01 Oct 2008

Decision Theory on Dynamic Domains Nabla Derivatives and the Hamilton-Jacobi-Bellman Equation

John E. Seiffertt IV
Missouri University of Science and Technology, jes0b4@mst.edu

Donald C. Wunsch
Missouri University of Science and Technology, dwunsch@mst.edu

Suman Sanyal

Decision Theory on Dynamic Domains

Nabla Derivatives and the Hamilton-Jacobi-Bellman Equation

John Seiffertt and Donald C. Wunsch II
Department of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, MO, USA
{jes0b4, dwunsch}@mst.edu

Suman Sanyal
Department of Mathematics and Computer Science
Clarkson University
Potsdam, NY, USA

Abstract— **The time scales calculus, which includes the study of the nabla derivatives, is an emerging key topic due to many multidisciplinary applications. We extend this calculus to Approximate Dynamic Programming. In particular, we investigate application of the nabla derivative, one of the fundamental dynamic derivatives of time scales. We present a nabla-derivative based derivation and proof of the Hamilton-Jacobi-Bellman equation, the solution of which is the fundamental problem in the field of dynamic programming. By drawing together the calculus of time scales and the applied area of stochastic control via Approximate Dynamic Programming, we connect two major fields of research.**

***Keywords*—approximate dynamic programming, time scales, reinforcement learning, Hamilton-Jacobi-Bellman equation**

I. INTRODUCTION

The mathematics of time scales seeks to bring together in an organized and rigorous way the discrete and continuous domains [29]. This calculus establishes a unified framework for analysis of both difference equations and differential equations. Such *dynamic equations* on time scales ([16],[17]) have been applied in population biology [11], quantum calculus [12], boundary value problems [20], real time communications networks [24], intelligent robotic control [25], financial engineering [38], and switched linear circuits [33], among others. Due to its reconciliatory nature with respect to the discrete and the continuous, the time scales calculus admits a suite of dynamic derivatives. The standard delta derivative most closely mirrors the derivative found in traditional analysis. Other derivatives, such as the alpha, nabla, and diamond-alpha, are also widely studied ([1], [17], [37]). This current work focuses on the nabla derivative.

Dynamic programming [4] provides a method for generating optimal solutions for multi-stage decision processes. The standard algorithm for dynamic programming involves a computationally intensive backwards induction update rule. In [39], we extend this algorithm to general time scales using the delta derivative. It is common to find, however, that industrial-scale applications prove intractable when attacked using this algorithm. Therefore, in the face of impossible optimality, much research searches for suboptimal policies. The field of Approximate Dynamic Programming (ADP) considers these approaches ([21], [26]).

ADP, also known as a branch of the more genral paradigm of reinforcement learning in various literature [21], seeks to solves the Hamilton-Jacobi-Bellman equation. In discrete time, backwards induction is often used. For continuous time domains the HJB equation must be consulted in its partial differential equation form. In this paper, we extend the HJB equation to dynamic domains, those consisting of both discrete and contiuous areas. Furthermore, in this pursuit we employ the use of the nabla derivative

We assume familiarity with dynamic programming, and will briefly sketch our notation and assumptions here. The requirements of our dynamic programming framework are as follows: a time scale $\mathbb{T}$ in which our decision points lie, controls $u(x(t),t)$, a stochastic disturbance $w(t)$, states $x(t)$ which evolve according to a rule $f(x(t),u(x(t),t),w(t),t)$, and a cost/reward $r(x(t),u(x(t),t),w(t),t)$ where the cost at a terminal decision point T is piecewise defined as $r_T(x(T))$. A policy π is a set of state-control pairs for each point in $\mathbb{T}$ such that each control is valid for both the state and time. We denote by π^t the tail of the policy π beginning with time step t. We also introduce a cost-to-go function given by

$$J_\pi(x(t_0),t_0) = E\left\{\int_{t_0}^{T} r(x(\tau),u(x(\tau),\tau),w(\tau),\tau)\Delta\tau\right\} \qquad (1)$$

which measures the expected cost of a policy π. We assume these expected values are finite and well-defined.

