

Proposed Framework for Applying Adaptive Critics in Real-Time Realm

George G. Lendaris¹, Roberto A. Santiago², Michael S. Carroll²

1. Professor of Systems Science and of Electrical & Computer Engineering, lendaris@sysc.pdx.edu

2. Graduate Students, Systems Science Program, robes@sysc.pdx.edu, carroll@pdx.edu

Northwest Computational Intelligence Laboratory, www.nwcil.pdx.edu

Portland State University, PO Box 751, Portland, OR 97207

Abstract

Adaptive Critics have shown much promise for designing optimal nonlinear controllers in an off-line context. Still, their greatest potential exists in the context of reconfigurable control, that is, real time controller redesign in response to (substantial) changes in plant dynamics. To accomplish this, a framework is proposed for the application of adaptive critics in real-time control (for those critic methods requiring a model of the plant). The framework is presented in the context of work being done in reconfigurable flight control by the NW Computational Intelligence Lab (NWCIL) at Portland State University. The proposal incorporates recent work (by others) in fast and efficient on-line plant identification, considerations for bounding the computational costs of converging neural networks, and a novel approach (by us) toward the task of assuring system stability during the adaptation process. The potential and limitations of the proposed framework are discussed. It is suggested that with the recent rapid reduction in computational barriers, only certain theoretical issues remain as the central barriers to successful on-line application of the methods.

1. Background

Reinforcement Learning, and in particular, the Adaptive Critic (AC) methodology based on Dynamic Programming, has been shown to have significant potential for application to the design of controllers for nonlinear systems [4][32][26][31][15][22]. Work reported to date has tended to be applicable in an off-line context. While this may itself be a useful endeavor, to achieve “prime time” status, the approach will have to achieve success in on-line applications, with the ability to (rapidly) adjust the controller design in response to significant changes in dynamics of the plant being controlled. In recognition of this, a **major research objective** in the Adaptive Critic realm is to accomplish such on-line re-design of the controller in response to major changes in plant dynamics (in some circles, this is called ‘reconfigurable control’). Achievement of the following three attributes will be fundamental to success in such an endeavor: 1) reasonable bounds on computational requirements; 2) rapid convergence of the learning/re-design process; and likely the most important, 3) stabil-

ity of the plant at each step of the re-design process.

For those applications where the plant is highly nonlinear, and the operating conditions range over a large portion of state space, most known methods of designing controllers rely on piecewise linear approximations to accomplish the design. Use of neural networks (NNs) whose elements have nonlinear activation functions allows a fully nonlinear controller (e.g., [23]); the challenge is to come up with the design of such a NN controller. The benefit of the AC methodology is its ability to design such nonlinear controllers for nonlinear plants, where the resulting controller is approximately optimal (according to stated design criteria), and to do so over an extended range of state space. The methodology may be described as providing the controller broad-based “experience” during training, so that it can operate successfully over a wide range of operating conditions. In contexts wherein the plant's dynamics change only in a range defined as *nominal* during operation, a reasonable strategy is to use the AC method off-line to design the controller in a way that is robust to such changes, with the resulting design subsequently used on-line. The situation is different when the plant's dynamics may change significantly during operation.

When a plant's dynamics do change significantly, sometimes dramatic changes must be made in the control actions to even keep the system stable. For example, maneuverability and stability of some high performance aircraft are very sensitive to the location of the center of gravity (c.g.). For nominal c.g. positions, it is possible to “trim” the aircraft (make biased deflections of control surfaces) to keep the aircraft in balanced flight. Then if the c.g. shifts a little, new trim adjustments are made to continue the aircraft in balanced flight. Even though such changes in trim bring the aircraft back into balance, the relative effectiveness (in this case, *gain*) of the various control surfaces change. In addition, when aircraft control surfaces are trimmed to accommodate changes in the plant dynamics, there is correspondingly less remaining control range (‘control authority’) available to the pilot. If, for example, there were a sudden, unplanned movement of cargo toward the back of the aircraft, this could result in a shift of c.g. past the nominal range, and make the airplane statically unstable. Then the job of the stability augmentation system is to maintain the aircraft flyable. Significant reduction in available con-

This work was partially supported by the National Science Foundation, Grant ECS-9904378, and by NASA, SBIR Phase III Contract NAS1-01070.

trol authority can make the plane difficult, or in the extreme, impossible to control. For that reason, in cases of changes in the dynamics of the aircraft close to this extreme, the pilot's objective becomes that of landing the aircraft in any fashion possible, while not worrying about "optimal" handling characteristics. I.e., the pilot strives for a "safe fail" (in contrast to fail-safe) mode of operation.

