Introduction

The field of Artificial Intelligence has three main machine learning methodologies, one of which, and the focus of this paper, being Reinforcement Learning. Reinforcement Learning is concerned with how agents, with their own level of intelligence and policies, take actions in a set environment to maximize the level of reward they can get. The level of reward that the agent receives is based upon the action and its repercussions in the environment and the environments response. Typically, the environment, in engineering applications, is the Euclidean space, and the algorithms in this process are typically dynamic programming techniques. These mathematical techniques were first originated with the work of Richard Bellman, and then was further developed by Bellman, his coworkers, and many other researchers in the years to come. Bellman originally developed these in the 1950s and 1960s, with much more research done with dynamic programming since. It is used in a multitude of scientific and engineering applications, mostly to solve either continuous or discrete-time dynamic optimization problems. These include minimization or maximization of a performance criterion along set trajectories in a dynamic system.

In Dynamic Programming, the steps are performed iteratively, in a loop. This iteration allows for the method of successive approximations in policy space, and this is then utilized for updating the feedback on each time interval. This resulting control is used to approximate the optimal control, within a level of approximation.

Problem Formulation

One of the most important components in solving this problem is determining an optimal feedback control unit. The Bellman's Optimatlity Principle states: "An optimal policy has a property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regards to the state resulting from the first decision."

The goal is to find the optimal feedback control unit $u^{\text{opt}}(x(t))$. This minimizes the performance criterion. The main issue with regard to this is that it is incredibly challenging, if not impossible, to find this in a general case. The optimal feedback controller is only able to be found in rare cases, one of which is the optimal linear-quadratic case, which utilizes a linear dynamic system and quadratic performance criterion. The linear-quadratic optimal case is what will be explored in this paper, as well as how to solve it.

The main methodology to solve the dynamic optimization problem is via successive iterations, also known as policy iterations. This has been utilized in many different approaches and papers. In essence, how successive iterations work is by garnering feedback from the system, updating the policy, and repeating these same steps until the desired result is found. The goodness of the approximation depends on the updating frequency, rate of change of linear

model parameters, and effectiveness of using steady state control in the final phase of the process. The method relies on two forms of feedback: the updating of the parameters of the model at each time interval, and the dependence of the control on the state of the system over each time interval. [3]

Main Results

State Space form and Cost

In this paper, a general time-invariant continuous-time linear dynamic system is represented in its state space form by

$$f(x,u,t) = \frac{dx(t)}{dt} = Ax(t) + Bu(t) \qquad x(t_0) = x_0 \tag{1}$$

In this equation, $x(t) \in R^n$, and is an n-dimensional state vector, $u(t) \in R^n$, and is an m -dimensional control vector. A and B are $(n \times m)$ constant matrices. The scalar performance criterion, also known as the cost, associated to this dynamic system is given by:

$$J(x(t_0)) = \frac{1}{2}\int_{t_0}^{t_f} (x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau))d\tau + \frac{1}{2}x^T(t_f)P_f x(t_f) \tag{2}$$

$$Q = Q^T \geq 0, P_f = P_f^T \geq 0, R = R^T > 0$$

where Q, P_f, R are all weight matrices used to define relative importance of state variables and inputs in defining the total cost of J. The main goal is to minimize the cost, J, using feedback control $u(x(t))$. In this version of the equation, the term $\frac{1}{2}$ is introduced as a scaling factor for convenience. The equation can further be broken up into two main components. The active component, which is the part being integrated, and the terminal constraint, which is the second half of the equation. In this paper, a focus will be taken on the active portion of the equation and will do optimization for $t_f \to \infty$. This equation has a Hamiltonian version, which can be simplified and written out as

$$H\left(x(t), u(t), \frac{\partial J^{opt}(x(t))}{\partial x}\right) = \frac{1}{2}(x^T(t)Qx(t) + u^T(t)Ru(t)) + \left(\frac{\partial J^{opt}(x(t))}{\partial x}\right)^T (Ax(t) + Bu(t)) \tag{3}$$

