Chapter 19

The Adaptive Dynamic Programming Theorem

John J. Murray, Chadwick J. Cox,* and Richard E. Saeks*

Abstract: The centerpiece of the theory of dynamic programming is the Hamilton-Jacobi-Bellman (HJB) equation, which can be used to solve for the optimal cost functional $V^o$ for a nonlinear optimal control problem, while one can solve a second partial differential equation for the corresponding optimal control law $k^o$. Although the direct solution of the HJB equation is computationally untenable, the HJB equation and the relationship between $V^o$ and $k^o$ serves as the basis for the adaptive dynamic programming algorithm. Here, one starts with an initial cost functional and stabilizing control law pair $(V_0, k_0)$ and constructs a sequence of cost functional/control law pairs $(V_i, k_i)$ in real time, which are stepwise stable and converge to the optimal cost functional/control law pair, for a prescribed nonlinear optimal control problem with unknown input affine state dynamics.

19.1 Introduction

Unlike the many soft computing applications where it suffices to achieve a "good approximation most of the time," a control system must be stable all of the time. As such, if one desires to learn a control law in real-time, a fusion of soft computing techniques (to learn the appropriate control law) with hard computing techniques (to maintain the stability constraint and guarantee convergence) is required. To implement this fused/hard computing approach to control, an adaptive dynamic program-

*This research was performed in part on the National Science Foundation SBIR contracts DMI-9660604, DMI-9860370, and DMI-9983287.

379
ming algorithm, which uses soft computing techniques to learn the optimal cost (or return) functional for a stabilizable nonlinear system with unknown dynamics, and hard computing techniques to verify the stability and convergence of the algorithm was developed in [8], where

- the underlying fusion of soft and hard computing concepts was described,
- the adaptive dynamic programming algorithm was formulated,
- a global convergence theorem for the algorithm with a limited sketch of the proof was introduced, and
- several examples of its application in flight control were presented.

The purpose of this chapter is to provide a detailed proof of the adaptive dynamic programming theorem.

The centerpiece of dynamic programming is the Hamilton-Iacobi-Bellman (HIB) equation [2, 3, 7], which one solves for the optimal cost functional $V^o(x_0, t_0)$. This equation characterizes the cost to drive the initial state $X_0$ at time $t_0$ to a prescribed final state using the optimal control. Given the optimal cost functional, one may then solve a second partial differential equation (derived from the HIB equation) for the corresponding optimal control law $k^o(x, t_0)$, yielding an optimal cost functional/optimal control law pair $(V^o, k^o)$.

Although direct solution of the HIB equation is computationally untenable (the so-called “curse of dimensionality”), the HIB equation and the relationship between $V^o$ and the corresponding control law $k^o$, derived therefrom, serves as the basis of the adaptive dynamic programming algorithm [8]. In this algorithm we start with an initial cost functional/control law pair $(V_0, k_0)$, where $k_0$ is a stabilizing control law for the plant, and construct a sequence of cost functional/control law pairs $(V_i, k_i)$ in real time, which converge to the optimal cost functional/control law pair $(V^o, k^o)$ as follows.

- Given $(V_i, k_i); i = 0, 1, 2, \cdots$; run the system using control law $k_i$ from an array of initial conditions $x_0$, covering the entire state space (or that portion of the state space where one expects to operate the system);
- Record the state $x_i(x_0, \cdot)$ and control trajectories $u_i(x_0, \cdot)$ for each initial condition;
- Given this data, define $V_{i+1}$ to be the cost to take the initial state $x_0$ at time $t_0$ to the final state, using control law $k_i$;
- Take $k_{i+1}$ to be the corresponding control law derived from $V_{i+1}$ via the HIB equation;
- Iterate the process until it converges.

In Sections 19.2 and 19.3, we will show that (with the appropriate technical assumptions) this process is

- globally convergent to
- the optimal cost functional $V^o$ and is
- stepwise stable; i.e., $k_i$ is a stabilizing controller at every iteration with Lyapunov function $V_i$.  