent to the greedy policy described in chapter 2.}\]
computed with (3.2). The parameters of $Q^L$ are estimated based on the data set, generated using feedback $L$. The function $Q^L$ can be estimated by writing its definition recursively:

$$Q^L(x_k, u_k) = \sum_{i=k}^{\infty} r_i = r_k + \sum_{i=k+1}^{\infty} r_i = r_k + Q^L(x_{k+1}, Lx_{k+1}). \quad (3.17)$$

Note that this definition implies that the data is generated using a stabilizing feedback. In case the feedback $L$ is not stable, the function $Q^L(x_k, u_k)$ is not defined because the sum of future reinforcement is not bounded. Therefore the correct values of $H^L$ do not exist and cannot be estimated.

From (3.17) it follows that:

$$r_k + Q^L(x_{k+1}, Lx_{k+1}) - Q^L(x_k, u_k) = 0. \quad (3.18)$$

If in this equation $Q^L$ is replaced by its approximation $\hat{Q}^L$ the left hand side is the Temporal Difference (TD). Because both functions are quadratic, the right hand side of (3.18) is only zero if $\hat{Q}^L$ has the same parameters as $Q^L$. The parameters of $\hat{Q}^L$ can be estimated by reducing the distance between the TD and zero. This can be formulated as a least squares estimation as in (3.10). We define:

$$\phi_k^T = [x_k^T \quad u_k^T], \quad \phi_{k+1}^T = [x_{k+1}^T \quad L^T x_k^T] \quad \text{and} \quad w_k^T = v_k^T K^L u_k - v_{k+1}^T K^L v_{k+1}. \quad (3.19)$$

Note that the definition of $\phi_{k+1}^T$ is slightly different from $\phi_k^T$. It is possible to write (3.18) as:

$$r_k = Q^L(x_k, u_k) - Q^L(x_{k+1}, Lx_{k+1})$$

$$= \phi_k^T H^L \phi_k - \phi_{k+1}^T H^L \phi_{k+1} + w_k$$

$$= \text{vec}^q(\phi_k \phi_k^T)^T \text{vec}^q(H^L) - \text{vec}^q(\phi_{k+1} \phi_{k+1}^T)^T \text{vec}^q(H^L) + w_k$$

$$= \text{vec}^q(\phi_k \phi_k^T - \phi_{k+1} \phi_{k+1}^T)^T \text{vec}^q(H^L) + w_k = \text{vec}^q(\Phi_k) \text{vec}^q(H^L) + w_k. \quad (3.20)$$

Note that the matrix $\Phi_k$ also depends on $L$. For all time steps the following holds:

$$Y_{QL} = \begin{bmatrix} r_0 \\ \vdots \\ r_{N-1} \end{bmatrix} = \begin{bmatrix} \text{vec}^q(\Phi_0)^T \\ \vdots \\ \text{vec}^q(\Phi_{N-1})^T \end{bmatrix} \text{vec}^q(H^L) + \begin{bmatrix} w_0 \\ \vdots \\ w_{N-1} \end{bmatrix} = X_{QL} \theta_{QL} + V_{QL}. \quad (3.21)$$

so that

$$\hat{\theta}_{QL} = (X_{QL}^T X_{QL})^{-1} X_{QL}^T Y_{QL} \quad (3.22)$$

Note that for the SI approach the parameters of the system are unknown, but still the weighting of the design matrices can be made. Conceptually this does not make any sense. In a practical situation it is more likely that some scalar indication of performance is available, like for instance the energy consumption of the system. We compute the direct cost using (3.2) for fair comparison.

Define $\text{vec}^q(A)$ as the function that stacks the upper triangle elements of matrix $A$ into a vector.
3.2 LQR WITH AN UNKNOWN SYSTEM

gives an estimation of $\text{vec}(H^L)$. Since $\hat{H}$ should be symmetrical it can be derived from $\text{vec}(H^L)$. By applying (3.16) to matrix $\hat{H}$ the resulting feedback $\hat{L}_{\text{QL}}$ for the Q-Learning approach is computed. This should be an approximation of $L'$.

This variant of Q-learning only applies to the LQR framework, therefore we will refer to it as the Linear Quadratic Regulation Q-Learning (LQRQL) approach. The main difference is that it does not require restarting the system several times to generate sufficient data for the correct estimation of all Q-values. Instead it uses prior knowledge about the function class of the Q-function that is chosen to fit the LQR task. This can be seen as generalization over the state-action space, which for the LQR task is globally correct. This also holds for the approaches in [17][44]. The only difference between these approaches and our approach using (3.25), is that we choose to learn in one step using the entire data set. The main reason for doing this is to make it possible to compare the result with the SI approach. Also the analysis of the outcome is easier when the estimation is not performed recursively.

3.2.5 The Performance Measure

We have described two different methods that use measurements to optimize a feedback resulting in $\hat{L}_{\text{SI}}$ and $\hat{L}_{\text{QL}}$. For the comparison a scalar performance measure is required to indicate which of these feedbacks performs best. There are three ways to measure the performance:

- **Experimental**: Run the system with the resulting feedbacks and compute the total costs. The performances of both approaches can only be compared for one specific setup and it does not indicate how "optimal" the result is. For a real system, this is the only possible way to compare the results.

- **Optimal Feedback**: In a simulation there is knowledge of $A$ and $B$, so the optimal feedback $L^*$ can be computed using (3.4). A norm\(^9\) $\|L^* - \hat{L}\|$ will not be a good performance measure because feedbacks with similar $\|L^* - \hat{L}\|$ can have different future costs. It is even possible that if $\|L^* - \hat{L}_1\| < \|L^* - \hat{L}_2\|$, $\hat{L}_1$ results in an unstable closed loop while $\hat{L}_2$ results in a stable closed loop. This means that this measure can be used to show that the resulting feedback approaches the optimal feedback, but this does not show that this will result in lower total costs.

- **DARE Solution**: Knowledge about $A$ and $B$ can be used in (3.5) to compute the solution of the DARE $K^*$. This gives the future costs (3.11) when starting in $x_0$ and using $L^*$. The costs when using an approximated feedback $\hat{L}$ can be expressed also like (3.11), but then with matrix $K^L$. Comparing the matrices $K^*$ and $K^L$ results in a performance indication that only depends on the initial state.