Clearly, the above discussion is not only pertinent to flight control, but to any on-line control context where significant changes in plant dynamics are possible.

2. Adaptive Critic Approach

Adaptive Critics are a specific approach to reinforcement learning, wherein the evaluative feedback for the learner from the environment/teacher is provided by a "critic." While the notion of critic goes back at least to 1959 [30], a very important extension was made in 1983, in which the critic also learns (hence *adapts*) -- with the objective of providing increasingly better evaluative feedback [4].

The application context assumed here is CONTROL, where the controller is the learner. Additionally, we adopt the long-standing quest to design controllers which are 'optimal', where optimal is determined according to some criterion function, with names such as 'cost function' and 'utility function' being typical. Entire books have been written to describe various approaches, results, and methods for designing optimal controllers (e.g., [1][8]). Applications range from the easier task of designing controllers for linear systems (based on quadratic criterion functions), to the more complex design of controllers for non-linear systems. For the linear-quadratic case, complete solutions are available (Ricatti Equation, etc.) and are computationally tractable. For the general non-linear case, the method known as Dynamic Programming [2][11][5] provides a unified mathematical formalism for developing optimal controls. Historically, however, its application has not been computationally tractable. **The good news here** is that with Adaptive Critic methods, a good approximation to the Dynamic Programming method can be implemented with the capability to design a controller in a system with a non-linear plant. Furthermore, in the context of today's extremely cost effective computational power, this can be done in a computationally tractable manner.

From the family of Adaptive Critic designs [32][26] [27] [28], the one chosen for the present work is the **Dual Heuristic Programming (DHP)** method. Central to this method is the use of a critic to estimate not simply a scalar value for the J (Secondary Utility) function of Dynamic Programming, but to estimate the derivatives of this function with respect to each of the states of the system. These values are in turn used with a differentiable plant model to generate derivatives of the J function with respect to the controller's outputs. This rich error feedback to the controller has proven effective in training controllers in many con-

texts, e.g., [15][27][31][38].

3. Proposed Framework

The objective of this paper is to *explore a framework* for applying the DHP Adaptive Critic methodology to the reconfigurable control problem (i.e. the task of on-line redesign of a controller in the context of significant changes in plant dynamics). The re-design is to occur in a manner that assures the "best" controller *under the given circumstances*. In this paper, "best" implies optimal in the realm of nominal (and certain ranges of off-nominal) operation. However, in extreme off-nominal contexts, the best controller may be one that can only ensure safe-fail operation. Figure 1 depicts the seven major components of the proposed framework along with the most significant interdependencies among these components.

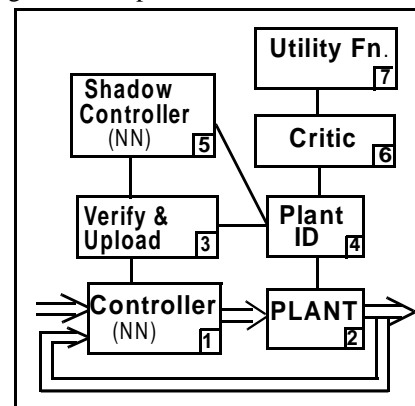


Figure 1. Real-Time Adaptive Critic Controller Re-Design Framework.

Starting from the bottom of the diagram, there is the classic closed loop controller configuration. The Controller (Box #1) and the Plant (Box #2) of this loop are the first two major components of the framework. A central feature of the proposed framework is that three distinct but interdependent processes run in parallel with overlapping use of the components depicted in the diagram.

- 1) The On-line Plant ID process, which provides a continuously updated model of the Plant (Box #4).
- 2) The Controller Re-Design process, which develops a new candidate controller (Shadow Controller, Box #5). The controller re-design is accomplished via the DHP AC procedure, which uses the Plant ID (Box #4), the Critic (Box #6), and the Utility Function (Box #7).
- 3) The Verify Stability & Upload process (Box #3). In this process, stability characteristics of Box #5 are tested via the current Plant Model, and if certain criteria are met, uploads this design into Box #1.

