

# Greedy Adaptive Critics for LQR Problems: Convergence Proofs

Tomas Landelius   Hans Knutsson  
tc@isy.liu.se   knutte@isy.liu.se

Department of Electrical Engineering, Computer Vision Laboratory  
Linköping University, 581 83 Linköping, Sweden  
Phone: +46 13 282651, Fax: +46 13 138526

October 4, 1996

## Abstract

A number of success stories have been told where reinforcement learning has been applied to problems in continuous state spaces using neural nets or other sorts of function approximators in the adaptive critics.

However, the theoretical understanding of why and when these algorithms work is inadequate. This is clearly exemplified by the lack of convergence results for a number of important situations. To our knowledge only two such results have been presented for systems in the continuous state space domain. The first is due to Werbos and is concerned with linear function approximation and heuristic dynamic programming. Here no optimal strategy can be found why the result is of limited importance. The second result is due to Bradtke and deals with linear quadratic systems and quadratic function approximators.

Bradtke's proof is limited to ADHDP and policy iteration techniques where the optimal solution is found by a number of successive approximations. This paper deals with greedy techniques, where the optimal solution is directly aimed for. Convergence proofs for a number of adaptive critics, HDP, DHP, ADHDP and ADDHP, are presented.

Optimal controllers for linear quadratic regulation (LQR) systems can be found by standard techniques from control theory but the assumptions made in control theory can be weakened if adaptive critic techniques are employed. The main point of this paper is, however, not to emphasize the differences but to highlight the similarities and by so doing contribute to a theoretical understanding of adaptive critics.

## 1 Introduction

Reinforcement learning is a general and powerful way to formulate complex learning problems. The goal of the system is to maximize, a long terms sum of an instantaneous reward provided by the teacher. In its extremum form it only requires that the teacher can provide a measure of success, i.e. that the system can be given a probability of its mission being completed successfully. This should be a minimum requirement for a definition of any relevant problem, including the case with uncertain teachers.

A number of success stories can be found in the literature where reinforcement learning have been applied to problems in continuous state spaces. The theoretical understanding of why and when these algorithms work is however inadequate. In the past, proofs of convergence for reinforcement learning algorithms have been restricted to finite-state systems. To our knowledge there are only two exceptions. The first is due to Werbos [Werbos, 1990] where he shows convergence for a linear system in a continuous state space. However also the instantaneous payoff function is linear which results in the lack of an optimal strategy. This makes the result of minor interest. The second result was presented by Bradtke [Bradtke, 1993] and deals with linear quadratic regulation (LQR) problems. This proof is limited to a special form of reinforcement learning known as Q-learning. It is based on *policy iterations*, a nested loop scheme, where a

number of policies must be evaluated, each for a sufficiently long time period, in order to obtain the optimal one by successive approximations. In this paper we present proofs of convergence for a number of reinforcement learning algorithms that learn the optimal policy by *certainty equivalence control*. In contrast to policy iteration the policy is here updated, in a single loop, towards the optimal one at each time step.

Both the previous result by Bradtke and our investigations concern the class of problems referred to as LQR problems. Even though these systems are simple in comparison to the systems in real life it should still be pointed out that a deeper theoretical understanding of reinforcement learning in continuous state spaces will be invaluable in the design of practical systems.

It is only recently that the close relationship between reinforcement learning, as approximate dynamic programming, and adaptive optimal control has been recognized [Sutton et al., 1991]. An introduction to these relations is provided by White and Sofge [White and Sofge, 1992]. The major difference between the approaches lies in that some of the assumptions made in optimal control can be relaxed when reinforcement learning is employed. In our view the main point of this paper is, however, not to emphasize on the differences but to exemplify the potential of bringing theoretical results from control theory into the study of reinforcement learning algorithms. It is our hope that this paper will help produce new insights in the control theoretic as well as the reinforcement learning community.

## 2 Setting the stage

To provide the background and establish the notation, this section will describe the formulation of a general reinforcement learning problem. It also presents the method of dynamic programming which is the foundation for all the adaptive critics to be presented later. Finally, details are given for the specific class of problems for which the convergence results in this paper are valid, the linear quadratic regulation problems.

### 2.1 The mission

We believe that almost all learning problems can be stated as maximization problems where the entity to maximize is the expected value of a utility function  $F$ . The task is then to find the best probability distribution for the response or output signal  $p^*(y)$  at each time instance

$$p^*(y_j) = \arg \max_{p(y_j)} E\{F(\{u_i, y_i\}, u_j, p(y_j), p(u_k, y_k))\}. \quad (1)$$

All probability distributions in the equation above are conditional with respect to the appropriate history of  $u$  and  $y$ . The function  $F$  depends on the probability for the current output  $y_j$ , what has happened up til the actual time instance  $j$ , i.e.  $u_j$  and  $\{u_i, y_i\}$ ,  $i < j$ , as well as on the probability for future responses  $p(u_k, y_k)$ ,  $k > j$ .

The search over all probability distributions only becomes feasible when it is restricted to a parameterized family of distributions. The simplification is often brought even further by employing a maximum likelihood approach. In this case the goal is to determine the response that maximizes the expected value of  $F$ :

$$y^* = \arg \max_{y_j} E\{F(\{u_i, y_i\}, u_j, y_j, p(u_k, y_k))\}. \quad (2)$$

In unsupervised learning we often have complete knowledge of the function  $F$  since we ourselves have invented the measure against which the self-organizing structure should be evaluated. In this case the function  $F$  is typically based on information or energy aspects.

The situation is a bit different in supervised learning. Here we know the correct response  $y^*$  and often also the form of the function  $F$ . If not, the assumption is that if we get close to the correct targets this will result in a high value of  $F$ . Typically one decides for  $F$  to be the negative sum of squared errors between the generated responses and the correct ones.

In most reinforcement learning formulations it is assumed that the system under consideration is Markovian, i.e. that everything that has happened in the past can be described by a state vector associated with the current time instance. This means that we can bring the information in the set  $\{u_i, y_i, u_j\}$  into a single state vector  $x_j$ . Another standard assumption is that  $F$  can

be written as a sum of instantaneous utilities or rewards,  $r(x_j, y_j)$ .

$$F = F(x_j, y_j) = \sum_{k=j}^N r(x_k, y_k, k).$$

This means that the task is defined as to maximize the expected long term payoff. The time dependence is usually modelled as exponential decay of future rewards making rewards in the far away future less important

$$F = \sum_{k=j}^N r(x_k, y_k, k) = \sum_{k=j}^N \gamma^{k-j} r(x_k, y_k). \quad (3)$$

Now, let us rule out the possibilities of either having complete knowledge of the function  $F$  or having a set of training data  $\{x_i, y_i\}$  telling us what the correct output is for a given input. If we instead focus on the case where the only information available is a measure of the instantaneous reward,  $r$ , we are facing a reinforcement learning problem. Then there are broadly speaking two ways to approach the problem of finding the optimal policy differing in what function to parameterize.

1. Back-propagation of utility. Parameterizes the policy  $y_j = g(x_j, w)$  and finds the maximum likelihood solution in eq. 3 in terms of the parameter vector  $w$

$$w^* = \arg \max_w E \left\{ \sum_{k=j}^N r(x_k, g(x_k, w), k) \right\}. \quad (4)$$

2. Adaptive critics. In this case the expected value of  $F$  in eq. 3 is parameterized,  $Q(x, y, w) = E\{F(x, y)\}$ . Furthermore the function  $r$  is unknown and only samples from  $r$  can be obtained. The maximum likelihood response  $y^*(x, w) = \arg \max_y Q(x, y, w)$  can then be used in the search for optimal parameters

$$w^* = \arg \max_w E \left\{ \sum_{k=j}^N r(x_k, y^*(x_k, w), k) \right\}. \quad (5)$$

Note the resemblance between the two problem formulations in eqs. 4 and 5. They differ only in which function to parameterize. In back-propagation of utility, the parameterization of  $y = g(x, w)$  induces a parameterization of  $E\{F(x, y)\}$ . Adaptive critics parameterize  $E\{F\} = Q(x, y, w)$  which induces a parameterization of the response  $y(x)$ . However, there is a difference in the way the optimal parameters are searched for.

Methods involving back-propagation of utility, such as explicit criterion minimization (ECM) [Åström and Wittenmark, 1989] and model predictive control (MPC) [Morari, 1993], rely on the existence of a complete model of the environment and the system. The structure and the parameters of the instantaneous reward  $r$  is often known a priori. Together these models turn the sum in eq. 3 into a possibly non-linear optimization problem in terms of the system parameters. Once the solution is found, the parameters of the environmental model are estimated and used for calculation of the optimal system parameters. These methods do not handle stochastic systems where there might be several equally good outputs for a single input. They might also run into trouble if the model used to find the optimal system does not fit the true environment.

Adaptive critics, on the other hand, approximate *dynamic programming* (DP), the only exact and “efficient” method for finding optimal strategies over time in noisy non-linear environments. The problem with true DP is that even if it is efficient compared to other procedures the cost of running DP is proportional, or worse, to the number of possible states for the environment, which grow exponentially with the number of dimensions for a fix quantization of the state space. This is why adaptive critics in continuous domains by necessity need to rely on parameterized functions and hence becomes approximations to DP.

## 2.2 Dynamic programming

One of the most important strengths of DP and adaptive critics is that they are applicable also to problem formulations involving stochastic systems. This is because the parameterized function  $Q(x, y, w)$  can be used as a probability distribution,  $p(y | x, w)$ . In the following we will

however restrict our considerations to deterministic systems and only produce the most likely response.

We will from now on refer to the system as the total of the environment,  $f$ , and the regulator or policy,  $g$ . The system is assumed to be Markovian in terms of the state vector  $x$  and the regulator output  $y$ , i.e. the next state of the system only depends on the previous state and the present output from the regulator

$$x_{k+1} = f(x_k, y_k), \quad y_k = g(x_k). \quad (6)$$

This formulation implies that the regulator is provided with a valid state vector. If this is not the case we suffer from *perceptual aliasing* and need to somehow extract the hidden state information from whatever signals available. The standard attempt to solve this problem is to extend the state space representation by introducing some type of memory [Watkins, 1989]. One classic example is to build an observer using a Kalman filter.