---

\(^8\)This should not be confused with the Least Square TD approach [18][13], that applies to MDPs.

\(^9\)Here $\hat{L}$ indicates the resulting feedback of both approaches.
We will define a performance measure based on the DARE solution, because it is the least sensitive to the settings of the experiment.

When using $\hat{L}$, the value function $V^\hat{L}(x_0)$ gives the total costs $J^\hat{L}$ when starting in state $x_0$. It is given by:

$$V^\hat{L}(x_0) = \sum_{k=0}^{\infty} r_k = x_0^T \sum_{k=0}^{\infty} (A^T + \hat{L}^T B)^k (S + \hat{L}^T R \hat{L})(A + B \hat{L})^k x_0 = x_0^T K^L x_0 \tag{3.26}$$

where $K^L$ is again a symmetric matrix. (It is clear that this matrix only exists when the closed loop $A + B \hat{L}$ has all its eigenvalues in the unit disc). When $L^*$ is used the total costs $V^*(x_0)$ can be computed using (3.11). Let the relative performance (RP) $\rho(x_0)$ be the quotient between $V^\hat{L}(x_0)$ and $V^*(x_0)$, so:

$$\rho^\hat{L}(x_0) = \frac{V^\hat{L}(x_0)}{V^*(x_0)} = \frac{x_0^T K^\hat{L} x_0}{x_0^T K^* x_0} = \frac{x_0^T (K^*)^{-1} K^\hat{L} x_0}{x_0^T x_0} = x_0^T \Gamma^L x_0 \tag{3.27}$$

where $\Gamma^L = (K^*)^{-1} K^L$. Only when $\hat{L} = L^*$, $\Gamma^L$ is the unit matrix. The RP $\rho^\hat{L}(x_0)$ is bounded below and above by the minimal and maximal eigenvalues of $\Gamma^L$:

$$\rho_{\min}^\hat{L} = \lambda_{\min}(\Gamma^L) \leq \rho^\hat{L}(x_0) \leq \lambda_{\max}(\Gamma^L) = \rho_{\max}^\hat{L} \quad \forall x_0 \neq 0 \tag{3.28}$$

Note that $\rho_{\min}^\hat{L} = \rho_{\max}^\hat{L} = 1$ if and only if $\hat{L} = L^*$ and that $\rho_{\min}^\hat{L} \geq 1 \forall \hat{L}$.

According to (3.28) three possible measures for the RP can be used: $\rho_{\min}^L$, $\rho_{\max}^L$ or $\rho^\hat{L}(x_0)$. It does not matter for the comparison which measure is used, so in general we will use $\rho^\hat{L}$ to indicate one of these measures. Note that $\rho_{\min}^\hat{L}$ and $\rho_{\max}^\hat{L}$ only depend on the feedback $\hat{L}$ and the four matrices $A, B, S$ and $R$ that define the problem. In a practical situation $\rho_{\max}^\hat{L}$ seems the best choice because it represents the worst case RP with respect to $x_0$. In this chapter we will call feedback $L_1$ better than feedback $L_2$ if $\rho^{L_1} < \rho^{L_2}$.

### 3.2.6 Overview

The schematic overview in figure 3.1 summarizes this section. The setup at the left shows the system parameters and noise, but also the feedback $L$ and exploration noise $e$ to generate the measurements indicated with $x, u$ and $r$ (note that $r$ is computed using $S$ and $R$). The SI approach is shown at the top and the Q-Learning approach at the bottom. The computation of the optimal feedback using $A$ and $B$ is shown in the middle.

For LQRQL figure 3.1 shows no explicit optimization, because this is implicitly included in the estimation of the Q-function. This is the difference between the SI and the QL approach: the SI approach is a two step method, where the estimation and the optimization are performed independently. LQRQL is a one step method, where estimation and optimization are performed at once. Figure 3.1 also shows that no additional information is required to derive $\hat{L}_{QL}$ from $H^L$.

The question mark at the very right of figure 3.1 indicates the comparison between $\rho_{\delta SI}$ and $\rho_{\delta QL}$. In the next section we will relate this comparison to the amount of exploration, indicated with the $e$ at the very left of figure 3.1.
3.3 The Influence of Exploration

In this section we will investigate the influence of the exploration on the relative performances and the comparison. We will start in 3.3.1 by reformulating the method of the estimation to make it possible to express the estimation error of the linear least squares estimation. In 3.3.2 and 3.3.3 the influence of the exploration is investigated for both methods. In 3.3.4 the exploration characteristic will be introduced to describe the influence of the exploration on the performance of the resulting feedbacks.

3.3.1 The estimation reformulated

Both approaches described in the previous section, are based on a linear least squares estimation. The equations (3.10) and (3.25) can be written as:

\[ \hat{\theta} = (X^T X)^{-1} X^T Y. \]  

(3.29)

This solution \( \hat{\theta} \) depends only on the matrices \( X \) and \( Y \), so no additional parameters influence the result. For a fair comparison this is an important property. In practice (3.29) is hardly ever used because of its poor numerical performance. Different decomposition methods exist to overcome numerical problems. This is important for the implementation of the simulations, but we will also use it to investigate the influence of the exploration. The matrix inversion in (3.29) is the main problem for our analysis, because this makes it very hard to see how the exploration influences the estimation.
In QR-decomposition\(^{10}\) \(X\) is decomposed into an upper triangular square matrix \(M\) and a unitary matrix \(Z\), so that (3.29) can be written as:

\[
\hat{\theta} = ((ZM)^T ZM)^{-1} (ZM)^T Y = M^{-1} Z^T Y.
\] (3.30)

This is sufficient for an efficient implementation but it still uses a matrix inversion, making it hard to see how this solution depends on the exploration. To see the influence of the exploration, this solution has to be rearranged even more.

The definition of \(Z\) and \(M\) in appendix A.1 makes use of projection matrices \(P\). Let \(P_i\) be the projection matrix corresponding to \(X_{*i}\), which represents the \(i^{th}\) column of \(X\). Then \(P_i\) can be defined recursively according to:

\[
P_i = P_{i-1} - \frac{P_{i-1}X_{*i-1}X_{*i-1}^T P_{i-1}^T}{\|P_{i-1}X_{*i-1}\|^2_2} \quad \text{and} \quad P_1 = I.
\] (3.31)

So \(P_i\) depends on all columns of \(X\) from \(X_{*i}\) to \(X_{*i-1}\). Multiplying these columns with \(P_i\) results in a zero vector, so the part of \(X_{*i}\) that is a linear combination of the columns \(X_{*i}\) to \(X_{*i-1}\) does not contribute to outcome of \(P_i X_{*i}\).