These processes, which comprise the operation of the framework, are next explored in the context of bringing together results from researchers in various sub-fields. Important to this exploration are opportunities where computational requirements, with hopefully provable bounds, can be matched with modern computation capabilities.

Optimism in this context will be balanced by the identification of fundamental research issues that may still present difficulties for the proposed framework.

4. On-line Plant Identification

It is important to note that all processes within the proposed framework rely on the availability of a good plant model. Consequently, the system identification process (see Figure 2) is perhaps the most limiting factor in terms of the quality of performance for the entire framework. While there has been much work done in applying neural networks to system identification, experience suggests that domain specific approaches can provide more accurate solutions at a reduced computational expense.

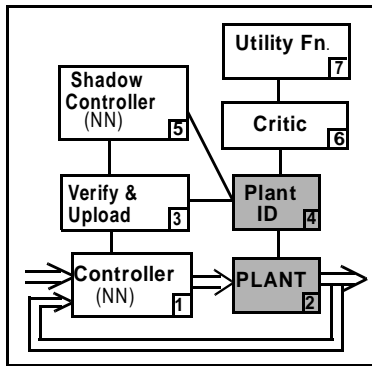


Figure 2. Components for On-Line Plant Identification

For the general context of system identification in the presence of noise, the work of [17] provides a good foundational reference. For the aircraft context, the issue of reducing the number of calculations to accomplish plant identification was addressed in [21]. In that work, the real-time parameter estimation techniques rely on a recursive discrete Fourier analysis, allowing computation to be limited to one addition and two multiplications per variable. Such computational efficiencies will have significant impact on the on-line implementability of the procedure. An important additional attribute claimed in [21] is that the method is robust to noise and data dropout problems. Furthermore, this approach also provides confidence bounds on plant parameter estimates, and these may then be used to assess the overall value of the current Plant Model or to bound computations based on these parameter. Additional relevant system identification work is given in [25][20], where multiple self-organizing feature map NNs are collectively used to model the global response of the plant, and are configured to specifically apply to the control context. Another issue that is fundamental to any on-line identification process is the availability of data that has sufficient information content to allow the identification to occur. Each problem context is different, and problem-specific considerations are in order. Generally speaking, however, there is a concern about whether the data available during normal operation are sufficient. In flight control, for example, the data representing the response of the plane in

steady-state level flight may not contain enough information about the dynamics of the plane to allow robust identification. Additional sensors or algorithmic solutions may be useful in some contexts (e.g., see [18].), but it may also be necessary to inject selected control signals to excite the dynamic modes of the plant. In the case of a dramatic change in plant dynamics, criteria for what testing protocols will be allowable would likely be more lenient (e.g., in the “safe-fail” condition mentioned above).

A strategy that may also be useful here is to develop pre-trained Plant ID models for different possible failure modes (and associated controllers); then, the on-line Plant ID could learn just enough about the plant to select one of the “reserve” models, and then adaptively refine it. (Also, the associated controller would initialize the Box 5 -- cf. next section). This should save time in critical contexts.

5. Controller Re-design

In the proposed framework, the controller re-design process is accomplished by the DHP method. Referring to Figure 3, data from the Utility Function (Box #7) is used for

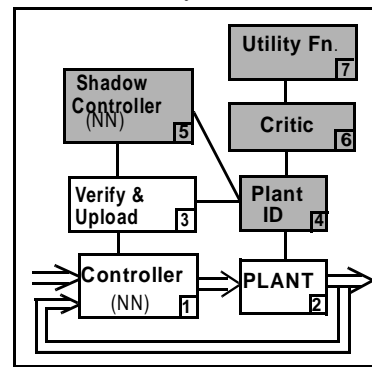


Figure 3. Components for DHP Controller Re-Design Process.

updating the Critic (Box #6), and data provided by the Critic is propagated through the Plant ID (Box #4) to provide a design update for the Shadow Controller (Box #5). The critic and shadow controller components have classically been neural networks, but more recently include other computing paradigms, such as Fuzzy Logic [31]. While the DHP method, and AC methods in general, have been very promising for non-linear controller design, certain difficulties (e.g., with assurance that the design of the candidate controller is stabilizing at each step of the process (called ‘stepwise stability’ in [22]), and with convergence of the design process) have limited their use to off-line contexts. We note that the term ‘convergence’ takes on a double meaning in the context of AC systems: 1) As it relates to the convergence properties of an individual neural network while learning a particular I/O mapping (e.g., for the shadow critic and controller), and 2) As it relates to the convergence properties of the two-loop process involved in AC methods (e.g., see [15] for the DHP case), wherein both the controller and the critic are being iteratively re-designed.