A key function in dynamic programming is the value function, here denoted by  $V(x)$ , which model the long term reward collected using a regulator  $g$  and starting out with the system in state  $x$

$$\begin{aligned} V_g(x_j) &= F(x_j, g(x_j)) = \sum_{k=j}^N r(x_k, g(x_k)) \\ &= r(x_j, g(x_j)) + V_g(f(x_j, g(x_j))). \end{aligned}$$

Here the decay factor is set to one for simplicity and without loss of generality. An optimal regulator is now defined as a mapping  $g^*$  that will result in the largest possible collection of rewards

$$V_*(x) \geq V_g(x), \quad \forall \{g, x\}. \quad (7)$$

This also means that there may exist several different optimal regulators, but they all induce the same optimal value function  $V_*$ . Dynamic programming is a number of techniques that let us find the optimal value function as well as the optimal regulator, or policy as it will most frequently be referred to from now on. Note that in order to make practical use of the following discussion for systems working in continuous state spaces a parameterization of either  $V$  or  $g$  is necessary.

By combining the recursive formulation of the value function in eq. 7 with the definition of the optimal value function in eq. 7 we see that using the optimal regulator, i.e. following the optimal policy,  $g^*$  will give rise to maximal long term reward

$$V_g(x) \leq V_*(x) = r(x, g^*(x)) + V_*(f(x, g^*(x))). \quad (8)$$

This now gives us an expression for the optimal response since the equation above states that the output from an optimal regulator  $y^*$  must maximize

$$y^* = g^*(x) = \arg \max_y \{r(x, y) + V_*(f(x, y))\}. \quad (9)$$

Now, combining eqs. 9 and 8 we obtain what is known as Bellman's optimality equation

$$V_*(x) = \max_y \{r(x, y) + V_*(f(x, y))\}. \quad (10)$$

This equation can be stated in a more compact form by the introduction of the Q-function [Denardo, 1967, Watkins, 1989]. This function describes the long term reward if we output an arbitrary  $y$  this time instance and then use our regulator  $g$  to generate future outputs

$$Q_g(x, y) = r(x, y) + V_g(f(x, y)). \quad (11)$$

From this definition we see that  $V_g(x) = Q_g(x, g(x))$ . We then find that the optimal Q-function must obey the following

$$\begin{aligned} Q_*(x, y) &= r(x, y) + V_*(f(x, y)) \\ &= r(x, y) + Q_*(f(x, y), g^*(f(x, y))). \end{aligned}$$

An advantage with this formulation is that now we can find optimal responses without reference to the environment model  $f$

$$y^* = g^*(x) = \arg \max_y Q_*(x, y). \quad (12)$$

It also makes Bellman's optimality equation become especially simple

$$V_*(x) = \max_y Q_*(x, y). \quad (13)$$

There are basically two ways of learning the optimal policy  $y = g^*(x)$ . Policy iteration and greedy iterations which is also known as certainty equivalence control. These two approaches will be discussed next.

## Policy iteration

In this case we iterate nested loops alternating between policy *evaluation* and policy *improvement*. The inner loop evaluates the current fixed policy  $g$  and finds its corresponding value function  $V_g$  which satisfies

$$V_g(x) = r(x, g(x)) + V_g(f(x, g(x))). \quad (14)$$

Once this function is estimated the policy is updated in an outer loop to be in accordance with the new estimate of  $V$

$$g(x) = \arg \max_y \{r(x, y) + V_g(f(x, y))\}. \quad (15)$$

These two loops are then iterated till convergence. It has been shown that this procedure will indeed converge to the optimal value function and the optimal policy if the two steps above can be solved correctly [Howard, 1960].

## Greedy iteration

In this case both the steps involved in policy iteration are performed together. Hence, we directly try to find the optimal value function and the optimal policy using a bootstrapping procedure where the current policy is always considered to be the optimal one, i.e. that  $\hat{g}^* = g^*$ . The search is often done using iterative techniques and in each iteration of a single loop, the parameterized estimate  $\hat{V}^*$  is updated as to satisfy Bellman's optimality equation and this results in the policy being updated to what would be the optimal policy if  $\hat{V}^*$  was the optimal value function  $V_*$

$$\begin{aligned} \hat{V}_*(x) &= r(x, \hat{g}^*(x)) + \hat{V}_*(f(x, \hat{g}^*(x))) \\ \hat{g}^*(x) &= \arg \max_y \{r(x, y) + \hat{V}_*(f(x, y))\}. \end{aligned}$$

This motivates why the method is also called certainty equivalence control since the current estimate of the optimal value function is treated as if it was the optimal one in the derivation of the new estimate of the optimal policy. In this case the two equations above are evaluated in reverse order because the focus is on the parameterized policy  $\hat{g}^*(x, w)$  and the update of the parameters  $w$  induce an updated version of  $\hat{V}_*(x)$ .

Both previous proofs of convergence for continuous state spaces has been concerned with policy iteration. One treated linear and one dealt with quadratic instantaneous rewards. In the next section we will prove convergence for a number of adaptive critics using greedy iterations applied to LQR systems.

## 2.3 Linear quadratic regulation

In the following we will concentrate on what is known as the problem of discrete-time linear quadratic regulation (LQR). Consider the discrete-time, continuous multi-variable system where the system, consisting of the environment and the regulator, are described by *linear* mappings

$$x_{k+1} = f(x_k, y_k) = Ax_k + By_k \quad (16)$$

$$y_k = g(x_k) = Lx_k. \quad (17)$$

where the matrices  $A \in \mathbb{R}^{n \times n}$  and  $B \in \mathbb{R}^{n \times m}$  constitute a controllable pair  $(A, B)$ . Controllability means that there exists a sequence of control outputs  $\{y_j\}_{j=1}^N$  that transfers every initial state  $x_0$  to any final state  $x_N$ . Formally this requirement can be stated as

$$\text{rank} [A \ AB \ A^2B \ \dots \ A^{n-1}B] = n. \quad (18)$$

It can then be shown that this also means that a stabilizing feedback matrix  $L \in \mathbb{R}^{m \times n}$  can be found so that the matrix  $A + BL$  has all its eigenvalues in the open unit disc. The instantaneous cost (negative reward) associated with this system is a *quadratic* form.

$$r(x_k, y_k) = x_k^T Q x_k + y_k^T R y_k, \quad (19)$$

where  $Q$  is positive semidefinite and  $R$  is positive definite, denoted by  $Q \geq 0$  and  $R > 0$  respectively. We consider the matrices  $Q$  and  $R$  to be symmetric since a quadratic form with a non symmetric matrix can be restated as one with a symmetric matrix.

Note that we can restrict our treatment, without lack of generality, to the case where the instantaneous cost is described by a block diagonal matrix. This is because the coordinate transformation  $v = y + R^{-1}S^T x$  will restate a problem with a full cost matrix, where the off diagonal block is given by the matrix  $S$ , as a block diagonal problem in terms of the variables  $x$  and  $v$ . In this case the matrices  $A$  and  $Q$  are replaced with  $A - BR^{-1}S^T$  and  $Q - SR^{-1}S^T$  respectively. Naturally the full cost matrix must also in this case be positive semidefinite.

The task is now to find the feedback system  $g$  that maximize the reward (minimize the cost) in the *long run*. This can more specifically be stated as the minimization of the discounted sum of the instantaneous costs given by

$$V_g(x_k) = \sum_{j=k}^{\infty} \gamma^{j-k} r(x_j, y_j) = \sum_{j=k}^{\infty} \gamma^{j-k} (x_j^T Q x_j + y_j^T R y_j), \quad (20)$$

where  $0 \leq \gamma \leq 1$  is the discount factor.

It can be shown that the optimal feedback is linear,  $g^*(x) = L^* x$ , and with linear feedback we have  $x_{k+1} = (A + BL)x_k = D x_k$  [Åström and Wittenmark, 1989]. For such a system the value function  $V$  modelling the long term cost becomes a quadratic form

$$\begin{aligned} V(x_k) &= \sum_{j=k}^{\infty} \gamma^{j-k} r(x_j, Lx_j) &&= \sum_{j=k}^{\infty} \gamma^{j-k} (x_j^T Q x_j + y_j^T R y_j) \\ &= \sum_{i=0}^{\infty} \gamma^i (x_{i+k}^T Q x_{i+k} + y_{i+k}^T R y_{i+k}) &&= \sum_{i=0}^{\infty} \gamma^i x_{i+k}^T (Q + L^T R L) x_{i+k} \\ &= x_k^T \left[ \sum_{i=0}^{\infty} \gamma^i (D^T)^i (Q + L^T R L) D^i \right] x_k = x_k^T K x_k. \end{aligned}$$

This sum will be convergent since the matrix  $D$  has all its eigenvalues in the open unit disc. Since it is a sum of positive numbers it must be the case that the matrix  $K \geq 0$ . Again, we consider the  $K$  in the quadratic form to be symmetric.

For a linear system with a quadratic reward also the Q-function becomes a quadratic form:

$$\begin{aligned} Q(x_i, y_i) &= r(x_i, y_i) + V(x_{i+1}) = (x_i^T \ y_i^T) \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + x_{i+1}^T K x_{i+1} \\ &= (x_i^T \ y_i^T) G \begin{pmatrix} x_i \\ y_i \end{pmatrix} + (Ax_i + By_i)^T K (Ax_i + By_i) \\ &= (x_i^T \ y_i^T) \left[ G + \begin{pmatrix} A^T \\ B^T \end{pmatrix} K (A \ B) \right] \begin{pmatrix} x_i \\ y_i \end{pmatrix} = (x_i^T \ y_i^T) H \begin{pmatrix} x_i \\ y_i \end{pmatrix}. \end{aligned}$$

This shows that also the matrix  $H$  describing the Q-function will be positive semidefinite since it is a sum of  $G \geq 0$  (block-diagonal with the blocks  $Q \geq 0$  and  $R > 0$ ) and a term of the form  $F^T K F$  which is also positive semidefinite since  $K \geq 0$ .

### 3 Adaptive critics

As stated previously there is a difference between adaptive critics and techniques from optimal control using back-propagation of utility. In optimal control one often poses an optimization

problem that is solved for optimal feedback parameters given a system model. The solution is then fed with the estimated system parameters to give optimal parameters for the system at hand.

Adaptive critics on the other hand try to model the value function or one of its relatives describing the long term reward. This estimate can then be used to find optimal feedback in a number of ways as will be described later. Another difference is that some of the adaptive critics do not need to estimate an explicit model of the environment and that stochastic systems can be treated within the framework. We will however, as previously stated, only consider the deterministic LQR problem in this paper.

There are several variants of adaptive critics in the literature. Two of the most common are TD-methods [Sutton, 1988] and Q-learning [Watkins, 1989]. These two methods, as well as most of the adaptive critics presented in the literature so far, has been classified in four categories by Werbos [Werbos, 1992]. The classification is based on the choice of function to model and parameterize, see table 1.