Appendix A.2 shows that matrices \(P\) can be used to solve (3.30) without matrix inversion. Let \(\hat{\theta}_{*i}\) be the last row and \(\hat{\theta}_{i*}\) be the \(i^{th}\) row. Then they are given by:\(^{11}\)

\[
\hat{\theta}_{*i} = \frac{X_{*i}^T P_i^T}{\|P_i X_{*i}\|^2_2} Y
\] (3.32)

\[
\hat{\theta}_{i*} = \frac{X_{*i}^T P_i^T}{\|P_i X_{*i}\|^2_2} (Y - \sum_{j=i+1}^n X_{*j} \hat{\theta}_{*j}) \quad \text{for} \quad i < n.
\] (3.33)

So \(\hat{\theta}\) can be obtained recursively by starting at the last row. If one of the columns of \(X\) is a linear combination of all other columns then \((X^T X)^{-1}\) is singular. In this situation \(P_i X_{*i}\) in (3.33) will become zero, resulting in a singularity as well.\(^{12}\) We will use (3.33) only for the theoretical analysis of the exploration and not for the implementation, because its numerical performance is even worse than (3.29).

### 3.3.2 The System Identification approach

We first rewrite the estimation for the SI approach and show how the resulting feedback depends on the estimation. We then express the estimation error and show how it depends on the exploration. Finally we show the consequences of the estimation error on the resulting feedback and its performance.

\(^{10}\)The name refers to matrices \(Q\) and \(R\), but we will use \(Z\) and \(M\) because we already use the symbols \(Q\) and \(R\).

\(^{11}\)In the rest of the chapter we will ignore the absence of the sum term for the \(n^{th}\) row by defining a dummy \(\hat{\theta}_{n+1*}\) that equals zero.

\(^{12}\)The scalar \(\|P_i X_{*i}\|^2_2\) in (3.33) is squared, so it will go faster to zero than the elements of vector \(X_{*i}^T P_i^T\).
The Estimation

To show the influence of the exploration, matrix $X_{Sl}$ in (3.9) can be split in a part that depends on $x$ and in a part that depends on $u$:

$$X_{Sl} = \begin{bmatrix} X & U \end{bmatrix} = \begin{bmatrix} X & XL^T + \mathcal{E} \end{bmatrix}$$

with $X = \begin{bmatrix} x_0^T \\ \vdots \\ x_N^T \end{bmatrix}$, $U = \begin{bmatrix} u_0^T \\ \vdots \\ u_N^T \end{bmatrix}$, and $\mathcal{E} = \begin{bmatrix} e_0^T \\ \vdots \\ e_N^T \end{bmatrix}$.

(3.34)

Also the control actions are split into a feedback part and an exploration part, but according to (3.9) some exploration is still contained in $X$ and $Y_{Sl}$.

Appendix A.3 shows that the columns of $\tilde{B}$ and $\tilde{A}$ are given by:

$$\tilde{B}_{si} = \frac{\mathcal{E}_{si}^T P_{n_s+i}^T}{\|P_{n_s+i} \mathcal{E}_{si}\|_2^2} (Y_{Sl} - \sum_{j=n_s+i+1}^{n_s+n_u} U_{ji} \tilde{B}_{sj})$$

(3.35)

$$\tilde{A}_{si} = \frac{X_{si}^T P_{n_s+i}^T}{\|P_{n_s+i}X_{si}\|_2^2} (Y_{Sl} - \sum_{j=i+1}^{n_s} X_{sj} \tilde{A}_{sj} - U \tilde{B}^T).$$

(3.36)

Without exploration the value of $\tilde{B}$ becomes infinite, because $\|P_{n_s+i} \mathcal{E}_{si}\|_2^2$ approaches zero faster than $\mathcal{E}_{si}^T P_{n_s+i}^T$. This also makes $\tilde{A}$ infinite. For low exploration the term $U \tilde{B}^T$ dominates the outcome of (3.36). Then $\tilde{A}$ becomes more linear dependent on $\tilde{B}$. So for low exploration ($\tilde{A}, \tilde{B}$) are more likely to be uncontrollable.

Appendix A.3 also shows that the columns of $\dot{D} = \dot{A} + L \tilde{B}$ are given by:

$$\dot{D}_{si} = \frac{X_{si}^T P_{i}^T}{\|P_{i} X_{si}\|_2^2} (Y_{Sl} - \mathcal{E} \tilde{B}^T - \sum_{j=i+1}^{n_s} X_{sj} \dot{D}_{sj}).$$

(3.37)

Because $\tilde{B}$ is multiplied with $\mathcal{E}$, $\dot{D}$ does not become very large for low amounts of exploration. Therefore we will use $\tilde{B}$ and $\dot{D}$ to obtain the resulting feedback $\dot{L}_{Sl}$.

The Feedback

To compute the feedback (3.5) should be solved using $\tilde{A}$ and $\tilde{B}$, so:

$$\dot{K} = \tilde{A}^T (\dot{K} - \dot{K} \tilde{B} (\tilde{B}^T \dot{K} \tilde{B} + R)^{-1} \tilde{B}^T \dot{K}) \tilde{A} + S.$$  

(3.38)

A unique solution $\dot{K}$ to (3.38) does not have to exist, especially when $\tilde{A}$ and $\tilde{B}$ are too large due to insufficient exploration. In this case the right hand side of (3.38) will become very small making $\dot{K} \approx S$. So we will assume that $K^* - \dot{K}$ is not too large. The feedback is computed according to (3.4) using the estimated matrices:

$$\dot{L}_{Sl} = -(R + \tilde{B}^T \dot{K} \tilde{B})^{-1} \tilde{B}^T \dot{K} \tilde{A} = (R + \tilde{B}^T \dot{K} \tilde{B})^{-1} \tilde{B}^T \dot{K} (\tilde{B}L - \dot{D}).$$

(3.39)

By replacing $\tilde{A}$ by $\tilde{B}L - \dot{D}$, two possible outcomes can already be given:
• **Too low exploration:** \( \hat{L}_{SI} \approx L \)

If the amount of exploration is much too low, \( \hat{B} \) in (3.39) becomes much too large because of the low value of \( \mathcal{E} \) in (3.35). So \( \hat{D} \) and \( \hat{R} \) can be neglected, resulting in \( \hat{L}_{SI} \approx (\hat{B}^T K \hat{B})^{-1} \hat{B}^T K B L = L \). This means that the outcome will approximately be the feedback that was used to generate the data!