5.1 Convergence of individual neural networks.

A major component of the computational requirements for application of NNs relates to issues of training convergence. In an effort to bound and/or minimize the number of cycles to learn a given I/O mapping, for example, techniques have been developed in the areas of functional link networks [13], and more recently, in neural network initialization via algebraic techniques [9]. It is well known that a multilayer feed forward NN structure whose hidden-layer elements have “squashing” activation functions comprise a Universal Function Approximator [10]. Unfortunately, for multilayer NNs using the basic version of the Backprop algorithm, the number of cycles to convergence is known to be rather large. Accordingly, consideration is often given to single layer NNs with linear output elements, where Delta Rule training is known to require far fewer iterations.

If defined appropriately, single layer NNs can also be Universal Function Approximators. For example, it is known that if the weights that connect the inputs to the hidden layer in a single-hidden-layer feed-forward structure are all randomly selected, and if this layer’s elements have sigmoid activation functions, then the outputs of this layer comprise a legitimate basis set [13][16]. If these randomly assigned weights to the hidden layer are frozen, there remains a *single layer of adjustable weights* in this structure. Accordingly, it may be treated as a ‘single layer NN’ and may be trained with the classical Delta Rule. Such training is known to be very fast, and convergence to a global optimum is guaranteed. To assure high quality approximation, a large number of hidden layer elements may be needed, but in the present computing milieu, this may be just a small additional computational cost. Statements similar to the above may be made regarding Radial Basis Function NNs [24]. In this case, the basis is a set of elements with Gaussian activation functions. The “placement” of the RBF nodes (in state space) is often chosen to be coincident with the training exemplars. The choice of whether to use Sigmoid or Gaussian activations is usually informed by other problem-dependent considerations.

Still, there is reason for caution with single-layer type networks. While information theoretic work such as that in [3] provides support for such basis networks (e.g., Functional Link networks [13], and the particular variety found in [16]), they nevertheless also indicate certain situations in which there would be unreasonable computational cost associated with learning complex and large multivariate mappings. In certain unfortunate circumstances, the number of hidden elements required may grow exponentially [16]. In contrast, if the input layer of weights is also adapted, as in the MLP, then the number of hidden elements may be more manageable. This comes at the cost of not having reasonable bounds or assurances of convergence.

5.2 Computational Issues of Convergence.

Consideration of convergence have become important in recent theoretical and simulation work for Adaptive Critic applications, such as in the areas of dead-zone compensation and tracking [16], and aircraft control [9]. The improved computational characteristics in the aforementioned functional link networks and algebraically initialized neural networks offer promise for placing bounds on the number of iterations required to converge.

The key computational load for a ‘single-layer’ NN structure such as described in the previous Section, is determined by the number of multiply/adds from the inputs to the hidden layer elements. For the sigmoid and Gaussian calculations, standard practice is to use table look up. While the single-layer option normally entails more computations during regular (on-line) operation, with the high computing power now available, the extra computational overhead during on-line operation will likely be manageable, and worthwhile for gaining the improved speed of this convergence during the re-design phase of operation. The indicated computational overhead may also be mitigated by taking advantage of optimized linear-algebra software libraries, dedicated hardware (like digital signal processing chips), and/or parallel processing implementations.

Thus, the following trade-off considerations are in order:

- i.) The number of computations to make a forward pass through the NN during on-line operation.
- ii) The total number of computations required to accomplish an acceptable level of re-design.

So long as the computations in i) may be accomplished within the time constraints of the operational system, then prime consideration should be given to ii). In the rest of this paper, we give this item priority due to the obvious need to accomplish re-design as fast as possible, especially just after a significant change in plant dynamics.