Necessary Conditions

There are three necessary conditions for the minimum of the cost function:

$$\frac{dx(t)}{dt} = \frac{\partial}{\partial p}\{H(x(t), u(t), p(t))\} = Ax(t) + Bu(t) \qquad x(t_0) = x_0$$

$$\frac{dp(t)}{dt} = -\frac{\partial}{\partial x}\{H(x(t), u(t), p(t))\} = -Qx(t) - A^T p(t), \qquad p(t_f) = \frac{\partial}{\partial x}\left\{\frac{1}{2}x^T(t_f)P_f x(t_f)\right\} = P_f x(t_f)$$

$$0 = \min(u)\{H(x(t), u(t)p(t))\} = \frac{\partial}{\partial u}\{H(x(t), u(t), p(t))\} \rightarrow Ru(t) + B^T p(t) = 0 \rightarrow u^{\text{opt}}(p(t)) = -R^{-1}B^T p(t)$$

Since $\frac{\partial^2 H}{\partial u^2} = R > 0$, the sufficient condition for minimum is satisfied. After substituting in the value of optimal control into the original state equation, the resulting state-costate equations form is:

$$\frac{dx(t)}{dt} = Ax(t) - BR^{-1}B^T p(t), \qquad x(t_0) = x_0 \tag{4}$$

$$\frac{dp(t)}{dt} = -Qx(t) - A^T p(t) \qquad p(\infty) = 0 \tag{5}$$

Now the steps to solve the dynamic optimization via successive approximation will be delved into.

Step 0: Precheck

Something that needs to be done before the equation can start to be solved is to check if the optimal linear quadratic controller exists. In essence, check that *(A,B)* is controllable and that *(A, Chol(Q))* is observable. Chol(Q) stands for the Cholesky decomposition of the weighted matrix Q.

Step 1: Initialization of Controls

Initialization is required, and it needs a feedback control input *u(x(t))*. This is shown as the following:

$$u^{opt}(x(t)) = -R^{-1}B^T P x^{opt}(t) = -F^{opt}x^{opt}(t) \tag{6}$$

where P is the unique positive definite solution to the Riccati algebraic equation, which will be delved into later in this paper. The optimal state trajectory is

$$\frac{dx^{opt}(t)}{dt} = \left(A - BF^{opt}(t)\right)x^{opt}(t), \qquad x^{opt}(t_0) = x_0 \tag{7}$$

with F^{opt} is the optimal feedback gain.

Step 2: Riccati Equation

This next section delves into utilizing and solving the Riccati equations. In this paper, $t_f \to \infty$. Due to t_f going to infinity, and subsequently $P_f = 0$, the procedure occurring under controllability and observability conditions, the differential Riccati is able to become the matrix algebraic Riccati equation, which is given by

$$0 = A^T P + PA + Q - PBR^{-1}B^T P \tag{8}$$

Note that this equation can be simplified where

$$A^T P + PA + Q - PSP = 0; \qquad S = BR^{-1}B^T$$

The required solution of this exists under a standard control oriented assumption, where the unique positive semi-definite stabilizing solution is obtained in terms of the Lyapunov equations. The Kleinman Algorithm is given by [2,4]:

$$(A - SP^{(i)})^T P^{(i+1)} + P^{(i+1)}(A - SP^{(i)}) + Q + P^{(i)}SP^{(i)} = 0$$
$$\text{with} \qquad (A - SP^{(0)}) \qquad \text{asymptotically stable}$$

This is proven to converge in [4].

There is a unique solution for this under the assumption that the coefficient matrix is asymptotically stable [4]. This will be delved further into in the next section of this paper. Thus, the approximate performance criterion can be evaluated using the following sequence of properties

$$J_1(x(0)) = \frac{1}{2}x^T(0)P_1x(0) \tag{9}$$

This updates the control in the system to:

$$u_2(x(t)) = -R^{-1}B^T P_1 x(t) = -F_2 x(t)$$

It is then possible to approximate the state trajectories and to approximate the system performance. As seen from (7) [2]:

$$\frac{dx_2(t)}{dt} = (A - BF_2)x(t), \qquad x_2(0) = x(0) = x_0 \rightarrow x_2(t) = e^{(A-BF_2)t}x(0)$$