• **High exploration:** \( \hat{L}_{SI} \approx L^* \)

For very high amounts of exploration the system noise \( V_{SI} \) in (3.9) can be neglected. The least squares estimation will almost be perfect, so solving the DARE and computing feedback will approximately have the optimal feedback \( L^* \) as outcome.

We can conclude that for insufficient exploration the relative performance does not change and for abundant exploration the relative performance approaches one. We will determine the minimal amount of exploration required to obtain the second outcome.

**The Estimation Error**

By defining \( y_k = x_{k+1} \) and using (3.7), it is possible to write:

\[
y_k = D^{k+1} x_0 + \sum_{i=0}^{k} D^{k-i} (Be_i + v_i).
\]

(3.40)

So \( Y_{SI} \) can be written as:

\[
Y_{SI} = XD^T + \mathcal{E}B^T + V_{SI}.
\]

(3.41)

This can be used to get an expression for the error in the estimations of \( B \) and \( D \).

\[
\hat{B}_{si} = \frac{\mathcal{E}_{si}^T P_{n_x+i}^T}{\|P_{n_x+i} + \mathcal{E}_{si}\|^2} (XD^T + \mathcal{E}B^T + V_{SI} - \sum_{j=n_x+i+1}^{n_x+n_u} U_{..j} \hat{B}_{..j}) \quad (3.42)
\]

\[
= \frac{\mathcal{E}_{si}^T P_{n_x+i}^T}{\|P_{n_x+i} + \mathcal{E}_{si}\|^2} (\mathcal{E}B^T + V_{SI} - \sum_{j=n_x+i+1}^{n_x+n_u} \mathcal{E}_{..j} \hat{B}_{..j}) \quad (3.43)
\]

\[
= B_{..i} + \frac{\mathcal{E}_{si}^T P_{n_x+i}^T}{\|P_{n_x+i} + \mathcal{E}_{si}\|^2} (V_{SI} - \sum_{j=n_x+i+1}^{n_x+n_u} \mathcal{E}_{..j} (\hat{B}_{..j} - B_{..j})). \quad (3.44)
\]

So the estimation error \( \hat{B}_{si} = \hat{B}_{si} - B_{si} \) is given by:

\[
\hat{B}_{si} = \frac{\mathcal{E}_{si}^T P_{n_x+i}^T}{\|P_{n_x+i} + \mathcal{E}_{si}\|^2} (V_{SI} - \sum_{j=n_x+i+1}^{n_x+n_u} \mathcal{E}_{..j} \hat{B}_{..j}) \quad (3.45)
\]

In the same way:

\[
\hat{D}_{si} = \frac{\mathcal{X}_{si}^T P_{n_x+i}^T}{\|P_{n_x+i} + \mathcal{X}_{si}\|^2} (XD^T + \mathcal{E}B^T + V_{SI} - \sum_{j=i+1}^{n_x} \mathcal{X}_{..j} \hat{D}_{..j}) \quad (3.46)
\]

\[
D_{si} = \frac{\mathcal{X}_{si}^T P_{n_x+i}^T}{\|P_{n_x+i} + \mathcal{X}_{si}\|^2} (V_{SI} - \mathcal{E}B^T - \sum_{j=i+1}^{n_x} \mathcal{X}_{..j} \hat{D}_{..j}). \quad (3.47)
\]
The estimation errors depend on the exploration noise in $\mathcal{E}$ and the system noise in $V_{SI}$.

The estimations $\hat{B}$ and $\hat{D}$ can be written as the sum of the correct value and the estimation error. So $\hat{B} = B + \hat{B}$ and $\hat{D} = D + \hat{D}$. These expressions can be used in (3.39) to see how the resulting feedback depends on the exploration and system noise, because the correct values $B$ and $D$ do not depend on $\mathcal{E}$ and $V_{SI}$.

### The Minimal Exploration

Expressions (3.45) and (3.47) hold for any $\mathcal{E}$ and $V_{SI}$. To focus on the amounts of exploration and system noise, the estimation errors should be expressed using $\sigma_e$ and $\sigma_v$. These errors depend on the configuration so only an indication of the level of magnitude of these errors can be given.

We have to make the following assumptions:

\[
E\{\xi_i^T \xi_i\} \sim c_1 N \sigma_e^2 \\
E\{\xi_i^T V_{SIi}\} \sim c_2 N \sigma_v \sigma_v \\
E\{|P_i X_i|_2^2\} \sim c_3 + c_4 \sigma_v^2 + c_5 \sigma_v \sigma_e + c_6 \sigma_e^2,
\]

with $c_1 \cdots c_6$ constants depending on $n_x$, $n_u$ and $B$. Note that these assumptions are rude approximations that indicate the expectation value over a time interval of size $N$. Assumption (3.49) indicates that the level of magnitude of the expectation value is proportional to the cross correlation between $\xi_i$ and $V_{SIi}$. Given the fixed time interval, this value may vary a lot depending on the particular noise sequences. This means that the constant $c_2$ depends very much on these noise sequences. The same holds for $c_5$ in (3.49). The constant $c_3$ is included to incorporate the dependency on the initial state $\mathbf{x}_0$.

Now it is possible to give an approximation of the expected estimation errors (3.45) and (3.47). They are proportional to:

\[
E\{\hat{B}\} \sim \frac{c_2 \sigma_v}{c_1 \sigma_e} \\
E\{\hat{D}\} \sim \frac{\sigma_v}{\sqrt{c_3 + c_4 \sigma_v^2 + c_5 \sigma_v \sigma_e + c_6 \sigma_e^2}}.
\]

Note that $E\{\hat{D}\}$ has a maximum for $\sigma_e = -\frac{c_4 \sigma_v}{2 c_6} \sim \sigma_v$, which in general is slightly less than $\sigma_v$.