We limit $w(t)$ to take on values in a countable set. While this constraint prohibits the use of disturbances such as Gaussian noise or Brownian motion it does permit models which can find useful application and is not a case of mathematicizing an approach into obscurity. For example, the representation of state-space systems in Markov decision processes [36] gives the $w(t)$ the form of *transition probabilities* $P(i,j,u)$ which indicate the probability the system evolves from state x_i to state x_j in response to control signal c. Such $w(t)$ are countable.

We consider the following state-space dynamical system defined on a time scale $\mathbb{T}$:

$$x^{\Delta}(t) = f(x(t), u(x(t), t), w(t), t) \quad (2)$$

where $t \in \mathbb{T}$. Our task is to calculate a policy π which minimizes the cost-to-go function J_π. We call such a π an *optimal policy* and denote the optimal cost-to-go as $J^*(x(t_0), t_0) = \min_\pi J_\pi(x(t_0), t_0)$, where the minimum is considered over all policies.

Bellman's Principle of Optimality aids in the solution to the optimization problem. This principle can be framed in the following way. Let π^* be an optimal policy. Then the optimal policy for the tail problem starting at time *n*, which is to minimize

$$E\left\{\int_n^T r(x(\tau), u(x(\tau), \tau), w(\tau), \tau)\Delta\tau\right\}, \quad (3)$$

is equal to the portion of π^*which overlaps π^n.

The dynamic programming algorithm, a form of backwards induction, involves stochastic optimization of control selection starting from the terminal time point T. Beginning with setting $J(x(T), T) = r_T(x(T))$, the algorithm proceeds via the following update rule:

$$J(x(t), t) = \min_c E\{r(x(t), u(x(t), t), w(t), t) + J(f(x(t), u((x(t), t), w(t), t), \sigma(t))\}$$

for $t \in \mathbb{T}$. This rule says that the cost-to-go of the current state $x(t)$ under a control $u(x(t), t)$ equals the expected value of the immediate cost $r(x(t), u(x(t), t), w(t), t)$ plus the future costs $J(f(x(t), u((x(t), t), w(t), t), \sigma(t))$. Recall that $\sigma(t)$ is the "next" point in our time scale $\mathbb{T}$ and the use of this forward-jump operator constitutes one of the key tools injecting richness into the unification of discrete and continuous domains.

It is our task in this paper to move beyond this dynamic programming algorithm and to consider the time scales extension via nabla derivatives of the full continuous version of the HJB partial differential equation. As such, this paper adds to our results in [39] by extending the Hamilton-Jacobi-Bellman equation to the case of the nabla dynamic derivative.

This section has introduced the dynamic programming paradigm and its relation to the time scales calculus. Section II reviews this calculus, Section III contains our proof of the Hamilton-Jacobi-Bellman equation using nabla derivatives, and Section IV concludes the paper with perspectives on merging this increasingly relevant area of mathematics with computational control and decision theory.

II. THE TIME SCALES CALCULUS

A. Fundamentals

We typically think of functions defined on a domain that is either entirely discrete or continuous. The study of time scales allows us to consider functions on *dynamic domains* which can be a mixture of the two. As such, we discuss *dynamic equation* in a general sense rather than specifying difference or differential equations.

A generalized time scale $(\mathbb{T}, \alpha)$ is such that $\mathbb{T} \subset \mathbb{R}$ is nonempty and every Cauchy sequence in $\mathbb{T}$ converges to either a point within $\mathbb{T}$ or to a finite infimum or supremum of $\mathbb{T}$ and α is a function from $\mathbb{T}$ into $\mathbb{T}$. Often in the time scales calculus we consider $\mathbb{T}$ to be any nonempty cosed subset of the real line and set α equal to the *forward jump operator* σ given by $\sigma(t) = \inf\{x \in \mathbb{T}: x > t\}$. However, we are not technically restricted to such domains in the general case.