	Method	Model
HDP	Heuristic Dynamic Programming	$V(x)$
DHP	Dual Heuristic Programming	$\partial V/\partial x$
ADHDP	Action Dependent HDP	$Q(x, y)$
ADDHP	Action Dependent DHP	$\partial Q/\partial x$ and $\partial Q/\partial y$

Table 1: The relationship between adaptive critics and their models.

Each of these adaptive critics will be given a short presentation in terms of what estimates and parameterizations are needed to produce targets for an algorithm that learns the model of the appropriate value function or its gradient. At the end of each section a proof of convergence for the method applied to the LQR problem will be given.

### 3.1 Heuristic Dynamic Programming

Heuristic DP, which includes temporal difference (TD) methods, is a method for estimating the value function  $V$ . If the policy is not fix, but updated with policy iteration or greedy iteration, the value function corresponding to the optimal policy can be estimated. Estimating the value function for a given policy only requires samples from the instantaneous reward function  $r$  while models of the environment and the instantaneous reward are needed to find the value function corresponding to the optimal policy. As seen earlier the value function for a policy  $g$  can be defined recursively as

$$V(x) = r(x, g(x)) + V(f(x, g(x))).$$

We can use the right hand side of this equation as targets  $d$  for any supervised learning algorithm that tries to approximate  $V$  with a parameterized model  $V(x, w)$ :

$$d(r, f, w) = r(x, g(x)) + V(f(x, g(x)), w). \quad (21)$$

Hence, we look for a new parameter vector  $w'$  that minimizes some error function, e.g. the expected squared error:

$$w' = \arg \min_{w'} E\{|V(x, w') - d(r, f, w)|^2\}. \quad (22)$$

Since a new parameter vector defines new targets this is a moving target problem. Because of this the issue of convergence becomes even more problematic than is usually the case with non-linear optimization procedures.

In order to find the parameters for the optimal value function we can use policy iterations or greedy iterations to update our policy. In both cases we need to parameterize the policy  $g(x) = g(x, u)$  and find the proper response parameters according to Bellman's optimality equation using our estimate of  $V$ :

$$u^* = \arg \max_u \{r(x, g(x, u)) + V(f(x, g(x, u)), w)\}. \quad (23)$$

One way to search for the optimal parameters is to employ a gradient algorithm that calculates  $\partial V_{g(u)}(x)/\partial u$ . In these calculations the requirement for a number of models becomes evident:

$$\frac{\partial V}{\partial u} = \frac{\partial r}{\partial g} \frac{\partial g}{\partial u} + \frac{\partial V}{\partial f} \frac{\partial f}{\partial g} \frac{\partial g}{\partial u}. \quad (24)$$

We see that we need models of three derivatives besides the known model of the derivative of our policy  $g$  with respect to its parameters  $u$ . The three models are the derivative of the instantaneous reward  $r$  with respect to the response  $g$ , that of the value function  $V$  with respect to the next state vector  $f$ , and also the derivative of the environment  $f$  with respect to the response  $g$ . These derivatives can be obtained e.g. by differentiating parameterized models of the functions  $r(x, y)$ ,  $V(x)$ , and  $f(x, y)$ .

Now, let us turn to the LQR problem and look at the parameterization of the different functions involved. In this case we will parameterize four functions according to:

$$V(x) = x^T K x \quad r(x, y) = x^T Q x + y^T R y \quad (25)$$

$$f(x, y) = A x + B y \quad y = g(x) = L x. \quad (26)$$

The parameters,  $Q, R$ , of the instantaneous cost and the parameters,  $A, B$ , of the environment can be obtained using any standard identification algorithm. The parameters of the value function can then be estimated by minimization of the moving target problem in eq. 22 through greedy, i.e. certainty equivalence, control.

First, let us look at how to generate responses. Use the expressions for the parameterized functions in eqs. 25 and 26 to obtain the derivatives needed in eq. 24. In the LQR case we obtain the maximum of eq. 23 by solving  $\partial V/\partial u = 0$  for  $u$ . Since we have  $\partial g/\partial u \neq 0$  we can simplify this equation to yield:

$$\begin{aligned} 0 &= \left( \frac{\partial r}{\partial g} + \frac{\partial V}{\partial f} \frac{\partial f}{\partial g} \right) \frac{\partial g}{\partial u} = \frac{\partial r}{\partial g} + \frac{\partial V}{\partial f} \frac{\partial f}{\partial g} \\ &= 2y^T R + 2(Ax + By)^T K B = y^T (R + B^T K B) + x^T A^T K B. \end{aligned}$$

The second row comes from the facts that  $g = y$  and that the next state,  $f = Ax + By$ , is assigned the value  $V(f) = f^T K f$  which means that  $\partial V/\partial f = 2f^T K$ . The equation above let us solve for the proper response

$$y = g(x, u) = -(R + B^T K B)^{-1} B^T K A x = L x, \quad (27)$$

where we view the parameter vector  $u$  as the vectorized version of the matrix  $L$ , i.e.  $u = \text{vec}(L)$  according to the definition of the  $\text{vec}$ -function in the appendix section A.2. Note that the parameterization of  $V(K), r(Q, R)$ , and  $f(A, B)$  induces a parameterization of the policy  $g(K, R, A, B)$ .

Now, to the estimation of the parameters of the optimal value function. We start out with a gradient algorithm to solve the minimization problem in eq. 22 with respect to  $K$ , the matrix corresponding to the parameter vector  $w$ :

$$\begin{aligned} K_{i+1} &= K_i - \eta \frac{\partial}{\partial K'} E \{ |x_i^T K' x_i - d(r, f, K_i)|^2 \}_{K'=K_i} \\ &= K_i - \eta E \left\{ \frac{\partial}{\partial K'} |x_i^T K' x_i - d(r, f, K_i)|^2 \right\}_{K'=K_i} \\ &= K_i - 2\eta E \{ x_i x_i^T (x_i^T K_i x_i - d(r, f, K_i)) \} \\ k_{i+1} &= k_i - \alpha E \{ v_i (v_i^T k_i - d(r, f, K_i)) \}. \end{aligned} \quad (28)$$

Between the last two lines in the equation above we introduce the notation  $v = \text{vec}(x x^T)$ ,  $k = \text{vec}(K)$ ,  $\alpha = 2\eta$ , and use the fact that a quadratic form can be expressed as a scalar product as described in section A.2 of the appendix.

The target for the HDP learning algorithm was defined in eq. 21. Using the parameterization for our system given by eqs. 25 and 26 the target can be expressed as:

$$\begin{aligned} d(r, f, K_i) &= r(x, y) + V(f(x, g(x))) \\ &= x^T Q x + y^T R y + f^T K_i f \\ &= x^T Q x + y^T R y + (Ax + By)^T K_i (Ax + By). \end{aligned}$$

The use of the vec-function allows us to replace the expectation value in eq. 28 with the instantaneous value of the derivative. By so doing we get a stochastic approximation of the gradient algorithm:

$$\begin{aligned}
k_{i+1} &= k_i - \alpha v_i(v_i^T k_i - d(r(x_i, y_i), x_{i+1}, K_i)) \\
k_{i+1} &= k_i - \alpha v_i(v_i^T k_i - [x_i^T Q x_i + y_i^T R y_i + (A x_i + B y_i)^T K_i (A x_i + B y_i)]) \\
&= k_i - \alpha v_i(v_i^T k_i - [x_i^T (Q + L_i^T R L_i) x_i + x_i^T (A + B L_i)^T K_i (A + B L_i) x_i]) \\
&= k_i - \alpha v_i v_i^T \text{vec}(K_i - [Q + L_i^T R L_i + (A + B L_i)^T K_i (A + B L_i)]).
\end{aligned} \tag{29}$$

Note that all information needed by HDP to produce the target is given by the current reward and the value function evaluated in the next system state. Again, the step between the last two lines above comes from the fact that a quadratic form can be stated as a scalar product. Now, let us look at the update equation in the mean

$$E\{\text{vec}(K_{i+1})\} = E\{\text{vec}(K_i)\} - \alpha C_v E\{\text{vec}(\Delta K_i)\}, \tag{30}$$

where  $C_v = E\{v v^T\} > 0$  is a symmetric covariance matrix thanks to the introduction of the vec-function in the earlier stages. From section B.1 in the appendix we have that if a sequence  $w_{i+1} = w_i + \alpha \Delta w_i$  converges then so does the sequence  $w_{i+1} = w_i + \alpha C \Delta w_i$  if  $C > 0$  and is symmetric.

Stating equation 30 in terms of matrices, with  $W = E\{K\}$  and  $C_v = I$ , results in the following equation for the update of the value function parameters in the mean:

$$W_{i+1} = W_i - \alpha [W_i - (Q + L_i^T R L_i + (A + B L_i)^T W_i (A + B L_i))]. \tag{31}$$

That this sequence indeed converges to the parameters for the optimal value function  $V_*(x) = x^T K^* x$  is proved by theorem B.1 in the appendix, section B.2. Hence, we have proven that the stochastic approximation algorithm in eq. 29 converges, in the mean, to the matrix constituting the optimal value function for the linear quadratic system described in eqs. 25 and 26.

### 3.2 Dual Heuristic Programming

Dual heuristic programming is a method for estimating the gradient of the value function,  $\partial V / \partial x$ , rather than  $V$  itself. In order to do this we need samples from a function describing the gradient of the instantaneous reward function  $\partial r / \partial x$  as well as a differentiable model of the environment. To derive the update formula for DHP we start out with recursive formulation for the value function and then differentiate it with respect to the state vector  $x$ :

$$\begin{aligned}
V(x) &= r(x, g(x)) + V(f(x, g(x))) \\
\frac{\partial V}{\partial x} &= \frac{\partial r}{\partial x} + \frac{\partial r}{\partial g} \frac{\partial g}{\partial x} + \frac{\partial V}{\partial f} \left( \frac{\partial f}{\partial x} + \frac{\partial f}{\partial g} \frac{\partial g}{\partial x} \right).
\end{aligned} \tag{32}$$

The right hand side of eq. 32 can be used as target  $d$  for a gradient algorithm that tries to approximate  $V_x = \partial V / \partial x$  with a parameterized model  $V_x(x, w)$ :

$$d(r_x, r_g, f_x, f_g, f, w) = r_x + r_g g_x + V_f (f_x + f_g g_x). \tag{33}$$

Differentiation with respect to a variable is denoted with a subscript. We look for a parameter vector  $w$  that minimizes the expected squared error in the moving target problem:

$$w' = \arg \min_{w'} E\{|V_x(x, w') - d(r_x, r_g, f_x, f_g, f, w)|^2\}. \tag{34}$$

The gradient of the value function corresponding to the optimal policy can be found by means of policy iterations or greedy iterations. In order to complete the greedy step we need to parameterize our policy  $g(x) = g(x, u)$  and solve Bellman's optimality equation using our current estimate of  $\partial V / \partial x$ . As seen in the previous section the optimal parameters can for example be found by a gradient algorithm based on  $\partial V / \partial u$ :

$$\frac{\partial V}{\partial u} = \frac{\partial r}{\partial g} \frac{\partial g}{\partial u} + \frac{\partial V}{\partial f} \frac{\partial f}{\partial g} \frac{\partial g}{\partial u}. \tag{35}$$

In the calculations of this gradient the expression for  $\partial V / \partial f$  turns up and in HDP we needed to differentiate our model of  $V$  in order to obtain it. In DHP this is exactly the expression we try to model.