The errors in (3.52) are zero if $\sigma_v = 0$, so exploration is only required to prevent singularity in the computations of the least squares estimate.

For $\sigma_v \neq 0$ it is possible to neglect $\hat{D}$ and $R$ in (3.39) if $\hat{B}$ makes $\hat{B}$ much too large. The maximum of $E\{\hat{D}\}$ is less than one and $E\{\hat{B}\}$ is also less than one, so for $\sigma_e \approx \sigma_v$ the $\hat{D}$ and $R$ in (3.39) cannot be neglected. This is the minimal amount of exploration that is required, for more exploration the estimations will almost be correct. So as a rule of thumb:

*The amount of exploration should be larger than the amount of system noise!*
3.3.3 The LQRQ approach

Analog to the SI approach, we will determine the dependency of the estimation errors on the exploration and system noise. We also will show the consequences of the estimation error on the resulting feedback and its performance.

The Estimation

To show the influence of the exploration, matrix $X_{QL}$ in (3.24) should be rearranged in such a way that linear dependencies between the columns of $X_{QL}$ can be used. Write $\Phi_k$ as:

$$
\Phi_k = \Phi_k^T - \Phi_{k+1}^T
$$

$$
= \begin{bmatrix}
    x_k & u_k \\
    x_{k+1} & u_{k+1} \\
\end{bmatrix}
- \begin{bmatrix}
    x_{k+1}^T & x_{k+1}^T L^T
\end{bmatrix}
$$

$$
= \begin{bmatrix}
    \Phi_{kx}^T & \Phi_{ku}^T \\
    \Phi_{kx}^{uT} & \Phi_{ku}^{uT}
\end{bmatrix}
$$

(3.53)

with:

$$
\Phi_{kx}^T = x_k x_k^T - x_{k+1} x_{k+1}^T
$$

$$
\Phi_{ku}^T = L x_{k+1}^T + e_k x_k^T
$$

$$
\Phi_{ku}^{uT} = \Phi_{ku}^{uT} L^T + u_k e_k^T.
$$

(3.54)

The rows vec$^c(\Phi_k)$ of $X_{QL}$ do not have the elements arranged according to (3.54), so we redefine $X_{QL}$ as,

$$
X_{QL} = \begin{bmatrix}
    \text{vec}(\Phi_{kx})^T & \text{vec}(\Phi_{ku})^T & \text{vec}(\Phi_{ku}^{uT})^T \\
    \vdots & \vdots & \vdots \\
    \text{vec}(\Phi_{kx})^{N-1} & \text{vec}(\Phi_{ku})^{N-1} & \text{vec}(\Phi_{ku}^{uT})^{N-1}
\end{bmatrix}
$$

(3.55)

The submatrices $\Psi_{xx}$, $\Psi_{ux}$ and $\Psi_{uu}$ correspond to $\hat{H}_{xx}$, $\hat{H}_{ux}$ and $\hat{H}_{uu}$, which are rearrangements of the vectors $\tilde{\theta}_{xx}$, $\tilde{\theta}_{ux}$ and $\tilde{\theta}_{uu}$. Vector $\tilde{\theta}_{xx}$ has $n_{xx} = \frac{1}{2} n_x (n_x + 1)$ elements, vector $\tilde{\theta}_{ux}$ has $n_{ux} = n_x n_u$ elements and vector $\tilde{\theta}_{uu}$ has $n_{uu} = \frac{1}{2} n_u (n_u + 1)$ elements.

Let $L_r$ be the feedback matrix such that vec$(L \Phi_{kx}^T) = L_r$vec$(\Phi_{kx}^T)$ and let $L$ be the feedback such that vec$(\Phi_{ux})^T = L_r$vec$(\Phi_{ux})^T L^T$. Define matrix $\Upsilon$ with rows vec$(e_k x_k^T)$ and matrix $T$ with rows vec$^c(u_k e_k^T)$.

Then

$$
\Psi_{ux} = L_r \Psi_{xx} + \Upsilon \\
\text{and} \\
\Psi_{uu} = \Psi_{ux} L^T + T
$$

(3.56)

can be used to find expressions for $\tilde{\theta}_{xx}$, $\tilde{\theta}_{ux}$ and $\tilde{\theta}_{uu}$ using (3.33).

13For the implementation this is not required, only for the analysis.

14The function vec$(A)$ stacks all columns of $A$ into one vector. Note that here vec$(\Phi_{kx}^{uT})$ is used instead of vec$(\Phi_{kx}^{uT})$. Only the order of elements is different because $\Phi_k$ is symmetric and so vec$(\Phi_{kx}^{uT}) = vec(\Phi_{kx}^{uT})^T$. The reason for doing this is that the calculation of the feedback according to (3.16) makes use of $\hat{H}_{ux}$ and not $\hat{H}_{ux}$. 


The Feedback

To compute the feedback, (3.16) should be solved using \( \hat{H}_{ux} \) and \( \hat{H}_{uu} \), so:

\[
\hat{L}_{QL} = -\hat{H}_{uu}^{-1} \hat{H}_{ux}. \tag{3.57}
\]

\( \hat{H}_{uu} \) and \( \hat{H}_{ux} \) are obtained by rearranging the vectors \( \hat{\theta}_{ux} \) and \( \hat{\theta}_{uu} \). The difference with the SI approach is that the feedback directly follows from the estimations, so it can only be investigated by looking at \( \hat{\theta}_{ux} \) and \( \hat{\theta}_{uu} \). These estimations are according to (3.25) a solution to:

\[
Y_{QL} = \left[ \psi^{ux} \ \psi^{uu} \right] \begin{bmatrix} \theta_{xx} \\ \theta_{ux} \\ \theta_{uu} \end{bmatrix} + V_{QL}. \tag{3.58}
\]

The estimation of \( \hat{\theta}_{uu} \) using (3.33) is given by:

\[
\hat{\theta}_{uu,i} = \frac{\psi_{uu}^T P_{n_{ux}+n_{uu}+i}^T}{\|P_{n_{ux}+n_{uu}+i}^T \psi_{uu}\|^2} (Y_{QL} - \sum_{j=n_{ux}+n_{uu}+i+1}^{n_{ux}+n_{uu}} \psi_{uj}^T \hat{\theta}_{uu,j}) \tag{3.59}
\]