In this, we can keep furthering this iteration

$$J_2(x(0)) = \frac{1}{2}X^T(0)P_2x(0), \qquad 0 = (A - BF_2)^TP_2 + P_2(A - BF_2) + Q + F_2^TRF_2$$

This is further proven in [2]

Step k

The most interesting aspect of this is when comparing iteration J_1 and J_2. It was proved that [1,3]

$$J_1(x(0)) \geq J_2(x(0)) \tag{10}$$

This is the essence of policy iteration and successive approximations. Therefore, $x^TP_tx \geq x^TP_{t+1}x$ and thus the convergence will approximate to a consistent, optimized, cost. This last step, as shown in (10) and the previous step, can be iterated k-times until the convergence to the optimal performance, optimal system trajectory, and optimal control are all solved for. The following was proved [1,3]:

$$J_1(x(0)) \geq J_2(x(0)) \geq J_2x(0)) \geq ... \geq J_k(x(0)) \geq ... \geq J_{opt}(x(0)) = \frac{1}{2}x^T(0)P_{opt}x(0) \tag{11}$$

This can also be seen with the Hamiltonian, as dictated in (3) satisfy:

$$H_{min}^{(k+1)} \geq H^{(k+1)} \tag{12}$$

Since the approximate performance criteria are given in quadratic forms, if can also be observed that:

$$P_1 \geq P_2 \geq P_3 \geq ... \geq P_k \geq ... \geq P_{opt} \tag{13}$$

P_k matrices are obtained iteratively, from the algebraic Lyapunoc equations which converge to the solution of the algebraic Riccati equation.

For the quadratic cost functional the minimum is unique and therefore the sequence is strictly decreasing. Further-

more, it is bounded from below by 0. [1]

Kleinman's Algorithm

In this section, Kleinman's Algorithm for solving the Algebraic Riccati Equation is delved into. It is quite interesting to observe that the Kleinman's algorithm could have been derived using the Newtons method for solving the algebraic Riccati equation. The equation that is being derived into is

$$(A - SP^{(i)})^T P^{(i+1)} + P^{(i+1)}(A - SP^{(i)}) + Q + P^i SP^{(i)} = 0; \qquad k = 0, 1, 2, 3, ... \tag{14}$$

where

$$S = BR^{-1}B^T$$

This is incredibly useful for learning about the speed that the system converges, this is because the Newton method has quadratic convergence. However, to properly converge, the Newton algorithm should have a good initial guess. In this section, the Newton algorithm is derived. This will represent a linearization method, where the unknown variable P will be replaced by P^{i+1}, the terms will all be multiplied, and finally, the square and high order terms are neglected with respect to Δ.

This derivation will be done on an example of the algebraic Riccati equation:

$$A^T P + PA + Q - PSP = 0, \qquad S = BR^{-1}B^T \tag{15}$$

In this, the P terms can be replaced by $P^{(i+1)}$, which is equivalent to $P^{(i)} + \Delta$. The resulting equation is:

$$A^T(P^{(i)} + \Delta) + (P^{(i)} + \Delta)A + Q - (P^{(i)} + \Delta)S(P^{(i)} + \Delta) = 0 \tag{16}$$

From this, the terms in the quadratic term can be multiplied. This leads to:

$$A^T P^{i+1} + P^{i+1}A + Q - P^{(i)}SP^{(i)} - P^{(i)}S\Delta - \Delta SP^{(i)} - \Delta S\Delta = 0 \tag{17}$$

The quadratic term with respect to Δ can be neglected, and the resulting form is:

$$A^T P^{i+1} + P^{i+1}A + Q - P^{(i)}SP^{(i)} - P^{(i)}S\Delta - \Delta SP^{(i)} \approx 0 \tag{18}$$

After this, $\pm P^{(i)}SP^{(i)}$ is added/subtracted to this equation:

$$A^TP^{(i+1)} + P^{(i+1)}A + Q - P^{(i)}SP^{(i)} - P^{(i)}S\Delta - \Delta SP^{(i)} + P^{(i)}SP^{(i)} - P^{(i)}SP^{(i)} \approx 0 \tag{19}$$