Again we assume that the other derivatives can be found by differentiation of models estimated for parameterized versions of the instantaneous reward  $r(x, y)$ , and the next state  $f(x, y)$ . For the specific case that we are interested in here, the LQR problem, the parameterized functions look like:

$$\frac{\partial V(x)}{\partial x} = 2x^T K \quad r(x, y) = x^T Q x + y^T R y \quad (36)$$

$$f(x, y) = Ax + By \quad y = g(x) = Lx. \quad (37)$$

We now suppose that the parameters of the last two has been estimated through some standard identification algorithm and instead concentrate on how the parameters of the gradient of the value function can be estimated by minimization of the moving target problem in eq. 34. We will do this greedily, i.e. using certainty equivalence control. From the discussion of the HDP algorithm we have already obtained the expression for the greedy response

$$y = g(x, u) = -(R + B^T K B)^{-1} B^T K A x = Lx, \quad (38)$$

where we recognize  $K$  as the matrix in our parameterized version of  $\partial V/\partial x$  in eq. 36. Again we view the parameter vector  $u$  as the vectorized version of the matrix  $L$ , i.e.  $u = \text{vec}(L)$  according to the definition of the  $\text{vec}$ -function in the appendix, section A.2.

Let us show how to estimate the parameters for the gradient of the optimal value function. Start out with a gradient algorithm to solve the minimization problem in eq. 34 with respect to  $K$ , the matrix corresponding to the parameter vector  $w$ :

$$\begin{aligned} 2K_{i+1} &= 2K_i - \eta \frac{\partial}{\partial K'} E\{|2x_i^T K' - d(r_x, r_g, f_x, f_g, f, K_i)|^2\}_{K'=K_i} \\ &= 2K_i - \eta E \left\{ \frac{\partial}{\partial K'} |2x_i^T K' - d(r_x, r_g, f_x, f_g, f, K_i)|^2 \right\}_{K'=K_i} \\ &= 2K_i - \alpha E \{x_i(2x_i^T K_i - d(r_x, r_g, f_x, f_g, f, K_i))\}. \end{aligned} \quad (39)$$

Here the constant  $\alpha = 2\eta$  was introduced between the last two lines.

The target for the DHP learning algorithm was defined in eq. 33. Using the parameterization for our system given by eqs. 36 and 37 the target can be expressed as:

$$\begin{aligned} d(r_x, r_g, f_x, f_g, f, K_i) &= r_x + r_y g_x + V_f(f_x + f_y g_x) \\ &= 2x^T Q + 2y^T R L + 2f^T K_i (A + B L) \\ &= 2x^T Q + 2y^T R L + 2(Ax + By)^T K_i (A + B L). \end{aligned}$$

Substituting this into eq. 39 and replacing the expectation value with the instantaneous value of the derivative we get the stochastic approximation version of the gradient algorithm:

$$\begin{aligned} 2K_{i+1} &= 2K_i - \alpha x_i(2x_i^T K_i - d(r_x(i), r_y(i), f_x(i), f_y(i), x_{i+1}, K_i)) \quad (40) \\ &= 2K_i - \alpha x_i(2x_i^T K_i - [2x_i^T Q + 2y_i^T R L_i + 2(Ax_i + By_i)^T K_i(A + B L_i)]) \\ K_{i+1} &= K_i - \alpha x_i x_i^T (K_i - [Q + L_i^T R L_i + (A + B L_i)^T K_i(A + B L_i)]). \end{aligned}$$

Note that DHP needs samples from the derivative of the reward, a model for the derivative of the environment, and samples from the derivative of the value function in the next system state to produce the target. Let us look at the convergence of the update rule in the mean

$$E\{K_{i+1}\} = E\{K_i\} - \alpha C_x (E\{K_i\} - E\{[Q + L_i^T R L_i + (A + B L_i)^T K_i(A + B L_i)]\}), \quad (41)$$

where  $C_x = E\{xx^T\} > 0$  is a symmetric covariance matrix. From section B.1 in the appendix we have that if a sequence  $w_{i+1} = w_i + \alpha \Delta w_i$  converges then so does the sequence  $w_{i+1} = w_i + \alpha C \Delta w_i$  if  $C > 0$  and is symmetric.

Restating equation 41 in terms of matrices, with  $W = E\{K\}$  results in the same update rule in the mean, as the one for HDP in the previous section

$$W_{i+1} = W_i - \alpha [W_i - (Q + L_i^T R L_i + (A + B L_i)^T W_i (A + B L_i))]. \quad (42)$$

That this sequence indeed converges to the parameters in the optimal value function  $V_*(x) = x^T K^* x$  is proved in theorem B.1 found in section B.2 of the appendix. Since the same parameter matrix describes the gradient  $\partial V/\partial x = 2x^T K$  we have proved that the update formula in eq. 40 converges, in the mean, to the matrix constituting the gradient of the optimal value function for the linear quadratic system described by eqs. 36 and 37.

### 3.3 Action Dependent HDP

Action dependent HDP, which is also known as Q-learning, is a method for estimating the Q-function for any policy, optimal or non-optimal. In difference with HDP and DHP this method only requires samples from the instantaneous reward function  $r$ . The right hand side of eq. 11, defining the Q-function, is here used as target for a stochastic gradient algorithm that approximates  $Q(x, y)$  with a parameterized model  $Q(x, y, w)$ :

$$d(r, f, g(f), w) = r(x, y) + Q(f(x, y), g(f(x, y)), w). \quad (43)$$

We use a stochastic gradient method to search for the parameter vector that minimize the expected squared error:

$$w' = \arg \min_{w'} E\{|Q(x, y, w') - d(r, f, g(f), w)|^2\}. \quad (44)$$

Here we use greedy iterations to update our policy in order to find the optimal Q-function. The policy is parameterized as  $g(x) = g(x, u)$ . The proper response is given by Bellman's optimality equation in terms of the Q-function,  $y^* = \arg \max_y Q(x, y)$ . A solution can be obtained by employing a gradient algorithm based on  $\partial Q / \partial y$ .

When applied to the LQR problem suitable parameterizations of the Q-function, the reward  $r$ , and the environment (next state),  $f$ , are given by:

$$Q(x, y) = (x^T \ y^T) \begin{pmatrix} H_{xx} & H_{xy} \\ H_{yx} & H_{yy} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad r(x, y) = x^T Qx + y^T Ry \quad (45)$$

$$f(x, y) = Ax + By \quad y = g(x) = Lx. \quad (46)$$

We can now use the parameterized version of  $Q$  to find optimal responses for the LQR system by solving  $\partial Q / \partial y = 0$

$$\begin{aligned} \frac{\partial Q}{\partial y} &= \frac{\partial}{\partial y} (x^T \ y^T) \begin{pmatrix} H_{xx} & H_{xy} \\ H_{yx} & H_{yy} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\ &= \frac{\partial}{\partial y} [x^T H_{xx}x + 2y^T H_{yx}x + y^T H_{yy}y] \\ &= 2H_{yx}x + 2H_{yy}y = 0. \end{aligned}$$

Solving for the optimal response  $y$  then gives

$$y = g(x, u) = -H_{yy}^{-1} H_{yx}x = L(H)x = Lx, \quad (47)$$

where we view the parameter vector  $u$  as the vectorized version of the matrix  $L$ , i.e.  $u = \text{vec}(L)$  according to the definition of the  $\text{vec}$ -function in the appendix, section A.2. Note that the original choice of parameterizing  $Q(H)$  induces a parameterization of the policy  $g(H)$ . In difference with HDP no more parameters than the ones used for  $Q$  are needed.

Now, to the estimation of the parameters of the optimal Q-function. We start out with a gradient algorithm to solve the minimization problem in eq. 44 with respect to  $H$ , the matrix corresponding to the parameter vector  $w$ :

$$\begin{aligned} H_{i+1} &= H_i - \eta \frac{\partial}{\partial H'} E\{|z_i^T H' z_i - d(r, f, g(f), H_i)|^2\}_{H'=H_i} \\ &= H_i - \eta E \left\{ \frac{\partial}{\partial H'} |z_i^T H' z_i - d(r, f, g(f), H_i)|^2 \right\}_{H'=H_i} \\ &= H_i - \alpha E \{ z_i z_i^T (z_i^T H_i z_i - d(r, f, g(f), H_i)) \} \\ h_{i+1} &= h_i - \alpha E \{ v_i (v_i^T h_i - d(r, f, g(f), H_i)) \}. \end{aligned} \quad (48)$$

Here we denote  $\alpha = 2\eta$  and obtain the last equality by utilizing the fact that a quadratic form can be written as a scalar product, as described in section A.2 of the appendix, together with the notation  $z^T = (x^T \ y^T)$ ,  $v = \text{vec}(zz^T)$  and  $h = \text{vec}(H)$ .