\( T_{ee} \) has vec\(^a\)(\(e_k e_k^T\))^T as rows, because the vec\(^a\)(\(L_x \epsilon_k^T\))^T has no effect on the multiplication with matrices \( P_{n_{ux}+n_{uu}+i} \). Equation (3.59) is similar to (3.35) and without exploration \( T_{ee} \) becomes zero causing a singularity (just like \( E \) in (3.35)). The estimation of \( \hat{\theta}_{ux} \) has a similar form as (3.36):

\[
\hat{\theta}_{ux,i} = \frac{\psi_{ux}^T P_{n_{ux}+i}^T}{\|P_{n_{ux}+i}^T \psi_{ux}\|^2} (Y_{QL} - \sum_{j=n_{ux}+i+1}^{n_{ux}} \psi_{uj}^T \hat{\theta}_{ux,j} - \psi_{uu} \hat{\theta}_{uu}). \tag{3.60}
\]

The linear relation \( \psi_{uu} = \psi_{ux} L^T + T \) resembles \( U = X L^T + E \). So it also possible to define a \( \hat{\theta}_d \), equivalent to the closed loop (3.37) in the SI approach, according to:

\[
\hat{\theta}_{d,i} = \frac{\psi_{ux}^T P_{n_{ux}+i}^T}{\|P_{n_{ux}+i}^T \psi_{ux}\|^2} (Y_{QL} - T_{ee} \hat{\theta}_{uu} - \sum_{j=n_{ux}+i+1}^{n_{ux}} \psi_{uj}^T \hat{\theta}_{d,j}). \tag{3.61}
\]

Since \( \hat{\theta}_d = \hat{\theta}_{ux} + \hat{\theta}_{uu} L^T \) can be rearranged to \( \hat{H}_d = \hat{H}_{ux} + \hat{H}_{uu} L \), (3.57) can be written as:

\[
\hat{L}_{QL} = \hat{H}_{uu}^{-1}(\hat{H}_{uu} L - \hat{H}_d) = L - \hat{H}_{uu}^{-1} \hat{H}_d. \tag{3.62}
\]

With this result two possible outcomes can be obtained:

- **Too low exploration**: \( \hat{L}_{QL} \approx L \)

  If the amount of exploration is much too low \( \hat{H}_{uu} \) is much larger than \( \hat{H}_d \), so the second term in (3.62) can be ignored. The outcome will approximately be the feedback that was used to generate the data!

\(^{15}\)This is because matrix \( \Phi^{uu} \) is symmetric.
• High exploration: \( \hat{L}_{QL} \approx L' \)

For very high exploration, the value of \( V_{QL} \) in (3.58) can be neglected. So \( \hat{H} \) will be an almost perfect estimation of \( H^L \). Solving (3.57) will approximately have \( L' \) as outcome.

We can conclude that for insufficient exploration the relative performance does not change. For high amounts of exploration the estimation will almost be correct resulting in a relative performance that corresponds to \( L' \).

The Estimation Error

To find the minimal exploration we adapt the expressions for the estimation errors of the SI approach with the values for the Q-Learning approach. The error in the estimation \( \tilde{\theta}_{uu} \) is given by:

\[
\tilde{\theta}_{uu,i} = \frac{T^{eeT} P_{n_x+n_y+i}^{T}}{\|P_{n_x+n_y+i}T^{ee}\|_2} (V_{QL} - \sum_{j=n_x+n_y+i+1}^{n_x+n_y+n_u} T^{ee} \tilde{\theta}_{uu,j}).
\]

In the same way for the estimation \( \tilde{\theta}_d \)

\[
\tilde{\theta}_{d,i} = \frac{\gamma_{si}^{T} P_{n_x+n_y+i}^{T}}{\|P_{n_x+n_y+i} \gamma_{si}\|_2} (V_{QL} - T^{ee} \tilde{\theta}_{uu} - \sum_{j=n_x+n_y+i+1}^{n_x+n_y+n_u} \gamma_{si} \tilde{\theta}_{d,j}).
\]

The level of magnitude of the errors \( \tilde{\theta}_{uu,i} \) and \( \tilde{\theta}_{d,i} \) remains the same after rearranging it to \( \hat{H}_{uu} \) and \( \hat{H}_d \).

The Minimal Exploration

To get an approximation of the minimal amount of exploration, we start again with some assumptions. Since \( T^{ee} \) has \( \text{vec}(e_k e_{d}^T) \) as rows, we will assume that \( E\{T_{si}^{eeT} T_{si}^{ee}\} \sim c_1 N \sigma_e^4 \).

Using the definition of \( \omega_k \) in (3.19) we will assume that \( E\{T_{si}^{eeT} V_{QL}\} \sim c_2 N \sigma_e^2 \sigma_r^2 \). We further assume that \( E\{|P_{n_x+n_y+i} \gamma_{si}|^2\} \sim \sigma_e^2 E\{|P_{i} |X_{si}|^2\} \), so that the levels of magnitude of the expected errors are:

\[
E\{\hat{H}_{uu}\} \sim \frac{c_2 \sigma_e^2}{c_1 \sigma_r^2}
\]

\[
E\{\hat{H}_d\} \sim \frac{\sigma_e^2}{\sqrt{(c_3 + c_4 \sigma_e^2) \sigma_r + c_3 \sigma_e \sigma_r^2 + c_6 \sigma_e^3}}.
\]

Both errors in (3.66) will be zero if \( \sigma_r = 0 \). This corresponds with the noise free situation in [16] and [44]. In this situation the only purpose of the exploration is to prevent singularity in the computations of the least squares estimate.

The maximum of \( E\{H_d\} \) can be expressed as \( \sigma_e (1 + \kappa) \), where \( \kappa > 0 \) is some value that depends on the constants and \( \sigma_r \). Without specifying the constants it is impossible to get an idea about the size of \( \kappa \). The only thing we can conclude is:
3.3. THE INFLUENCE OF EXPLORATION

The amount of exploration required by the LQRQL approach is larger than the amount of exploration required by the SI approach!

3.3.4 The Exploration Characteristics

Our main contribution in this chapter is comparing the performances for the SI and LQRQL approach. Especially we focused on the influence of the amount of exploration on these performances. We will summarize our results in this section. For this we define the exploration characteristic as the expected performance as a function of the amount of exploration $\sigma_e$. As a performance measure we will use the relative performance introduced in section 3.2.5. First we will give a description of the similarities and then of the differences between the two solution methods.