In the nabla calculus we study time scales of the form $(\mathbb{T}, \rho)$, where ρ is the *backwards jump operator* defined by $\rho(t) = \sup\{x \in \mathbb{T}: x < t\}$. We can think of these jump operators σ and ρ as allowing us to move through the time scale, with σ taking us to the "next" element and ρ returning us to the "previous" element. For example, when $\mathbb{T} = \mathbb{R}$, then $\sigma(t) = \rho(t) = t$. For a discrete time scale such as the integers $\mathbb{T} = \mathbb{Z}$ then the jump operators give us elements $\sigma(t) = t + 1$ and $\rho(t) = t - 1$ as we would expect. These concepts begin to yield higher rewards when we consider time scales outside our routine mathematical experience. Consider, for example, the q-time scale studed in the quantum calculus [32]: $\mathbb{T} = q^{\mathbb{N}_0}$ (where $q > 1$). The forward and backward jump operators are given by $\sigma(t) = q^{t+1}$ and $\rho(t) = q^{t-1}$, respectively. We find the need to use the functions σ and ρ with such frequency that it is convenient to define two new symbols f^σ and f^ρ as follows: $f^\sigma(t) = f(\sigma(t))$ and $f^\rho(t) = f(\rho(t))$.

Also associated with each time scale is a *graniness function* μ which measures the uniformity of the domain. For continuous domains the graininess is $\mu(t) = 0$ and for the integers the graininess is $\mu(t) = 1$. This graininess function is defined by $\mu(t) = \sigma(t) - t$. This function is highly non-differentiable and plays an importat role in many formulas of the time scales calculus, defining as it does perhaps the most critical characteristic of the dynamic domain.

The notion of *density* is important when we allow both discrete and continuous intervals in our domain. Formally, this concept is defined as follows: a point $t \in \mathbb{T}$ is *right-scattered* if $\sigma(t) > t$, *left-scattered* if $\rho(t) < t$, *isolated* if it is both right- and left-scattered, *right-dense* if $t < \sup \mathbb{T}$ and $\sigma(t) = t$, *left-dense* if $t > \inf \mathbb{T}$ and $\rho(t) = t$, and, finally, if t is both right- and left- dense then we say simply that t is dense.

B. The Time Scales Calculus of a Single Variable

The usual derivative of the time scales calculus is defined as follows. Let $f: \mathbb{T} \to \mathbb{R}$ be a function. Then the delta derivative $f^\Delta(t)$ of f at a point $t \in \mathbb{T}^\kappa$, where $\mathbb{T}^\kappa$ coincides with $\mathbb{T}$ except at a left-scattered maximum, if one should exist, is defined to be the number such that given $\varepsilon > 0$ there is a neighborhood U of t such that

$$|[f(\sigma(t)) - f(s)] - f^\Delta(t)[\sigma(t) - s]| \leq \varepsilon|\sigma(t) - s|$$

for all $s \in U$, where *neighborhood* is defined such that $U = (t - \delta, t + \delta)$ for some $\delta > 0$. Note that this follows the classical definition of the derivative, with the traditional $x + h$ increment replaced by the forward jump operator $\sigma(t)$. This sort of translation is a common theme in the calculus of time scales. The delta derivative $f^{\Delta}(t)$ becomes $f'(t)$ when $\mathbb{T} = \mathbb{R}$ and becomes the standard difference operator on $\mathbb{T} = \mathbb{Z}$.

This derivative is a forward measure. Our results in [39] deal with the Hamilton-Jacobi-Bellman equaion and dynamic programming using this derivative. Here our attention is drawn to the backwards derivative, defined as follows. Let $f: \mathbb{T} \to \mathbb{R}$ be a function. Then the nabla derivative $f^{\nabla}(t)$ of f at a point t is given by the number, provided it exists, such that given $\varepsilon > 0$ there exists a neighborhood U of t such that

$$|[f(\rho(t)) - f(s)] - (\rho(t) - s)f^{\nabla}(t)[\sigma(t) - s]| \\ \leq \varepsilon|\rho(t) - s|$$

for every $s \in U$. At left-scattered points this becomes the left difference operator found in the traditional study of difference equations

Along with derivatives we study anti-derivatives. The Fundamental Theorem of Calculus is valid on time scales and takes a familiar form:

$$\int_a^b f^{\Delta}(t)\Delta t = f(b) - f(a) \tag{4}$$

and for nabla derivatives we have

$$\int_a^b f^{\nabla}(t)\nabla t = f(b) - f(a) \tag{5}$$

We do not delve deeply into the theory of integration on time scales in this paper. Instead, the presentation of these formulas will suffice for our proof of the Hamilton-Jacobi-Bellman equation in the nabla calculus. For a thorough overview of integration theory on time scales the reader is directed to [14], [16], and [27].