The target for the ADHDP learning algorithm was given in eq. 43 and can be expressed for

the LQR problem using the parameterization in eqs. 45 and 46:

$$\begin{aligned} d(r, f, g(f), H_i) &= r(x, y) + Q(f(x, y), g(f(x, y)), H_i) \\ &= x^T Q x + y^T R y + (f^T \ g^T(f)) H_i \begin{pmatrix} f \\ g(f) \end{pmatrix} \end{aligned} \quad (49)$$

$$\begin{aligned} &= z^T \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} z + ((Ax + By)^T \ L^T (Ax + By)^T) H_i \begin{pmatrix} Ax + By \\ L(Ax + By) \end{pmatrix} \\ &= z^T \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} z + z^T \begin{pmatrix} A & B \\ LA & LB \end{pmatrix}^T H_i \begin{pmatrix} A & B \\ LA & LB \end{pmatrix} z. \end{aligned} \quad (50)$$

The reason for vectorizing the update rule for the gradient algorithm in eq. 48 is that it allows us to replace the expectation value with the instantaneous value and arrive a stochastic approximation algorithm. The update rule for this algorithm is found by insertion of the expression for the target, given by eq. 50, into the update rule in eq. 48:

$$\begin{aligned} h_{i+1} &= h_i - \alpha v_i (v_i^T h_i - d(r(z_i), z_{i+1}, Q(z_{i+1}, H_i))) \\ &= h_i - \alpha v_i \left( v_i^T h_i - \left[ z_i^T \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} z_i + z_i^T \begin{pmatrix} A & B \\ LA & LB \end{pmatrix}^T H_i \begin{pmatrix} A & B \\ LA & LB \end{pmatrix} z_i \right] \right) \\ &= h_i - \alpha v_i v_i^T \text{vec} \left( H_i - \left[ \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} + \begin{pmatrix} A & B \\ LA & LB \end{pmatrix}^T H_i \begin{pmatrix} A & B \\ LA & LB \end{pmatrix} \right] \right). \end{aligned} \quad (51)$$

Note that ADHDP only needs the current reward together with the Q-function, evaluated for the next state and response, to produce the target. Let us look at the vectorized update equation in the mean

$$E\{\text{vec}(H_{i+1})\} = E\{\text{vec}(H_i)\} - \alpha C_v E\{\text{vec}(\Delta H_i)\}, \quad (52)$$

where  $C_v = E\{vv^T\} > 0$  is a symmetric covariance matrix. From eq. B.1 we have that if a sequence  $w_{i+1} = w_i + \alpha \Delta w_i$  converges then so does the sequence  $w_{i+1} = w_i + \alpha C \Delta w_i$  if  $C > 0$  and is symmetric. Stating equation 52 in terms of matrices, with  $W = E\{H\}$  and  $C_v = I$ , results in the following equation for the update of the value function parameters in the mean:

$$W_{i+1} = W_i - \alpha \left[ W_i - \left( \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} + \begin{pmatrix} A & B \\ LA & LB \end{pmatrix}^T W_i \begin{pmatrix} A & B \\ LA & LB \end{pmatrix} \right) \right]. \quad (53)$$

That this sequence indeed converges to the parameters for the optimal Q-function  $Q_*(x) = z^T H^* z$  is proved by theorem B.2 in section B.3 in the appendix. Hence, we have proven that the update formula in eq. 51 converges, in the mean, to the matrix constituting the optimal value function for the linear quadratic system described by eqs. 45 and 46.

### 3.4 Action Dependent DHP

In action dependent DHP we model and estimate the gradient of the Q-function. In order to do this ADDHP needs samples from the derivatives of the instantaneous reward with respect to the state and the response together with a differentiable model of the environment. The gradients of the Q-function with respect to  $x$  and  $y$  are found by taking the gradient of eq. 11:

$$\frac{\partial Q}{\partial x} = \frac{\partial r}{\partial x} + \frac{\partial Q}{\partial f} \frac{\partial f}{\partial x} + \frac{\partial Q}{\partial g} \frac{\partial g}{\partial f} \frac{\partial f}{\partial x} \quad (54)$$

$$\frac{\partial Q}{\partial y} = \frac{\partial r}{\partial y} + \frac{\partial Q}{\partial f} \frac{\partial f}{\partial y} + \frac{\partial Q}{\partial g} \frac{\partial g}{\partial f} \frac{\partial f}{\partial y}. \quad (55)$$

Notice the absence of  $\partial y / \partial x$  since the Q-function is defined by the value of giving *any* response  $y$  and *then* following the policy  $y = g(x)$ , hence we have  $\partial y / \partial x = 0$  in this equation.

Targets for an algorithm that learns the parameters for the gradient of the Q-function,  $\partial Q(z, w) / \partial z$ ,  $z^T = (x^T \ y^T)$ , are found in the right hand side of eqs. 54 and 55. Since we know the parameterization of the policy we also know the derivative  $\partial g / \partial f$ . From the equations above we see that we need samples from  $\partial r / \partial x$  and  $\partial r / \partial y$  together with models for  $\partial f / \partial x$  and  $\partial f / \partial y$  to produce the targets:

$$dx(r_x, f, g, f_x, f_y) = r_x + Q_f f_x + Q_g g_f f_x \quad (56)$$

$$dy(r_y, f, g, f_x, f_y) = r_y + Q_f f_y + Q_g g_f f_y. \quad (57)$$

We use a gradient algorithm to search for the parameter vector that minimizes the expected squared error in the moving target problem

$$w' = \arg \min_{w'} E\{|\partial Q(z, w)/\partial z - d(r_z, f, g, f_z)|^2\}, \quad (58)$$

where we have the notion  $d = (dx \ dy)$ . Greedy iterations update our policy in order to find the gradient of the optimal Q-function. The policy is parameterized as  $g(x) = g(x, u)$  and the proper response is given by Bellman's optimality equation  $y^* = \arg \max_y Q(x, y)$ .

When applied to the LQR problem a suitable parameterization of the gradient of the Q-function, the gradient of the reward with respect to the state and the response,  $\partial r/\partial z$ , and the derivative of the environment with respect to the state and response,  $\partial f/\partial z$ , is given by:

$$\left(\frac{\partial Q}{\partial x} \ \frac{\partial Q}{\partial y}\right) = (x^T \ y^T) \begin{pmatrix} H_{xx} & H_{xy} \\ H_{yx} & H_{yy} \end{pmatrix} \quad r(x, y) = x^T Qx + y^T Ry \quad (59)$$

$$f(x, y) = Ax + By \quad y = g(x) = Lx. \quad (60)$$

In section 3.3 on ADHDP it was shown that the greedy response according to the current estimate of  $H$  is given by:

$$y = g(x, u) = -H_{yy}^{-1} H_{yx}x = L(H)x = Lx. \quad (61)$$

The targets for the LQR case are found by substituting equation eqs. 59 and 60 into eqs. 56 and 57. Note that eq. 59 give the gradients of  $Q$  with respect to the next state,  $f$ , and response,  $g(f)$ , as  $(f^T \ g^T(f))H$ . The targets can then be expressed according to:

$$\begin{aligned} dx &= 2x^T Q + 2(f^T H_{xx} + g^T(f)H_{yx})A + 2(f^T H_{xy} + g^T(f)H_{yy})LA \\ dy &= 2y^T R + 2(f^T H_{xx} + g^T(f)H_{yx})B + 2(f^T H_{xy} + g^T(f)H_{yy})LB. \end{aligned}$$

These expressions can be put together into a single formula for the target  $d = (dx \ dy)$ :

$$\begin{aligned} d &= 2(x^T \ y^T) \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} + 2(f^T \ g^T(f)) \begin{pmatrix} H_{xx} & H_{xy} \\ H_{yx} & H_{yy} \end{pmatrix} \begin{pmatrix} A & B \\ LA & LB \end{pmatrix} \\ &= 2z^T \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} + 2z^T \begin{pmatrix} A & B \\ LA & LB \end{pmatrix}^T H \begin{pmatrix} A & B \\ LA & LB \end{pmatrix}. \end{aligned} \quad (62)$$

Now, to the estimation of the parameters for the gradient of the optimal Q-function. We start out with a gradient algorithm to solve the minimization problem in eq. 58 with respect to  $Q$ , the matrix corresponding to the parameter vector  $w$ :

$$\begin{aligned} 2H_{i+1} &= 2H_i - \eta \frac{\partial}{\partial H'} E\{|2z_i^T H' - d(r_z, f, g(f), f_z, H_i)|^2\}_{H'=H} \\ &= 2H_i - \eta E \left\{ \frac{\partial}{\partial H'} |2z_i^T H' - d(r_z, f, g(f), f_z, H_i)|^2 \right\}_{H'=H_i} \\ &= 2H_i - \alpha E \{z_i(2z_i^T H_i - d(r_z, f, g(f), f_z, H_i))\}. \end{aligned} \quad (63)$$

As before we introduce  $\alpha = 2\eta$  in the last equation above. If we insert the expression for the target from eq. 62 into eq. 63 and replace the expectation value with the instantaneous value we get a stochastic approximation algorithm:

$$\begin{aligned} 2H_{i+1} &= 2H_i - \alpha z_i(2z_i^T H_i - d(r_z(i), z_{i+1}, f_z(i), Q_z(z_{i+1}, H_i))) \quad (64) \\ &= 2H_i - \alpha z_i \left( 2z_i^T H_i - \left[ 2z_i^T \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} + 2z_{i+1}^T H_i \begin{pmatrix} A & B \\ LA & LB \end{pmatrix} \right] \right) \\ H_{i+1} &= H_i - \alpha z_i z_i^T \left[ H_i - \left( \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} + \begin{pmatrix} A & B \\ LA & LB \end{pmatrix}^T H_i \begin{pmatrix} A & B \\ LA & LB \end{pmatrix} \right) \right]. \end{aligned}$$

Note that ADDHP needs samples from the derivatives of the reward and the derivative of the Q-function, evaluated for the next state and response, together with a differentiable model of the environment in order to calculate the target. In the mean the update equation becomes

$$W_{i+1} = W_i - \alpha C_z [W_i - \left( \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} + \begin{pmatrix} A & B \\ LA & LB \end{pmatrix}^T W_i \begin{pmatrix} A & B \\ LA & LB \end{pmatrix} \right)], \quad (65)$$

Method	Samples	Models
HDP	$r$	$V, \partial r / \partial y, \partial f / \partial y$
DHP	$\partial r / \partial x, \partial r / \partial y$	$\partial V / \partial x, \partial r / \partial x, \partial r / \partial y, \partial f / \partial x, \partial f / \partial y$
ADHDP	$r$	$Q$
ADDHP	$\partial r / \partial x, \partial r / \partial y$	$\partial Q / \partial x, \partial Q / \partial y, \partial f / \partial x, \partial f / \partial y$

Table 2: The information needed by adaptive critics to learn the optimal policy.

where  $W = E\{H\}$ , and  $C_z = E\{zz^T\}$ . In section B.1 it is shown that this sequence can be made convergent by an appropriate choice of  $\alpha$  if it converges with  $C_z = I$ . This equation is however identical to eq. 53 which is shown to be convergent by theorem B.2 in section B.3 of the appendix. Hence we have shown convergence, in the mean, for the stochastic approximation update rule in eq. 64.

## 4 Conclusions

We have presented convergence proofs for four stochastic approximation algorithms (HDP, DHP, ADHDP, and ADDHP) that converge to the true parameters of the optimal value function, or the derivative thereof. The proofs concern linear quadratic regulation in continuous state spaces. Previous theory has mainly been concerned with finite state domains and the results presented in this paper is one of the first steps towards a theoretical understanding of the convergence properties of adaptive critics. There exist many examples of successful applications of adaptive critics to far more complex problems than the LQR why every step towards bridging the gap between theory and practice should be regarded as momentous.