The outcome of both methods depends on two estimations; $\hat{B}$ and $\hat{D}$ for the SI approach, $\hat{H}_{uu}$ and $\hat{H}_d$ for LQRQL. These estimations can be viewed as the sum of the correct result and the estimation error (i.e. $\hat{B} = \bar{B} + \bar{B}$), where the exploration and system noise only affect the estimation error. Based on the influence of the exploration on the estimation errors, we can distinguish four types of outcomes. With the increase of the level of exploration we will have the following types of outcome:

I Singularity: No exploration will result in a singularity, so there is no outcome.

II Error Dominance: If the amount of exploration is much too low, but a feedback can be computed, the estimation errors will dominate the outcome. The resulting feedback will approximately be the feedback that was used to generate the data, so the relative performance does not improve.

III Sequence Dependent Outcome: For a certain amount of exploration, the estimation errors do not dominate the outcome but are still too high to be neglected. So the resulting feedback is partly based on the estimation error and partly on the correct value. The outcome will depend on the particular realization of the exploration noise sequence and system noise sequence. Therefore the relative performance can be anything, although it is bounded from below due to the error $\hat{D}$ or $\hat{H}_d$.

IV Correct Estimation: For sufficient exploration the estimation errors can be neglected. The SI approach will result in $L^*$ because the system’s parameters are estimated correctly. The LQRQL will result in $L'$ because the parameters of the Q-function are estimated correctly.

Two exploration characteristics for SI and LQRQL approaches are sketched in figure 3.2. To stress the differences, $\log(\rho^L - 1)$ is shown instead of $\rho^L$. The differences in the four types of outcomes are:

I Singularity: The amount of exploration appears quadratically in the estimations for the QL approach, so it requires more exploration to prevent singularity. This means that the lines in figure 3.2(b) start for higher values of $\sigma_e$ than in figure 3.2(a).
II Error Dominance: Only the value of $\sigma_e$ for which this is the outcome differs.

III Sequence Dependent Outcome: $\bar{D}$ has a maximum value for $\sigma_e < \sigma_e$ and $\bar{H}_d$ for $\sigma_e > \sigma_e$. Also the maximum values are different. So this outcome only differs in the lower bound for the relative performance.

IV Correct Estimation: For the SI approach the relative performance will approach one with the increase in the exploration level. For LQRQL the feedback will approach $L'$ so that the relative performance depends on the feedback that was used to generate the data. The only way to let the relative performance approach one is to do more policy improvement steps. In Figure 3.2b we see that $L_2 = L'_1$, so $L'_2$ is the result after two policy improvement steps when starting with $L_1$.

The main differences between the exploration characteristics in figure 3.2 are the amounts of exploration for which these outcomes occur. It is clear that there is a difference in the type IV outcome, because the SI approach will approximate $L^*$ and the LQRQL approach $L'$. Therefore does the outcome of the LQRQL approach depend on the $L$ and for the SI approach it does not.

The dashed lines in figure 3.2(a) indicate the lower bound on the relative performance due to $\bar{D}$ (the dashed line continues under the solid and bold line). The relative performance will not go below this line, even when an almost optimal feedback like $L_2$ is used. So

![Diagram](image)

(a) The SI exploration characteristic. The arrow with $L^*$ indicates that the characteristic will approach the optimal feedback if the exploration is increased.

![Diagram](image)

(b) The LQRQL exploration characteristic.

Figure 3.2. The Exploration Characteristics. Both figures show $\log(\rho^L - 1)$ as a function of $\sigma_e$ when data was generated using feedback $L_1$ (bold line) and feedback $L_2$ (solid line). The symbols next to these lines indicate the value that is being approximated. The feedback $L_2$ is almost optimal so that $\rho^{L_2}$ is almost one. The dashed line indicates the lower bound on the relative performances due to the errors $\bar{D}$ or $\bar{H}_d$. The grey area indicates outcome III, where any relative performance above the lower bound is possible.
taking \( \sigma_e = \sigma_v \) will not "improve" the feedback. (This amount of exploration will give an improvement for \( L_1 \)). Feedback \( L_2 \) can only be improved by increasing the amount of exploration even more. Figure 3.2(b) shows the same effect for the LQRQL approach due to \( \dot{H}_d \). It takes more exploration to guarantee \( L_2^1 \) than it takes to guarantee \( L_1^1 \). Both approaches have in common that for near optimal feedbacks more exploration is required to guarantee an improvement than just avoiding outcome III.

### 3.4 Simulation Experiments

The purpose of the simulation experiments is to verify the results presented in previous sections and show the exploration characteristics.

#### 3.4.1 Setup

We take a system according to (3.1) with the following parameters:

\[
A = \begin{bmatrix} \bar{A} & \bar{B} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.
\]

(3.67)

For the direct cost (3.2) we take \( S \) to be a unit matrix and \( R = 1 \). For this system the number of parameters that has to be estimated for both approaches equals 6. So taking \( N = 20 \) measurements should be enough for the estimation. The measurements are generated according to (3.6) with

\[
L = \begin{bmatrix} 0.2 & -0.2 \end{bmatrix}, \quad \sigma_v = 10^{-4}.
\]

(3.68)

For this value of \( L \) the closed loop is stable. The solution of the DARE and the optimal feedback are given by:

\[
K^* = \begin{bmatrix} 2.302 & 0.149 \\ 0.149 & 1.112 \end{bmatrix}, \quad L^* = \begin{bmatrix} 0.373 & 0.279 \end{bmatrix}.
\]

(3.69)

Also the relative performances for \( L \) can be computed:

\[
\rho^L_{\text{min}} = 1.469, \quad \rho^L_{\text{max}} = 2.093, \quad \rho^L(x_0) = 1.832.
\]

(3.70)

#### 3.4.2 Exploration Characteristic

We compute the exploration characteristic by doing the same simulation experiment for different values of \( \sigma_e \). To make sure that \( \sigma_e \) is the only parameter that differs, we always use the same realizations of exploration noise and system noise by using the same seeds for the random generator. We vary \( \sigma_e \) from \( 10^{-12} \) to \( 10^5 \).