Not all results from the traditional calculus can be proven to hold on time scales, however. One critical and glaring example is that the classical chain rule fails on general time scales. Potzche [15] gives the most general version of this chain rule. It turns out that many results of standard analysis, including the Hamilton-Jacobi-Bellman equation, are dependent on the chain rule as conceived on continuous time scales. Therefore, extending notions which apply to the movement of particles through space to areas such as an agent's decision process through time may very well require the development of new mathematical structures to accommodate the abstraction.

C. The Time Scales Calculus of Multiple Variables

We use a definition of partial derivatives on time scales given by Jackson [31]. Let $\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_n$ be time scales, set $\mathbb{T} = \mathbb{T}_1 \times \mathbb{T}_2 \times \cdots \times \mathbb{T}_n$, and let $f: \mathbb{T} \to \mathbb{R}$ be a function. Define the operators on $\mathbb{T}$ as $\sigma(t) = \big(\sigma(t_1), \sigma(t_2), \ldots, \sigma(t_n)\big)$ and $\rho(t) = \big(\rho(t_1), \rho(t_2), \ldots, \rho(t_n)\big)$. Also define $\mathbb{T}^{\kappa} = \mathbb{T}_1^{\kappa} \times \mathbb{T}_2^{\kappa} \times \cdots \times \mathbb{T}_n^{\kappa}$, $f^{\sigma_i(t)} = f(t_1, t_2, \ldots, t_{i-1}, \sigma_i(t_i), t_i, \ldots, t_n)$, $f^{\rho_i(t)} = f(t_1, t_2, \ldots, t_{i-1}, \rho_i(t_i), t_i, \ldots, t_n)$ and $f_i^s = f(t_1, t_2, \ldots, t_{i-1}, s, t_i, \ldots, t_n)$.

The *partial delta derivative of f at t with respect to t_i* is the number f^{Δ_i}, provided it exists, such that given any $\varepsilon > 0$ there exists a neighborhood U of t_i for $\delta > 0$ such that

$$|[f^{\sigma_i}(t) - f_i^s(t)] - f^{\Delta_i}(t)[\sigma_i(t) - s]| \leq \varepsilon|\sigma_i(t) - s|$$

for all $s \in U$, where *neighborhood* is defined such that $U = (t_i - \delta, t_i + \delta) \cap \mathbb{T}_i$.

Similarly, the *partial nabla derivative of f at t with respect to t_i* is the number f^{∇_i} such that given $\varepsilon > 0$ there exists a neighborhood U of t_i for $\delta > 0$ such that

$$|[f^{\rho}(t) - f_i^s(t)] - f^{\nabla_i}(t)[\rho_i(t) - s]| \leq \varepsilon|\rho_i(t) - s|$$

We can speak of higher order partials in the normal way. We can even consider mixed nabla and delta partials. Further details can be found in [15]. For our purposes we only need to consider the nabla case.

Fundamental to our proof of the Hamilton-Jacobi-Bellman equation in the nabla calculus will be the chain rule. This chain rule is an extension of the one presented in [15]. Let $t \in \mathbb{T}$, $x: \mathbb{T} \to \mathbb{R}$, $y: \mathbb{T} \to \mathbb{R}$, $x(\mathbb{T}) = \mathbb{T}_x$, $y(\mathbb{T}) = \mathbb{T}_y$, and $F\big(x(t), y(t)\big)$. Assume $x\big(\rho(t)\big) = \rho_x(x(t))$ and $y\big(\rho(t)\big) = \rho_y(t)$. If $F = f(x(t), y(t))$ is ρ_x-completely differentiable and x and y are differentiable, then

$$F^{\nabla}(t) = f^{\nabla_x}\big(x(t), y(t)\big)x^{\nabla}(t) \\ + f^{\nabla_y}\big(\rho_x(x(t)), y(t)\big)y^{\nabla}(t). \tag{7}$$

For more details on partial derivatives on time scales, includingthe reader is directed to [2] and [15].