This paper stresses the importance of what functions to represent and parameterize. The different adaptive critics need different information in order to find the optimal policy as summarized in table 4. The way the LQR problem is approached by control theorists and people using adaptive critics is also different. These differences become most evident when the ADHDP, or Q-learning, algorithm is applied. Here only samples from the instantaneous reward function together with the parameters of the Q-function are needed in order to arrive at the optimal controller. This is to be compared with optimal control techniques where the instantaneous reward function is assumed to be known and the parameters of the environment need to be estimated and used in the solution of a Riccati equation before the optimal controller can be calculated.

The proofs of converge fall into two classes, one for HDP and DHP and another that relates to ADHDP and ADDHP. Both of these proofs bears considerable resemblances to proofs of convergence for Kalman filters in control theory [Lancaster and Rodman, 1995]. We think this paper points out that work made in the field of optimal control may contribute with fresh ideas as well as clues on how to establish, or at least investigate, the convergence properties of previously proposed algorithms in the field of reinforcement learning.

## A Quadratic forms as scalar products

In this appendix it is shown that quadratic forms involving vectors  $x \in \mathbb{R}^n$  may be viewed as a vector scalar product in the vector space  $\mathbb{R}^{n \times n}$ .

### A.1 The Kronecker product

The following matrix product has many names, it is referred to as the *right* Kronecker product, the *direct*, or the *tensor* product. It will be useful when dealing with quadratic forms in the derivation of some of the convergence proofs involving outer product of state vectors. The Kronecker product between  $A \in \mathbb{R}^{m \times m}$  and  $B \in \mathbb{R}^{n \times n}$  is defined as

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1m}B \\ a_{21}B & a_{22}B & & a_{2m}B \\ \vdots & & \ddots & \\ a_{m1}B & a_{m2}B & \dots & a_{mm}B \end{pmatrix}. \quad (66)$$

## A.2 The vectorize function

The *vec* function will be used to vectorize, or flatten, matrices into vectors by stacking the columns  $A_{*1}, \dots, A_{*n}$  of a matrix  $A \in \mathbb{R}^{m \times n}$  to form a vector  $\text{vec}(A) \in \mathbb{R}^{mn}$  with  $mn$  components

$$\text{vec}(A) = \begin{pmatrix} A_{*1} \\ \vdots \\ A_{*n} \end{pmatrix}. \quad (67)$$

Together with the Kronecker product, defined in section A.1, we are now able to write quadratic forms as scalar products between vectors. First it can be shown that

$$\text{vec}(X^T A Y) = (Y^T \otimes X^T) \text{vec}(A). \quad (68)$$

Now let us look at a quadratic form including the vector  $x \in \mathbb{R}^n$  and the matrix  $A \in \mathbb{R}^{n \times n}$ .

$$x^T A x = (x^T \otimes x^T) \text{vec}(A) = (x_1 x^T \quad \dots \quad x_n x^T) \text{vec}(A) = \text{vec}(x x^T)^T \text{vec}(A). \quad (69)$$

Hence the quadratic form can be calculated as the scalar product between the vectorized outer product  $x x^T$  and the vectorized matrix  $A$ .

## B Proofs of convergence for adaptive critics

The idea behind the proofs in section B.2 and B.3 stems from the book by Lancaster and Rodman who use the strategy to prove convergence for the Riccati difference equation in connection with the discrete Kalman filter [Lancaster and Rodman, 1995]. As before, a positive definite matrix  $P$  and a positive semidefinite matrix  $S$  are denoted by  $P > 0$  and  $S \geq 0$  respectively. First we look at the convergence of update rules where the update step is multiplied with a positive definite gain matrix.

### B.1 Positive definite gain matrices

When proving convergence in the mean one often ends up with a covariance matrix in front of the update step vector. Here we will show that if we know that recursion formula,

$$x_{k+1} = x_k + \alpha \Delta_k \quad (70)$$

converge, then it will still converge if the difference vector  $\Delta$  is multiplied with a symmetric positive definite gain matrix  $C$ . Assume that the step length  $\alpha \in (0, 1]$ . Since the matrix  $C$  is symmetric we know that it can be diagonalized with an orthogonal matrix  $C = E D E^T$ ,  $E^T E = I$ , where  $D_{ii} > 0$  since it is also positive definite. Now let us look at the modified recursion formula involving the gain matrix  $C$ :

$$x_{k+1} = x_k + \alpha C \Delta_k \quad (71)$$

$$= x_k + \alpha E D E^T \Delta_k. \quad (72)$$

Change coordinates by multiplying this equation from the left with the orthogonal matrix  $E^T$

$$E^T x_{k+1} = E^T x_k + \alpha E^T E D E^T \Delta_k = E^T x_k + \alpha D E^T \Delta_k = E^T x_k + \alpha E^T D \Delta_k. \quad (73)$$

The last step is due to the commutative nature of diagonal matrices. Now we can change back to the original coordinates again by multiplying from the left with the matrix  $E$

$$\begin{aligned} x_{k+1} &= x_k + \alpha D \Delta_k \\ x_{k+1,j} &= x_{k,j} + \alpha D_{jj} \Delta_{k,j} \\ &= x_{k,j} + \beta_j \Delta_{k,j}, \end{aligned}$$

where the two last equations gives the recursion for each row. We now see that we are guaranteed convergence if we chose the constant  $\alpha$  so that all the new constants  $\beta_j$  lie in the correct interval  $\beta_j = \alpha D_{jj} \in (0, 1]$ . Hence if we choose the constant  $\alpha$  according to

$$0 < \alpha \leq \frac{1}{\lambda_M}, \quad (74)$$

where  $\lambda_M = \max\{D_{jj}\}$  is the largest of  $C$ 's eigenvalues, then we are assured that if the sequence generated by eq. 70 converges, then so does the one generated by eq. 71.

## B.2 Proofs of convergence. HDP and DHP

**Theorem B.1 (Convergence of HDP and DHP)** *Assume that  $\alpha \in (0, 1]$ ,  $R > 0$ ,  $Q \geq 0$ , and  $(A, B)$  is a controllable pair. Define a sequence  $\{W_k\}_{k=0}^\infty$  according to eq. 75 below with  $W_0 = 0$ . Then there is a unique  $W^* \geq 0$  such that  $W_k \rightarrow W^*$  as  $k \rightarrow \infty$  in this recursion formula:*

$$\begin{aligned} W_{k+1} &= f(W_k, L_k) \\ &= W_k + \alpha [Q + L_k^T R L_k + (A + B L_k)^T W_k (A + B L_k) - W_k], \end{aligned} \quad (75)$$

where the feedback matrix  $L_k$  is given by

$$L_k = L(W_k) = -(R + B^T W_k B)^{-1} B^T W_k A. \quad (76)$$

The limit  $W^*$  is also the unique positive semidefinite solution to the discrete arithmetic Riccati equation (DARE)

$$W^* = Q + A^T (W^* - W^* B (R + B^T W^* B)^{-1} B^T W^*) A. \quad (77)$$

**Lemma B.1** *Let an arbitrary sequence  $\{H_k\}_{k=0}^\infty \subset \mathbb{R}^{n \times n}$  be given and also a matrix  $Z_0 \geq 0$ . Use this matrix as a starting point and generate a sequence  $\{Z_k\}_{k=0}^\infty$  using eq. 75, i.e.  $Z_{k+1} = f(Z_k, H_k)$ . Let  $W_0$  be a matrix for which  $0 \leq W_0 \leq Z_0$  and generate another sequence according to equations 75 and 76, i.e.  $W_{k+1} = f(W_k, L(W_k))$ . Then  $0 \leq W_k \leq Z_k$  for  $k = 0, 1, 2, \dots$*

**Proof.** The proof is by induction. The relation  $0 \leq W_0 \leq Z_0$  is given and we assume that  $0 \leq W_k \leq Z_k$ . First define  $\hat{Z}_{k+1} = f(Z_k, L(Z_k))$  and for brevity write  $\hat{R} = (R + B^T Z_k B)$  and  $L_k = L(W_k)$ . Now expand  $H_k = L_k + (H_k - L_k)$  and prove that  $Z_{k+1} \geq \hat{Z}_{k+1}$ :

$$\begin{aligned} Z_{k+1} &= f(Z_k, H_k) \\ &= Z_k + \alpha [Q + (L_k + (H_k - L_k))^T R (L_k + (H_k - L_k)) \\ &\quad + (A + B L_k + B (H_k - L_k))^T Z_k (A + B L_k + B (H_k - L_k)) - Z_k] \\ &= Z_k + \alpha [Q + L_k^T R L_k + (A + B L_k)^T Z_k (A + B L_k) - Z_k] \\ &\quad + \alpha [(H_k - L_k)^T (R L_k + B^T Z_k (A + B L_k)) \\ &\quad + (L_k^T R + (A + B L_k)^T Z_k B) (H_k - L_k) + (H_k - L_k)^T (R + B^T Z_k B) (H_k - L_k)] \\ &= f(Z_k, L_k) + \alpha [(H_k - L_k)^T (\hat{R} L_k + B^T Z_k A) + (L_k^T \hat{R} + A^T Z_k B) (H_k - L_k)] \\ &= f(Z_k, L_k) + \alpha (H_k - L_k)^T \hat{R} (H_k - L_k) \\ &\geq f(Z_k, L_k) = \hat{Z}_{k+1} \end{aligned} \quad (78)$$

The second last equality is due to eq. 76 which gives  $\hat{R} L_k = -B^T Z_k A$ . The last inequality is due to  $(H_k - L_k)^T \hat{R} (H_k - L_k) \geq 0$  since  $\hat{R} = R + B^T Z_k B \geq 0$  because  $\alpha \in (0, 1]$ ,  $R > 0$  and  $Z_k \geq 0$  from the induction hypothesis.