5.3 Convergence of the two-loop Adaptive Critic process.

Convergence of the two-loop Adaptive Critic process is separate from the convergence issue described in Section 5.1. This process entails the Critic (Box #6) converging to a good estimate of the (true) J function corresponding to an optimal controller for the given plant, and the Shadow Controller (Box #5) converging to this optimal controller design. Since the designs of both boxes start with little or no ‘knowledge’ of the ultimate answer, a “bootstrapping” operation takes place wherein the estimate implemented in each box, respectively, of the J function and the corresponding optimal controller, are successively improved in an iterative manner. (Sometimes the term ‘stability’ is used in characterizing the two-loop DHP process, e.g., see [15]. In the context of this article we reserve the term ‘stability’ for characterizing the dynamics of the closed loop system comprising Boxes #1 and #2.) Obtaining bounds on estimates of the number of iterations required for convergence in the

Adaptive Critic processes is still problematic. To date, there exist proofs of convergence to a stabilizing controller only for the DHP and the ADHDP members of the AC family mentioned in Section 2. But, even in these cases, the stability results are limited to the linear quadratic case; and in the case of DHP, a modified training technique is required. Additional work in [16] is promising, and empirical results to date provide optimism for adequately fast convergence (depending on the context, of course).

6. Verify Stability & Upload New Controller Design

In the context of systems where (unrecoverable) instabilities could be life threatening or very expensive, it would not be acceptable to upload the shadow controller design into the closed loop system until the stabilizing properties of the new design are assessed. In the DHP process, while it may be true that the “converged” controller would be a stabilizing one (indeed, that it would be an approximately optimal controller), there is no ready-made assurance that each incremental design of the Controller NN is stabilizing. As a result, two operational modes are here defined: 1) converged operation, where analytical proofs of stability can be invoked, and 2) high-speed simulation operation where, because of time constraints, we must consider using intermediate candidate designs which meet weaker numerical conditions for stability verification (see Figure 4).

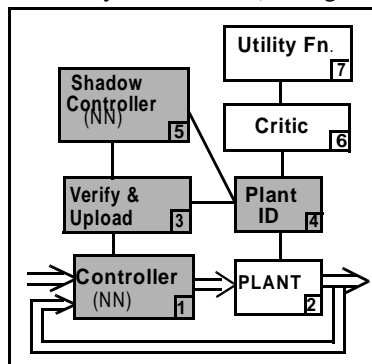


Figure 4. Components for the Verify Stability & Upload New Controller Design Processes

6.1 Converged Operation and Analytical Stability.

Result from [26], indicate that when we allow the DHP process to converge, the resulting controller is Lyapunov stable. A suggestion in [12] is “...(to) combine the training with the stability proof in order to obtain a controller that is provably stable.” Recent work reported in [14] demonstrates this approach, where IQC nonlinear stability analysis is combined with Q-Learning (a different AC method), yielding rather strong assurances of stability. Returning to DHP, during nominal (and a variety of off-nominal) operation of a plant, where plant dynamics are not changing rapidly, experience suggests there exists ample time for the DHP process to converge. As a result, little additional testing is needed to transfer the candidate controller design

from the shadow to the online controller. This is the ideal situation desired to comprise the majority of operation for this real-time adaptive control framework.

6.2 High Speed (Faster than Real Time) Simulation Operation for Controller Re-Design and Stability Verification.

In certain situations, there may not be ample time to allow the DHP process to converge, so consideration must be given to using interim controller designs. In such cases, it will become necessary to test the closed loop response of each interim shadow controller design prior to convergence of the two-loop DHP process. This testing can be accomplished in a faster-than-real-time simulation of the closed loop system with the latest version of the ID model.

A variety of methods are possible. A method currently explored calculates the eigen values of the closed-loop system at the current operating point in state space, and then calculates stability margins. When there is confidence that the design is stabilizing, then it can be upload into Box 1. Unfortunately there is still some question about how strong, and over how much of state space, such a stability test might be. Still, this type of stability verification can be done in a timely fashion with modern computation capabilities.

7. Discussion

While the purpose of this paper is to present an optimistic view of the potential of on-line Adaptive Critic based controller design, it would be irresponsible to ignore some of the possible limitations to this approach. Since this framework relies most centrally on a high-quality plant model, it is particularly vulnerable to problems of inadequate information content in the system identification step. As mentioned in the context of flight control above, sampling of the system states during steady-state operation may not adequately characterize the full dynamic range of the plant. It's also likely that state information from sensors may be highly unreliable during system failures, leaving the system identification process without good data at exactly the times when it needs them the most. While there may be ways to address this problem in many cases, it could present a fundamental limitation to the application of model-based Adaptive Critics to similar control problems.