Some grouping can be done, specifically of the 4th and 5th terms, and the 6th and 7th terms.

$$A^TP^{(i+1)} + P^{(i+1)}A + Q - P^{(i)}S(P^{(i)} + \Delta) - (\Delta + P^{(i)})SP^{(i)} + P^{(i)}SP^{(i)} \approx 0 \tag{20}$$

$$A^TP^{(i+1)} + P^{(i+1)}A + Q - P^{(i)}SP^{(i+1)} - P^{(i+1)}SP^{(i)} + P^{(i)}SP^{(i)} \approx 0 \tag{21}$$

Finally, the $P^{(i+1)}$ can be factored out, which leads to this final form:

$$\left(A - SP^{(i)}\right)^T P^{(i+1)} + P^{(i+1)}\left(A - SP^{(i)}\right) + Q + P^{(i)}SP^{(i)} \approx 0 \tag{22}$$

This represents Kleinman's Algorithm. A stabilizing initial condition still must be guessed to ensure convergence to a stabilized solution P.

$$\left(A - SP^{(i)}\right)^T P^{(i+1)} + P^{(i+1)}\left(A - SP^{(i)}\right) + Q + P^{(i)}SP^{(i)} = 0 \tag{23}$$

$$P_0 \text{ exists such at } \left(A - SP^0\right) < 0, i = 1, 2, 3, ...$$

Note: Due to quadratic rate of convergence, this algorithm tends to converge in 4-5 iterations, rarely more. However, if Δ is not sufficiently small, the Newton algorithm acts as a random number generator

This can be proved by utilizing Kleinman's Algorithm for Riccati Equation. This algorithm shows the following:

$$\left(A - SP^{(i)}\right)^T P^{(i+1)} + P^{(i+1)}\left(A - SP^{(i)}\right) + Q + P^{(i)}SP^{(i)} = 0 \tag{24}$$

$$\text{with } \left(A - SP^{(0)}\right) \text{ asymptotically stable}$$

It is stated that "under the assumption of the triple (A, B, $\sqrt{Q}$) being stabilizable-detectable, any stabilizing initial guess $P^{(0)}$ makes (24) convergent to the desired positive semi-definite stabilizing solution of" (15). [4] In short, what that means is that if the previous instance is stable, then the next one will be as well. This is proved using convergence:

$$\left(A - SP^{(i)}\right)^T \left(P^{(i+1)} - P\right) + \left(P^{(i+1)} - P\right) \left(A - SP^{(i)}\right) = -\left(P^{(i)} - P\right) S \left(P^{(i)} - P\right) \tag{25}$$

As the right-hand side of this equation is negative semi-definite, it shows that if there is a unique solution, then the solution must be positive semi-definite for every i. There is a unique solution when $A - SP^{(i)} < 0$, for all i. This shows that

$$P^{(i+1)} - P \geq 0, \qquad i = 0, 1, 2, ... \tag{26}$$

From our original (24), we have

$$\left(A - SP^{(i+1)}\right)^T \left(P^{(i+2)} - P^{(i+1)}\right) + \left(P^{(i+2)} - P^{(i+1)}\right) \left(A - SP^{(i+1)}\right) = -\left(P^{(i)} - P^{(i+1)}\right) S \left(P^{(i)} - P^{(i+1)}\right) \tag{27}$$

$$i = 0, 1, 2, ...$$

Assuming asymptotic stability of matrices $A - SP^{(i+1)}, i = 0, 1, 2, ...$ implies

$$P^{(i+1)} \geq P^{(i+2)}, \qquad i = 0, 1, 2, ... \tag{28}$$

Thus, the solution matrices form a monotonically non-increasing sequence, bounded from below by required solution.

$$P^{(1)} \geq P^{(2)} \geq P^{(3)} \geq ... \geq P \tag{29}$$

This is in line with (13), which also shows convergence.