Figure 3.3(a) shows \( \rho^L_{\text{min}} \), \( \rho^L_{\text{max}} \) and \( \rho^L(x_0) \) for the SI and LQRQL approach for one realization. The exploration intervals for the four types of outcomes can be seen in figure 3.3(a):
Figure 3.3. Simulation Results. The dotted vertical line indicates the system noise level $\sigma_v = 10^{-4}$. The dashed lines are the results for the SI approach and the solid lines are the results for the LQRQL approach.

I SI: $\sigma_e < 10^{-12}$ (not shown in figure 3.3(a)), LQRQL: $\sigma_e < 10^{-7}$.

II SI: $10^{-12} < \sigma_e < 10^{-7}$, LQRQL: $10^{-7} < \sigma_e < 10^{-4} = \sigma_v$. The values of $\rho^L_{\text{min}}$, $\rho^L_{\text{max}}$ and $\rho^L(x_0)$ in figure 3.3(a) agree with the values in (3.70).

III SI: $10^{-7} < \sigma_e < 10^{-4}$, LQRQL: $10^{-4} < \sigma_e < 10^{-2}$. This particular realization results in an increase of the RP for both methods.

IV SI: $\sigma_e > 10^{-4}$, LQRQL: $\sigma_e > 10^{-2}$. The RP for both approaches seem to be one.

In the results in figure 3.3(b) we want to focus on type III and IV outcomes. In figure 3.3(a) the outcomes of type IV are not very clear. Therefore figure 3.3(b) shows $\log(\rho^L_{\text{max}} - 1)$ and not $\rho^L_{\text{max}}$. The outcomes are shown for five different realizations.

III For some realizations $\rho^L_{\text{max}} < \rho^L_{\text{max}}$ and for other realizations $\rho^L_{\text{max}} > \rho^L_{\text{max}}$. This hold for both approaches but not always for the same realizations (not shown in figure 3.3(b), where the lines are not labeled). So if $\rho^L_{\text{max}}$ is high for one approach, it does not imply that it is also high for the other approach.

IV The RP for both approaches are not equal to one, they are close to one! For the SI approach the value of $\rho^L_{\text{max}}$ gets closer to one if the amount of exploration is increased. For LQRQL the value of $\rho^L_{\text{max}}$ does not approach one if the amount of exploration is increased. Instead it approaches $\rho^L_{\text{max}} > 1$. 

(a) $\rho^L_{\text{min}}$, $\rho^L_{\text{max}}$ and $\rho^L(x_0)$ for one realization. 

(b) $\log(\rho^L_{\text{max}} - 1)$ for five realizations.
3.5 Discussion

In this chapter we continued investigating RL in the context of LQR as in [16][44]. In order to make it more realistic we included system noise in our analysis. The proofs of convergence in [16][44] are only valid if there is no noise and sufficient exploration is used.

In this chapter we showed that the system noise determines the amount of exploration that is sufficient. There is a hard threshold level for the amount of exploration required. Just below that threshold the resulting feedback can be anything and even result in an unstable closed loop. This is the amount of exploration that has to be avoided. For the SI approach an alternative method was proposed to deal with the bias resulting from the system noise [21]. The idea is to add a bias towards the optimal solution. The effect of such an approach is that this may reduce the probability of an unstable closed loop for the type III outcome, but this can still not be guaranteed. Therefore avoiding type III outcomes is safer.

If we reduce the amount of exploration even more, we will find the feedback used to generate the data as the optimal solution. Although this is not dangerous, this result is not very useful. If the amount of exploration is reduced even more, no feedback can be computed because of a singularity in the least squares estimation. This effect is also present without noise, so the purpose of exploration in [16][44] is to prevent numerical problems with the recursive least squares estimation. The amount of exploration that is sufficient in that case is determined by the machine precision of the computer.

We also compared the result of LQRQQL with an indirect approach. We observed that the performance of this approach as a function of the amount of exploration is very similar to that of LQRQQL. The main difference is that the threshold level of the amount of exploration required is lower. This means that under the circumstances under which convergence of LQRQQL can be proven, it is wiser to use the indirect approach.

In [32] some additional experiments are described. There, two almost identical data sets are shown, where one data set did not change the feedback and where the other gave an improvement. This indicates that visual inspection of the data does not reveal whether sufficient exploration was used. For the LQRQQL approach we can look at the eigenvalues of $H$. If it has negative eigenvalues, the quadratic Q-function is not positive definite and therefore insufficient exploration was used. For the SI approach such an indication is not available.

We did not look at the influence of the feedback itself, but this can only have an effect for the higher amounts of exploration. Just below the threshold level the feedback determines the probability of an unstable closed loop. Since this situation has to be avoided, this is not of interest. For sufficient exploration the feedback will determine how many policy iteration steps are required. When starting with a good performing feedback only a few steps are required.

Our analysis was based on estimating the magnitudes of the estimation errors. These errors still depend on the number of time steps used. The contribution of the number of time steps on the performance for the indirect approach is described in [27]. The results presented there are very conservative and indicate that a large number of time steps are
required for a guaranteed improvement of the performance. Based on our experiment we see that the amount of time steps required is just a couple of times the number of parameters to estimate.

3.6 Conclusion

We have shown a fair comparison between two different approaches to optimize the feedback for an unknown linear system. For the system identification approach the estimation and optimization are performed separately. For the Q-Learning approach the optimization is implicitly included in the estimation. The comparison is fair because both approaches used the same data, and no other parameters had to be chosen. So the differences in performance are due to the approaches.

The first conclusion is that for insufficient exploration the result of the optimization will be the same as the feedback that was used to generate the data. So no change in feedback does not imply that the feedback is already optimal. This result is a consequence of the noise in the system. This noise introduces a bias in the estimation and when using insufficient exploration, this bias dominates the estimated outcome.

If the exploration is insufficient, but large enough that the resulting feedback will not be the same as the initial feedback, then the resulting performance becomes very unpredictable. The closed loop can be stable and the performance can be improved, but it is also possible that the closed loop becomes unstable. These results have to be avoided and therefore it is very important that sufficient exploration is used.

The second conclusion is that the LQRQL approach can be guaranteed to optimize the feedback. This is the first continuous state space problem with noise, for which a reinforcement learning approach can be guaranteed to work. This is the good news. The bad news is that if the conditions hold and a good outcome can be guaranteed, an alternative approach based on system identification will perform better. So we can not recommend to use the LQRQL approach for the linear quadratic regularization task.