The fidelity of the ID Model is important (indeed critical) for most of the model-dependent AC method. However, for the DHP method, there is empirical evidence that even rather poor predictive plant models, so long as they give derivatives of correct sign, are adequate for convergence [31]. Since for the proposed framework, an accurate predictive model of the plant plays a crucial role in the stability testing of candidate controllers, when the Plant ID model is known to satisfy only the above derivative condition, the Converged Operation mode would be in order. System ID methods like those of [21] show potential in that they provide confidence intervals which may in turn be used to

bound the results of any subsequent stability analysis, but it remains an empirical question whether such methods can be used reliably in on-line stability analysis.

The problem of scalability is also one that may constrain application of the approach presented here, as it has for many neural network applications. In the context of control applications, the number of inputs and outputs tends to be limited, and as mentioned above, neural network structures can be chosen specifically to limit the computational costs associated with large numbers of hidden nodes. However, there may still be issues with more complex problems. One approach, both for addressing the scaling issue and for reducing the computational load in general, is to apply methods that decompose or partition tasks in the problem domain (e.g., [34]). The beneficial consequence of such an approach in control applications would be that real-time computations would have to be performed only for those parts of the NN currently relevant to the task at hand.

8. Conclusions

The exponential increase in computing power realized in recent years now makes possible approaches to AC based control which would have been unthinkable not long ago. At the center of this process is on-line system identification with time constants short enough to be effective even in time critical situations. Recent work suggests that high speed plant identification can be done, and is therefore constrained not by computing issues, but by the dynamics of the plant itself. Given the ability to develop a plant model quickly, the problem of reconfigurable control becomes one of converging and/or testing a suitable controller. As explained above, there is reason to believe that with careful choice of NN architectures, on-line controller re-design and application of stability testing can be accomplished at speeds that would allow on-line application of the method.