I. Nabla Derivative Formulation

The Hamilton-Jacobi equation is a result of the calculus of variations and work extending this calculus to time scales is in its infancy [8]. These problems typically take the general form of minimizing the cost functional given by the following integral [10]:

$$J(y) = \int_a^b L(t, y^{\sigma}(t), y^{\Delta}(t))\Delta t. \tag{8}$$

From this the usual Euler and Legendre conditions can be derived on time scales. Our next result takes this a step further and proves the Hamilton-Jacobi equation for an alternate version, given by

$$J(x(t_0), t_0) = \int_{t_0}^{T} r(x(t), u(t))\Delta t, \tag{9}$$

of the above integral (8). Since equation (9) is the common cost functional of dynamic programming the resulting equation is given the name Hamilton-Jacobi-Bellman. In this way, the following theorem is a contribution to the development of the calculus of variations on time scales as well as to ADP. However, as we prove the HJB equation for a form other than that given by (8), there is still work to be done on Hamilton-Jacobi equations for more generalized cost functionals.

Consider now the system

$$x^{\nabla}(t) = f(x(t), u(t)) \tag{10}$$

where x represents states and u is the control. Let $t \in \mathbb{T}$, $x: \mathbb{T} \to \mathbb{R}$, and $x(\mathbb{T}) = \mathbb{T}_x$. The cost-to-go function $J: \mathbb{T}_x \times \mathbb{T} \to \mathbb{R}$ is given by

$$J(x(t_0), t_0) = \int_{t_0}^{T} r(x(t), u(t)) \nabla t \tag{11}$$

where t_0 is the initial decision point and $r(x(t), u(t))$ is the cost. Assume J is delta-differentiable and x is ρ_x-completely delta differentiable. Furthermore, require x to satisfy

$$x(\rho(t)) = \rho_x(x(t)). \tag{12}$$

Then the HJB equation on time scales is given by

$$0 = \min_{c}\Big(r\big(x(t), u(x(t), t)\big) + J^{\nabla_t}(x(t), t) + J^{\nabla_x}(x(t), \rho(t)) f(x(t), t)\Big). \tag{13}$$

This is an equation that any optimal policy of our minimization problem must satisfy. Since precious few industrial-scale applications admit an analytic solution of this equation, ADP is employed to develop approximation techniques for this purpose. The proof of this equation is our next theorem.

Theorem:

Let $V(x(t), t)$ be a solution to equation (15) such that

$$0 = \min_{c}\Big(r\big(x(t), u(x(t), t)\big) + V^{\nabla_t}(x(t), t) + V^{\nabla_x}(x(t), \rho(t)) f(x(t), t)\Big). \tag{14}$$

Assume the boundary condition $V(x(T), T) = r_T(x(T))$ and $\hat{x}(t_0) = x(t_0)$ and suppose $u^*(x(t), t)$ attains the minimum called for in equation (14) for all states and all time. Let $x^*(t)$ be the state trajectory, subject to the condition $x^*(t_0) = x(t_0)$, that corresponds to applying the controls $u^*(x(t), t)$ at each decision point t.

Then the function $V(x(t), t)$ is the optimal cost-to-go function $J^*(x(t), t)$ and the control $u^*(x(t), t)$ is optimal.

Proof:

Let $\hat{u}(x(t), t)$ be a control policy with state trajectory $\hat{x}(t)$. Our goal is to show that the policy $u^*(x(t), t)$ achieves a cost equal to at most this arbitrary $\hat{u}(x(t), t)$which will mean that $u^*(x(t), t)$ is our optimal control. We begin using equation (14) to give

$$0 \le r\big(\hat{x}(t), \hat{u}(x(t), t)\big) + V^{\nabla_t}(\hat{x}(t), t) + V^{\nabla_x}(\hat{x}(t), \rho(t)) f(\hat{x}(t), t). \tag{15}$$

Noting that, via (10), we have $x^{\Delta}(t) = f(x(t), u(t))$, we can rewrite (15) as

$$0 \le r\big(\hat{x}(t), \hat{u}(x(t), t)\big) + V^{\nabla_t}(\hat{x}(t), t) + V^{\nabla_x}(\hat{x}(t), \rho(t)) x^{\nabla}(t). \tag{16}$$

and, by using the chain rule we can rewrite as

$$0 \le r\big(\hat{x}(t), \hat{u}(x(t), t)\big) + V^{\nabla}(t). \tag{17}$$

Integrating over the time horizon yields

$$0 \le \int_{t_0}^{T} r\big(\hat{x}(t), \hat{u}(x(t), t)\big) \nabla t + \int_{t_0}^{T} V^{\nabla}(t) \nabla t. \tag{18}$$