The closed loop system-matrices $A - SP^{(i+1)}$ are always asymptotically stable, providing the inital guess $P^{(0)}$ is stabilizing. The proof of this is in [4], and is shown following.

This is done by contradiction, assuming that $A - SP^{(i)}$ is stable, but the matrix $A - SP^{(i+1)}$ is unstable. Thus,

$$\left(A - SP^{(i+1)}\right) x = \lambda x, \qquad x \neq 0, Re\{\lambda\} \geq 0 \tag{30}$$

which, using (25), derives

$$\begin{aligned} \left(A - SP^{(i+1)}\right)^T \left(P^{(i+1)} - P\right) + \left(P^{(i+1)} - P\right) \left(A - SP^{(i+1)}\right) = \\ - \left(P^{(i+1)} - P^{(i)}\right) S \left(P^{(i+1)} - P^{(i)}\right) - \left(P^{(i+1)} - P\right) S \left(P^{(i+1)} - P\right) \end{aligned} \tag{31}$$

Then, utilizing (30),

$$\begin{aligned} x^T 2Re\{\lambda\} \left(P^{(i+1)} - P^{(i)}\right) x \\ &= -x^T \left(P^{(i+1)} - P^{(i)}\right) S \left(P^{(i+1)} - P^{(i)}\right) x - x^T \left(P^{(i+1)} - P\right) S \left(P^{(i+1)} - P\right) \\ &= -x^T M x \end{aligned} \tag{32}$$

Since the left hand side of the last inequality is positive semi-definite by (28), and the right hand is negative semi-definite, $x^T M x = 0$ must be satisfied. This then implies

$$S \left(P^{(i+1)} - P^{(i)}\right) x = 0 \tag{33}$$

which then shows that

$$\left(A - SP^{(i)}\right) x = \left(A - SP^{(i+1)}\right) x = \lambda x \qquad Re\{\lambda\} \geq 0 \tag{34}$$

contradicting the assumption.

This proves that the Kleinman algorithm converges to a unique solution of the algebraic Riccati equation, and that this solution is unique, positive, stabilizing, and semi-definite. [4]

Conclusions

In this paper, the optimal linear-quadratic case of dynamic programming was explored and solved, with the optimal feedback control unit $u^{opt}(x(t))$ being solved for and found. This was solved through successive approximations in policy space, which appears to be a Reinforcement Learning algorithm. Successive approximations in policy space is a method where a system updates its policies based upon feedback from the environment and system response. This method set up a State Space and a scalar performance criterion, and then utilized the Hamiltonian to simplify it. It then presented three necessary conditions and initialized the controls. One of the main components of the paper, and of this problem, is utilizing the Riccati Equation, and this paper delved into the Algebraic and Matrix Riccati Equations to help solve. Finally, the iteration of gathering solutions and bringing feedback into the system was delved into, and the convergence was proven.

The paper also emphasized and showed proof of Kleinman's Algorithm, by showing the derivation from the original Algebraic Riccati Equation into Kleinman's Algorithm, and from there proving its statement. It shows that if the initial $P^{(0)}$ is stable, then the rest of the iterations will be stable as well.

References

[1]N. Puri and W. Glower,"Optimal control design via successive approximations," Proceedings of the Joint Automatic Control Conference, 335-344, Philadelphia, June 1967.

[2]D. Kleinman,"On an iterative technique for Riccati equations computations," IEEE Transactions on Automatic Control Vol. 13 114-115, Februrary 1968

[3] J. Baldwin and J. Sims-Williams,"An on-line control scheme using a successive approximation in policy space approach," Journal of Mathematical Analysis and Applications, Vol. 22, 523-536, June 1968.

[4] Z. Gajic, Lyapunov Matrix Equation in Systems Stability and Control, Academic Press, 1995 (Chapter 8, Section 8.1 Kleinman's Algorithm for Riccati Equation, pages 90-95).

[5] E. Vaisbord,"An approximate method for the synthesis of optimal control," Automation and Remote Control Vol. 24, 1626-1632, December 1963.

[6] G. Milhstein,"Successive approximations for solution of one optimal problem," Automation and Remote Control,Vol. 25, 321-329, March 1964