9. References

- [1] Athans, M. & P. Falb (1966), *Optimal Control Theory*, McGraw Hill.
- [2] Bellman, R.E. (1957), *Dynamic Programming*, Princeton Univ.Press.
- [3] Barron, A. R. (1993), "Universal Approximation Bounds for Superpositions of a Sigmoidal Function." *IEEE Transactions on Information Theory*, 39, 930-945.
- [4] Barto, A., R.Sutton & C.Anderson (1983)," Neuron like Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-13, No.5, Sep/Oct.
- [5] Bertsekas, D.P. & J.N. Tsitsiklis (1996), *Neuro-Dynamic Programming*, Athena Scientific.
- [6] Biega, V. & S.N. Balakrishnan (1996), "A Dual Neural Network Architecture for Linear and Nonlinear Control of Inverted Pendulum on a Cart," *Proceedings of 1996 IEEE International Conference on Control Applications*, Dearborn, MI, Sept.
- [7] Bradtke, S.J., B.E. Ydstie & A.G. Barto (1994), "Adaptive Linear Quadratic Control Using Policy Iteration," *Proceedings of the American Control Conference*, Baltimore, MD, June.
- [8] Bryson, Jr. & Y. Ho (1969), *Applied Optimal Control*, Blaisdell Pub.
- [9] Ferrari, S. & R. F. Stengel (2000), "Classical/Neural Synthesis of Nonlinear Control Systems," *Proceedings, AIAA Guidance, Navigation, and Control Conference*, Denver, CO, Aug.
- [10] Hornik, K., M. Stinchcombe & H. White (1989), "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks* vol 2, pp 359-366.
- [11] Howard, R. (1960), *Dynamic Programming and Markov Processes*, MIT Press.
- [12] Hrycej, T. (1997), *Neurocontrol: Toward an Industrial Control Methodology*, Wiley.
- [13] Igelnik, B. & Y-H Pao (1995), "Stochastic Choice of Basis Functions in Adaptive Function Approximation and the Functional-Link Net," *IEEE Transactions on Neural Networks*, vol 6, no6, November.
- [14] Kretchmar, R., P. Young, C.W. Anderson, D. Hittle, M. L. Anderson, C. Delnero (2001), "Robust Reinforcement Learning Control with Static and Dynamic Stability," *Computer Science Technical Report CS-00-102*, CSU, May.
- [15] Lendaris, G. & L. Schultz (2000), "Controller Design (from scratch) Using Approximate Dynamic Programming", *Proc of IEEE-International Symposium on Intelligent Control (IEEE-ISIC'2000)*, Patras, Greece, July.
- [16] Lewis, F.L., J. Campos, & R. Semic (1999), "On Adaptive Critic Architectures in Feedback Control," *Proceedings of 38th Conference on Decision & Control*, Phoenix, AZ, December.
- [17] Maine, R.E. & K.W. Iliff (1981), "Formulation and Implementation of a Practical Algorithm for Parameter Estimation with Process and Measurement Noise," *SIAM J. APPL. MATH*, vol 41, no 3, pp 558-579, December.
- [18] Melody, J.W., D. Pokhariyal, J. Meret, T. Basar, W.R. Perkins & M.B. Brag (2001), "Sensor Integration for In-flight Icing Characterization Using Neural Networks," *Proceedings 39th AIAA Aerospace Sciences Meeting & Exhibit*.
- [19] Miller, W.T. III, R.S. Sutton, & P.J. Werbos, eds., 1990, *Neural Networks For Control*, MIT Press.
- [20] Motter, M.A. (2000), "Predictive Multiple Model Switching Control With the Self-Organizing Map," *Proceedings of the IJCNN'2000, Como, Italy, July*.
- [21] Morelli, E.A. (2000), "Real-Time Parameter Estimation in the Frequency Domain," *Journal of Guidance, Control, and Dynamics*, vol 23, no 5, September-October.
- [22] Murray, J., C. Cox, G. Lendaris, J. Neidhoefer, & R. Saeks (2001), "Adaptive Dynamic Programming," submitted.
- [23] Narendra, K.S. (1996), "Neural Networks for Control: Theory and Practice," *Proceedings Of IEEE*, vol 10(2), pp 18-23.
- [24] Park, J. & I. W. Sandberg (1991), "Universal Approximation using Radial-Basis Function Networks," *Neural Computation*, vol. 3, pp. 246-257.
- [25] Principe, J.C., L. Wang & M.A. Motter (1998), "Local Dynamic Modeling with Self-Organizing Maps with Applications to Nonlinear System Identification and Control," *PROCEEDINGS of IEEE*, vol 86, no 11.
- [26] Prokhorov, R. (1997), *Adaptive Critic Designs and their Applications*, Ph.D. Dissertation, Department of Electrical Engineering, Texas Tech University
- [27] Prokhorov, D., R. Santiago & D. Wunsch (1995), "Adaptive Critic Designs: A Case Study for Neurocontrol," *Neural Networks*, vol 8(9), pp 1367-1372.
- [28] Prokhorov, D. & D. Wunsch (1997), "Adaptive Critic Designs," *IEEE Transactions On Neural Networks*, vol 8(5), pp 997-1007
- [29] Saini, G. & S.N. Balakrishnan (1997), "Adaptive Critic Based Neurocontroller for Autoland of Aircraft with Varying Glideslopes," *Proceedings IJCNN'97*, Houston, TX, IEEE Press.
- [30] Samuel, A.L (1959), "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal Of Research & Development*, vol 3, pp 210-229.
- [31] Shannon, T. & G. Lendaris (1999), "Qualitative Models for Adaptive Neuro-Control", *Proceedings of IEEE-Systems, Man and Cybernetics International Conference (IEEE-SMC'99)*, Tokyo, IEEE Press.
- [32] Werbos, P.J. (1990), "A Menu of Designs for Reinforcement Learning Over Time," Chapter 3 in [19].
- [33] Werbos, P.J. (1992), "Approximate Dynamic Programming for Real-Time Control and Neural Modeling," Chapter 13 in [36].
- [34] Werbos, P.J. (1997), Chapter 18, in *Dealing With Complexity: A Neural Networks Approach*, Karney, Warwick, Kurkova & Taylor, Eds., Springer Verlag, London.
- [35] Werbos, P.J. (1998) "Stable Adaptive Control Using New Critic Designs" URL: xxx.lanl.gov/abs/adap-org/9810001.
- [36] White, D.A. & D.A. Sofge, eds., 1992, *Handbook Of Intelligent Control: Neural, Fuzzy, And Adaptive Approaches*, Van Nostrand Reinhold.
- [37] Zaman, R., D. Prokhorov & D. Wunsch (1997). "Adaptive Critic Design in Learning to Play Game of Go," *Proceedings of IJCNN'97*, Houston, TX, June.
- [38] Visnevski, N. & D. Prokhorov (1996), "Control of a Nonlinear Multivariable System with Adaptive Critic Designs," *PROC Conference Artificial Neural Networks In Engineering -- ANNIE'96*, vol 6, pp 559-56, ASME Press.