Using the fundamental theorem (5) gives

$$0 \le \int_{t_0}^{T} r\big(\hat{x}(t), \hat{u}(x(t), t)\big) \nabla t + V(\hat{x}(T), T) - V(\hat{x}(t_0), t_0).$$

Substituting in our boundary conditions $V(x(T), T) = r_T(x(T))$ and $\hat{x}(t_0) = x(t_0)$ gives us

$$0 \le \int_{t_0}^{T} r\big(\hat{x}(t), \hat{u}(x(t), t)\big) \nabla t + r_T(x(T)) - V(x(t_0), t_0)$$

which can be rewritten as

$$V(x(t_0), t_0) \le \int_{t_0}^{T} r\big(\hat{x}(t), \hat{u}(x(t), t)\big) \nabla t + r_T\big(x(T)\big).$$

From our hypothesis, we assume the controls $u^*(x(t), t)$ and their corresponding state trajectory $x^*(t)$ minimize the value function $V(x(t), t)$. Using this information and the initial condition $x^*(t_0) = x(t_0)$, we can replace the inequality with equality in the case of these quantities:

$$V(x(t_0), t_0) = \int_{t_0}^{T} r\big(x^*(t), u^*(x(t), t)\big) \nabla t + r_T\big(x^*(T)\big).$$

Combining with the previous equation, we have

$$\int_{t_0}^{T} r\big(x^*(t), u^*(x(t), t)\big) \nabla t + r_T\big(x^*(T)\big) \le \int_{t_0}^{T} r\big(\hat{x}(t), \hat{u}(x(t), t)\big) \nabla t + r_T\big(x(T)\big).$$

This equation tells us that the cost of the policy $u^*(x(t), t)$ is less than or equal to the cost of any admissible policy $\hat{u}(x(t), t)$. We conclude the policy $u^*(x(t), t)$ is optimal and that, since $\hat{u}(x(t),t)$ is arbitrary, we have $V(x(t),t) = J^{*}(x(t),t)$. Therefore, any optimal policy must satisfy the HJB equation given by (14).■

II. Conclusion And Perspectives

The calculus of time scales in general, and the nabla calculus studied in this paper in particular, is an increasingly relevant and emerging area of mathematics with wide-ranging opportunities for application. We have established that the Hamilton-Jacobi-Bellman equation obtains using the nabla calculus. This equation forms the foundation of Approximate Dynamic Programming, which is now extended to the nabla calculus as well.

Simulations and other computationally grounded research and modeling in time scales remains a promising and open arena. Of particular utility is the Time Scales MatLab Toolbox from the Baylor University Time Scales Group [5]. Also of need are demonstrations of ADP-based controllers operating in a time scales framework in an applied and meaningful setting. Interesting here, too, would be examples of the distinctions among the classes of dynamic derivatives which populate the time scales calculus.

It is our position that while the study of time scales can be used in the development of a theoretical unification of decision theory on domains which are both discrete and continuous in parts, it has at the same time the potential to be utilized far beyond such purposes. We believe that there are important application areas where dealing simultaneously with discrete and continuous variables is critical ([42], [43]) and that the time scales calculus provides a natural and powerful framework for such exploration.

Acknowledgement

This work was supported in part by the Mary K Finley Endowment at the Missouri University of Science and Technology and in part by the National Science Foundation.

References

[1] Ahlbrandt, C., Bohner, M., & Ridenhour, J. (2000) Hamiltonian systems on time scales. *J. Math. Anal. Appl.* 578. pp 250-561.

[2] Ahlbrandt, C., & Morian, C. (2002) Partial differential equations on time scales. *Journal of Computational and Applied Mathematics.* 141(1-2). pp 35-55.

[3] Baylor University Time Scales Group. (2008) Ti.me.. Sc.a.. .les MATLAB Toolbox v. 1.1. retrieved 2-7-2008 at http://www.timescales.org

[4] Bertsekas, D. (2000) Dynamic Programming and Optimal Control, Second Edition, Vols 1 and 2. Athena Scientific. Belmont, MA.