Thus, the choice of  $H_k = L_k = L(Z_k)$  results in a lower bound on  $f(Z_k, H_k)$ . We can now apply the same argument to the sequence  $\{W_k\}$  which shows that

$$f(W_k, L(Z_k)) \geq f(W_k, L(W_k)) = W_{k+1}. \quad (79)$$

The induction hypothesis  $W_k \leq Z_k$  now yields

$$\begin{aligned} f(W_k, L(Z_k)) &= (1 - \alpha) W_k + \alpha [Q + (L(Z_k))^T R L(Z_k) + (A + B L(Z_k))^T W_k (A + B L(Z_k))] \\ &\leq (1 - \alpha) Z_k + \alpha [Q + (L(Z_k))^T R L(Z_k) + (A + B L(Z_k))^T Z_k (A + B L(Z_k))] \\ &= f(Z_k, L(Z_k)) = \hat{Z}_{k+1}. \end{aligned}$$

This in combination with eq. 79 gives  $W_{k+1} \leq f(W_k, L(Z_k)) \leq f(Z_k, L(Z_k)) = \hat{Z}_{k+1}$ . From eq. 75 we have that  $W_k \geq 0$  implies  $W_{k+1} \geq 0$  which, together with eq. 78, results in  $0 \leq W_{k+1} \leq \hat{Z}_{k+1} \leq Z_{k+1}$  and the induction is complete.  $\square$

**Lemma B.2** *Let the sequence  $\{W_k\}$  be defined as in lemma B.1. If  $(A, B)$  is a controllable pair then there is a matrix  $Y$  such that  $0 \leq W_k \leq Y$  for  $k = 0, 1, 2, \dots$*

**Proof.** Since  $(A, B)$  is controllable there is a matrix  $L$  such that  $D = A + BL$  has all its eigenvalues in the open unit disc. Now generate a sequence  $\{Z_k\}$  from  $Z_0 = W_0$  and the recurrence relation  $Z_{k+1} = f(Z_k, L)$ . Then lemma B.1 can be applied which results in  $0 \leq W_k \leq Z_k$ ,  $k = 0, 1, 2, \dots$  by choosing  $H_k \equiv L$ . For  $j = 1, 2, \dots$  let us look at

$$\begin{aligned} Z_{k+1} - Z_k &= f(Z_k, L) - f(Z_{k-1}, L) \\ &= (1 - \alpha)(Z_k - Z_{k-1}) + \alpha(A + BL)^T(Z_k - Z_{k-1})(A + BL) \\ \text{vec}(Z_{k+1} - Z_k) &= [(1 - \alpha)I + \alpha D^T \otimes D^T] \text{vec}(Z_k - Z_{k-1}) \\ &= E \text{vec}(Z_k - Z_{k-1}). \end{aligned}$$

Thus,

$$\text{vec}(Z_k - Z_{k-1}) = E^{k-1} \text{vec}(Z_1 - Z_0) \quad (80)$$

and hence

$$\text{vec}(Z_n) = \text{vec}(Z_0) + \sum_{k=1}^n \text{vec}(Z_k - Z_{k-1}) = \text{vec}(Z_0) + \sum_{k=1}^n E^{k-1} \text{vec}(Z_1 - Z_0). \quad (81)$$

Since the eigenvalues,  $\lambda_i$  of  $D$  satisfy  $|\lambda_i| < 1$  and the eigenvalues of  $D^T \otimes D$  is given by  $\lambda_i^2$  we have together with  $\alpha \in (0, 1]$  that the eigenvalues of  $E$  equals  $|(1 - \alpha) + \alpha\lambda_i^2| < 1$ . Hence, there exists a norm in which  $e = \|E\| < 1$  and

$$\|\text{vec}(Z_n)\| = \|Z_n\| \leq \|Z_0\| + \|Z_1 - Z_0\| \sum_{k=0}^{\infty} e^k = e_0, \quad (82)$$

where  $e_0$  is independent of  $n$ . Now chose  $Y = e_0 I$  and obtain  $0 \leq W_k \leq Z_k \leq \|Z_k\| I \leq e_0 I = Y$ .  $\square$

**Proof of Theorem B.1** Generate two sequences  $\{W_k\}$  and  $\{Z_k\}$  starting from  $W_0 = Z_0 = 0$  by

$$W_{k+1} = f(W_k, L(W_k)) \quad , \quad Z_{k+1} = f(Z_k, L(W_{k+1})),$$

where  $L(\cdot)$  is given by eq. 76. Then lemma B.1 implies that  $0 \leq W_k \leq Z_k$  and with  $L_k = L(W_k)$  we obtain

$$W_{k+1} - Z_k = (1 - \alpha)(W_k - Z_{k-1}) + \alpha(A + BL_k)^T(W_k - Z_{k-1})(A + BL_k). \quad (83)$$

Now,  $W_0 = 0$  implies  $W_1 = Q \geq 0$  so that  $W_1 - Z_0 \geq 0$ . We then make the induction hypothesis  $W_k - Z_{k-1} \geq 0$  and eq. 83 yields  $W_{k+1} \geq Z_k$ . This leads us to  $0 \leq W_k \leq Z_k \leq W_{k+1}$  meaning that the sequence  $\{W_k\}$  is nondecreasing and from lemma B.2 we know that this sequence has an upper bound. It therefore has a limit  $W^*$  and taking limits in equation 75 we find that it is the solution of equation

$$W^* = Q + L(W^*)^T R L(W^*) + (A + BL(W^*))^T W^* (A + BL(W^*)). \quad (84)$$

Expanding this equation in terms of  $L(W^*)$  we obtain eq. 77. Since this equation has a unique positive semidefinite solution [Lancaster and Rodman, 1995] and we have shown  $W^* \geq 0$  this matrix must be the unique solution and the proof is completed.  $\square$

### B.3 Proofs of convergence. ADHDP and ADDHP

**Theorem B.2 (Convergence of ADDHP and ADDHP)** Assume that  $\alpha \in (0, 1]$ ,  $R > 0$ ,  $Q \geq 0$ , and  $(A, B)$  is a controllable pair. Define a sequence  $\{W_k\}_{k=0}^{\infty}$  according to eq. 85 below with  $W_0 = 0$ . Then there is a unique  $W^* \geq 0$  such that  $W_k \rightarrow W^*$  as  $k \rightarrow \infty$  in the recursion formula

$$\begin{aligned} W_{k+1} &= f(W_k, L_k) \\ &= W_k + \alpha \left[ \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} + \begin{pmatrix} A & B \\ L_k A & L_k B \end{pmatrix}^T \begin{pmatrix} W_{k,xx} & W_{k,xy} \\ W_{k,yx} & W_{k,yy} \end{pmatrix} \begin{pmatrix} A & B \\ L_k A & L_k B \end{pmatrix} - W_k \right] \end{aligned} \quad (85)$$

where the feedback matrix  $L_k$  is given by

$$L_k = L(W_k) = \begin{cases} 0 & , \quad k = 0 \\ -W_{k,yy}^{-1} W_{k,yx} & , \quad k > 0. \end{cases} \quad (86)$$

Furthermore the feedback matrix will in the limit equal the optimal one given by

$$L^* = -(R + B^T K B)^{-1} B^T K A,$$

where  $K$  is the unique positive semidefinite solution to the DARE presented in equation 77.

**Lemma B.3** Let an arbitrary sequence  $\{H_k\}_{k=0}^\infty \subset \mathbb{R}^{n \times n}$  be given and also a matrix  $Z_0 \geq 0$ . Use this matrix as a starting point and generate a sequence  $\{Z_k\}_{k=0}^\infty$  using eq. 85, i.e.  $Z_{k+1} = f(Z_k, H_k)$ . Let  $W_0$  be a matrix for which  $0 \leq W_0 \leq Z_0$  and generate another sequence according to equations 85 and 86, i.e.  $W_{k+1} = f(W_k, L(W_k))$ . Then  $0 \leq W_k \leq Z_k$  for  $k = 0, 1, 2, \dots$ .

**Proof.** The proof is by induction in two steps. The relation  $0 \leq W_0 \leq Z_0$  is given and we assume that  $0 \leq W_k \leq Z_k$ . First define  $\hat{Z}_{k+1} = f(Z_k, L(Z_k))$  and for brevity write

$$F = \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} \quad \text{and} \quad L_k = L(Z_k).$$

Now rewrite  $H_k = L_k + (H_k - L_k)$  and prove that  $Z_{k+1} \geq \hat{Z}_{k+1}$ :

$$\begin{aligned} Z_{k+1} &= f(Z_k, H_k) \\ &= Z_k + \alpha \left[ F + \begin{pmatrix} A^T \\ B^T \end{pmatrix} \begin{pmatrix} I & H_k^T \\ Z_{k,xx} & Z_{k,xy} \\ Z_{k,yx} & Z_{k,yy} \end{pmatrix} \begin{pmatrix} I \\ H_k \end{pmatrix} (A \ B) - Z_k \right] \\ &= Z_k + \alpha \left[ F + \begin{pmatrix} A^T \\ B^T \end{pmatrix} (Z_{k,xx} + Z_{k,xy}(L_k + (H_k - L_k)) + (L_k + (H_k - L_k))^T Z_{k,yx} \right. \\ &\quad \left. + (L_k + (H_k - L_k))^T Z_{k,yy}(L_k + (H_k - L_k))) (A \ B) - Z_k \right] \\ &= Z_k + \alpha \left[ F + \begin{pmatrix} A^T \\ B^T \end{pmatrix} (Z_{k,xx} + Z_{k,xy}L_k + L_k^T Z_{k,yx} + L_k^T Z_{k,yy}L_k) (A \ B) - Z_k \right] \\ &\quad + \alpha \begin{pmatrix} A^T \\ B^T \end{pmatrix} ((H_k - L_k)^T (Z_{k,yx} + Z_{k,yy}L_k) + (Z_{k,xy} + L_k^T Z_{k,yy})(H_k - L_k))^T \\ &\quad + (H_k - L_k)^T Z_{k,yy}(H_k - L_k) (A \ B) \\ &= f(Z_k, L_k) + \alpha \begin{pmatrix} A^T \\ B^T \end{pmatrix} (H_k - L_k)^T Z_{k,yy}(H_k - L_k) (A \ B) \geq f(Z_k, L_k) = \hat{Z}_{k+1} \end{aligned} \quad (87)$$

The second last equality is due to  $Z_{k,yy}L_k = -Z_{k,yx}$  and the inequality comes from the fact that  $Z_{k,yy} \geq 0$ . This is shown by noting that  $Z_0 \geq 0$  and using the induction hypothesis  $Z_{k,yy} \geq 0$  together with the lower right block of equation 85 which gives

$$Z_{k+1,yy} = (1 - \alpha)Z_{k,yy} + \alpha [R + B^T L_k^T Z_{k,yy} L_k B] > 0,$$

since  $\alpha \in (0, 1]$  and  $R > 0$ .

Thus, the choice of  $H_k = L_k = L(Z_k)$  results in a lower bound on  $f(Z_k, H_k)$ . We can now apply the same argument to the sequence  $\{W_k\}$  which shows that

$$f(W_k, L(Z_k)) \geq f(W_k, L(W_k)) = W_{k+1}. \quad (88)$$

The induction hypothesis  $W_k \leq Z_k$  now yields

$$\begin{aligned} f(W_k, L(Z_k)) &= (1 - \alpha) W_k + \alpha \left[ F + \begin{pmatrix} A^T \\ B^T \end{pmatrix} (I \ L^T(Z_k)) W_k \begin{pmatrix} I \\ L(Z_k) \end{pmatrix} (A \ B) \right] \\ &\leq (1 - \alpha) Z_k + \alpha \left[ F + \begin{pmatrix} A^T \\ B^T \end{pmatrix} (I \ L^T(Z_k)) Z_k \begin{pmatrix} I \\ L(Z_k) \end{pmatrix} (A \ B) \right] \\ &= f(Z_k, L(Z_k)) = \hat{Z}_k. \end{aligned}$$

This in combination with eq. 88 gives  $W_{k+1} \leq f(W_k, L(Z_k)) \leq f(Z_k, L(Z_k)) = \hat{Z}_{k+1}$ . From eq. 85 we have that  $W_k \geq 0$  implies  $W_{k+1} \geq 0$  which, together with eq. 87, results in  $0 \leq W_{k+1} \leq \hat{Z}_{k+1} \leq Z_{k+1}$  and the induction is complete.  $\square$

**Lemma B.4** Let the sequence  $\{W_k\}$  be defined as in lemma B.3. If  $(A, B)$  is a controllable pair then there is a matrix  $Y$  such that  $0 \leq W_k \leq Y$  for  $k = 0, 1, 2, \dots$ .

**Proof.** Generate a sequence  $\{Z_k\}$  from  $Z_0 = W_0$  and the recurrence relation  $Z_{k+1} = f(Z_k, L)$ . Here the matrix  $L$  is chosen such that  $\|A + BL\| < 1$  which is possible since the pair  $(A, B)$  is controllable. Then lemma B.3 can be applied which results in  $0 \leq W_k \leq Z_k$ ,  $k = 0, 1, 2, \dots$  by choosing  $H_k \equiv L$ . For  $j = 1, 2, \dots$  let us look at

$$\begin{aligned} Z_{k+1} - Z_k &= f(Z_k, L) - f(Z_{k-1}, L) \\ &= (1 - \alpha)(Z_k - Z_{k-1}) + \alpha \begin{pmatrix} A^T \\ B^T \end{pmatrix} (I \ L^T)(Z_k - Z_{k-1}) \begin{pmatrix} I \\ L \end{pmatrix} \\ \text{vec}(Z_{k+1} - Z_k) &= \left[ (1 - \alpha)I + \alpha \begin{pmatrix} A & B \\ LA & LB \end{pmatrix}^T \otimes \begin{pmatrix} A & B \\ LA & LB \end{pmatrix}^T \right] \text{vec}(Z_k - Z_{k-1}) \\ &= E \text{vec}(Z_k - Z_{k-1}). \end{aligned}$$

Thus,

$$\text{vec}(Z_k - Z_{k-1}) = E^{k-1} \text{vec}(Z_1 - Z_0) \quad (89)$$

and hence

$$\text{vec}(Z_n) = \text{vec}(Z_0) + \sum_{k=1}^n \text{vec}(Z_k - Z_{k-1}) = \text{vec}(Z_0) + \sum_{k=1}^n E^{k-1} \text{vec}(Z_1 - Z_0). \quad (90)$$

Now let us show that  $\|E\| < 1$ . This is done by first looking at the norm of  $D = (I \ L^T)^T (A \ B)$  by studying  $D^k$

$$\begin{aligned} D^k &= \begin{pmatrix} I \\ L \end{pmatrix} (A \ B) \underbrace{\begin{pmatrix} I \\ L \end{pmatrix} \cdots \begin{pmatrix} I \\ L \end{pmatrix}}_{k-1 \text{ pairs}} (A \ B) = \begin{pmatrix} I \\ L \end{pmatrix} (A + BL)^{k-1} (A \ B) \\ \|D^k\| &\leq \left\| \begin{pmatrix} I \\ L \end{pmatrix} \right\| \|(A + BL)^{k-1}\| \|(A \ B)\| \leq \gamma \|(A + BL)\|^{k-1}. \end{aligned}$$

Taking the limit  $k \rightarrow \infty$  of the last row in the equation above results in

$$\lim_{k \rightarrow \infty} \|D^k\| \leq \lim_{k \rightarrow \infty} \|D\|^k \leq \lim_{k \rightarrow \infty} \gamma \|(A + BL)\|^{k-1} = 0, \quad (91)$$

since  $\|A + BL\| < 1$  for our choice of  $L$  due to  $(A, B)$  being controllable. This then means that  $\|D\| \in (0, 1)$  giving us,  $\lambda_i$  of  $D$  satisfying  $|\lambda_i| < 1$ . The eigenvalues of  $D^T \otimes D^T$  is then given by  $\lambda_i^2$  and together with  $\alpha \in (0, 1]$  we have that the eigenvalues of  $E$  equals  $|(1 - \alpha) + \alpha \lambda_i^2| < 1$ . Hence, there exists a norm in which  $e = \|E\| < 1$  and from eq. 90 we get

$$\|\text{vec}(Z_n)\| = \|Z_n\| \leq \|Z_0\| + \|Z_1 - Z_0\| \sum_{k=0}^{\infty} e^k = e_0, \quad (92)$$

where  $e_0$  is independent of  $n$ . Now chose  $Y = e_0 I$  and obtain  $0 \leq W_k \leq Z_k \leq \|Z_k\| I \leq e_0 I = Y$ .  $\square$

**Proof of Theorem B.2** Generate two sequences  $\{W_k\}$  and  $\{Z_k\}$  starting from  $W_0 = Z_0 = 0$  by

$$W_{k+1} = f(W_k, L(W_k)) \quad , \quad Z_{k+1} = f(Z_k, L(W_{k+1})),$$

where  $L(\cdot)$  is given by eq. 86. Then lemma B.3 implies that  $0 \leq W_k \leq Z_k$  and with  $L_k = L(W_k)$  we obtain

$$W_{k+1} - Z_k = (1 - \alpha)(W_k - Z_{k-1}) + \alpha \begin{pmatrix} A^T \\ B^T \end{pmatrix} (I \ L_k^T)(W_k - Z_{k-1}) \begin{pmatrix} I \\ L_k \end{pmatrix} (A \ B). \quad (93)$$

Now,  $W_0 = 0$  implies  $W_1 = F \geq 0$  so that  $W_1 - Z_0 \geq 0$ . We now make the induction hypothesis  $W_k - Z_{k-1} \geq 0$  and eq. 83 yields  $W_{k+1} \geq Z_k$ . This leads us to  $0 \leq W_k \leq Z_k \leq W_{k+1}$  meaning that the sequence  $\{W_k\}$  is nondecreasing and from lemma B.4 we know that this sequence has

an upper bound. It therefore has a limit  $W^*$  and taking limits in equation 85 we find that it solves the equation

$$W^* = \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} + \begin{pmatrix} A^T \\ B^T \end{pmatrix} (I L^T(W^*)) W^* \begin{pmatrix} I \\ L(W^*) \end{pmatrix} (A \ B). \quad (94)$$

Expanding this equation in terms of  $L(W^*)$  we obtain, with the notation  $W = W^*$

$$\begin{pmatrix} W_{xx} & W_{xy} \\ W_{yx} & W_{yy} \end{pmatrix} = \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} + \begin{pmatrix} A^T \\ B^T \end{pmatrix} (W_{xx} - W_{xy} W_{yy}^{-1} W_{yx}) (A \ B) \quad (95)$$

$$= \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} + \begin{pmatrix} A^T \\ B^T \end{pmatrix} K (A \ B). \quad (96)$$

From this definition of the matrix  $K$  we see that  $K = (I L^T(W)) W (I L^T(W))^T \geq 0$  since the limit  $W \geq 0$ . Hence, the matrix  $K$  is a positive semidefinite solution of the equation

$$K = W_{xx} - W_{xy} W_{yy}^{-1} W_{yx} = Q + A^T K A - A^T K B (R + B^T K B)^{-1} B^T K A,$$

which is recognized as eq. 77. Since this equation has a unique positive semidefinite solution and we have shown  $K \geq 0$  this matrix must be the unique solution. Finally, let us look at the resulting feedback matrix by expressing it in terms of the matrix  $K$  using eq. 96

$$L(W) = -W_{yy}^{-1} W_{yx} = -(R + B^T K B) A^T K B.$$

This means that the limit  $W = W^*$  will indeed result in the optimal feedback matrix  $L^* = L(W^*)$  and the proof is concluded.  $\square$

## References

- [Åström and Wittenmark, 1989] Åström, K. J. and Wittenmark, B. (1989). *Adaptive Control*. Addison-Wesley publishing comp.
- [Bradtke, 1993] Bradtke, S. J. (1993). Reinforcement learning applied to linear quadratic regulation. In *Advances in Neural Information Processing Systems 5*, San Mateo, CA. Morgan Kaufmann.
- [Denardo, 1967] Denardo, E. V. (1967). Contraction mappings in the theory underlying dynamic programming. *SIAM Review*, 9(2):165–177.
- [Howard, 1960] Howard, R. A. (1960). *Dynamic programming and Markov processes*. John Wiley & Sons Inc.
- [Lancaster and Rodman, 1995] Lancaster, P. and Rodman, L. (1995). *Algebraic Riccati Equations*. Oxford University Press.
- [Morari, 1993] Morari, M. (1993). *Model Predictive Control*. Prentice Hall.
- [Sutton, 1988] Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44.
- [Sutton et al., 1991] Sutton, R. S., Barto, A. G., and Williams, R. J. (1991). Reinforcement learning is direct adaptive optimal control. In *Proc. of the American Control Conf.*, pages 2143–2146.
- [Watkins, 1989] Watkins, C. (1989). *Learning from Delayed Rewards*. PhD thesis, Cambridge University.
- [Werbos, 1992] Werbos, P. (1992). *Handbook of Intelligent Control*, chapter Approximate dynamic programming for real-time control and neural modelling. Van Nostrand Reinhold. D. A. White and D. A. Sofge, Eds.
- [Werbos, 1990] Werbos, P. J. (1990). Consistency of HDP applied to a simple reinforcement learning problem. *Neural Networks*, 3:179–189.
- [White and Sofge, 1992] White, D. A. and Sofge, D. A. (1992). *Handbook of intelligent control: fuzzy, neural, and adaptive approaches*. New York Van Nostrand Reinhold cop.