[5] Bohner, M. (2004) Calculus of variations on time scales. *Dynamic Systems and Applications*. 13. pp 339-349.

[6] Bohner, M., Fan, M., & Zhang, J. (2007) Periodicity of scalar dynamic equations and applications to population models. *Journal of Mathematical Analysis and Applications,* 330(1). pp 1-9.

[7] Bohner, M., & Hudson, T. (2007) Euler-type boundary value problems in quantum calculus *International Journal of Applied Mathematics and Statistics* 9(J07). pp 19-23.

[8] Bohner, M., & Guseinov, G. (2005) Multiple integration on time scales. *Dynamic Systems and Applications*, 14(3-4). pp 579-606.

[9] Bohner, M., & Guseinov, G. (2004) Partial Differentiation on Time Scales. *Dynamic Systems and Applications*. 13. pp 351-379.

[10] Bohner, M. & Peterson, A. (2003) Advances in Dynamic Equations on Time Scales. Birkhauser. Boston, MA.

[11] Bohner, M. & Peterson, A. (2001) Dynamic Equations on Time Scales: An Introduction with Applications. Birkhauser. Boston, MA.

[12] DaCunha, J., Davis, J., & Singh, P. (2004) "Existence results for singular three point boundary value problems on time scales," *J. Math. Anal. Appl*. 295. pp 378–39.

[13] Gravagne, I., Marks, R., Davis, J., & DaCunha, J. (2004) Application of time scales to real time communications networks. *American Mathematical Society Western Section meeting, special session on Time Scales*. University of Southern California.

[14] Gravagne, I., Davis, J., & Marks, R. (2005) How deterministic must a real-time controller be? *Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems,* Alberta, Canada.Aug. 2-6, 2005, pp. 3856 – 3861

[15] Guseinov, G. (2003) Integration on time scales. *J. Math. Anal. Appl*., Vol. 285. pp.107-127.

[16] Hilger, S. (1990) Analysis on measure chains—a unified approach to continuous and discrete calculus. *Results Math.* 18. pp 18-56.

[17] Jackson, B. (2006) Partial dynamic equations on time scales. *Journal of Computational and Applied Mathematics.* 186. pp 391-415.

[18] Kac, V., & Pokman, C. (2001) Quantum Calculus. Springer. New York, NY.

[19] Marks, R., Gravagne, I., Davis, J., & DaCunha, J. (2006) Nonregressivity in switched linear circuits and mechanical systems. *Mathematical and Computer Modelling*. 43. pp 1383-1392.

[20] Potzsche, C. (2002) Chain rule and invariance principle on measure chains. *J. Comput Appl Math.* 141. pp 249-254.

[21] Powell, W. (2007) Approximate Dynamic Programming: Solving the Curses of Dimensionality. Wiley Series in Probability and Statistics. Hoboken, NJ.

[22] Puterman, M. (*1994*) Markov Decision Processes. Discrete Stochastic Dynamic Programming. Wiley Series in Probability and Mathematical Statistics. New York, NY.

[23] Rogers, J., & Sheng, Q. (2007) "Notes on the diamond-α dynamic derivative on time scales," J. Math. Anal. Appl. 326. pp 228–241.

[24] Sanyal, Suman. (2008) Stochastic Dynamic Equations. *PhD Dissertation.* Missouri University of Science and Technology.

[25] Seiffertt, J., Sanyal, S., & Wunsch, D. (2008) Hamilton-Jacobi-Bellman equations and approximate dynamic programming on time scales. *IEEE Transactions on Systems, Man, and Cybernetics, Part B.* (to appear.)

[26] Si, J., Barto, A., Powell, W., & Wunsch, D. (2004) Handbook of Learning and Approximate Dynamic Programming. IEEE Press Series on Computational Intelligence.

[27] Werbos, P. (2005) Backwards differentiation in AD and neural nets: past links and new opportunities. in Bucker, M., Corliss, G., Hovland, P., Naumann, U., & Boyana, N. (eds) *Automatic differentiation: applications, theory, and implementations.* Springer. New York, NY.

[28] White, D., & Sofge, D. (eds) (1992) Handbook of Intelligent Control. Van Nostrand.

.