How Deterministic Must a Real-Time Controller Be?

Ian A. Gravagne*, John M. Davis, Robert J. Marks

Abstract

Real-time computing platforms are ubiquitous in the fields of automation, robotics and control. These are often based on a real-time operating system that offers a high level of determism to support processes such as discrete-event and feedback control. Using methods derived the theory of dynamic systems on time scales, this paper discusses through modeling and simulation the effects of non-determinism. The paper's conclusion supports the hypothesis that strict determinism is not always required to obtain adequate overall system performance.

1 Introduction

Often, designers of modern engineering systems such as robotics and automation systems face a challenge when selecting a computing platform on which to base the system controller. In the very simplest of systems, the controller may be a small microprocessor responding to a periodic interrupt. In more complex systems, the controller may host an operating system that supports multi-threading, a file system, networking components and more. Typical choices include QNX, RT-Linux, μCOS, HyperKernel, RTX and others. The principal feature offered by these platforms is determinism, the ability to respond to an event fast and predictably.

However, even advanced real-time operating systems are not 100% deterministic. Furthermore, the processes to be controlled do not always require a high level of determinism, for example, systems with relatively slow dynamics or high error tolerances. Also, determinism is particularly difficult to guarantee in distributed controller networks, which are garnering significant interest as their use skyrockets in certain automation fields including manufacturing, robotics and automotive [7]-[11]. While special scheduling techniques have been developed to alleviate problems related to blocking and non-determinism in networked control situations [1][11], the fact remains that a high degree of pseudo-random timing variation exists in virtually all but the simplest controllers.

*Corresponding Author, School of Engineering and Computer Science, Baylor University. Email: Ian_Gravagne@baylor.edu

This paper examines the effects of timing variation in feedback controllers using advances in a relatively new field of mathematics called *dynamic equations on time scales*. The theory permits dynamic systems to be modeled and analyzed independently of the underlying time domain, allowing for a rigorous and holistic study of non-uniformly sampled systems. The central hypothesis of the paper is that real-time computational platforms may often not need to be as deterministic as currently thought; simulation examples are provided in support of this hypothesis.

Readers not already familiar with time scale theory are encouraged to read the appendix.

2 Time Scale System Analysis

We start with the assumption that the plant to be controlled can be approximated by a linear system of the form

$$\begin{aligned} \dot{x} &= Ax + Bu, \quad A \in \mathbb{R}^{n\times n}; B \in \mathbb{R}^{n\times m} && (1)\\ u &= Kx, \quad K \in \mathbb{R}^{m\times n}, && (2) \end{aligned}$$

and, for simplicity, that full-state feedback control is available to make the system behave as desired. It is recognized that linear systems represent but a small fraction of the variety of dynamical systems encountered in robotics-related fields. However, time scale theory has not yet advanced to the point where complex nonlinear systems can be considered. This is an area of ongoing research.

We next discretize to an isolated time scale $\mathbb{T}$, consisting of possibly non-uniformly spaced points with unknown graininess. A discretization process capturing typical sample-and-hold behavior, described in [4], yields

$$\begin{aligned} x^{\Delta}(t) &= \operatorname{expc}(\mu(t)A)(A+BK)x(t) \\ &\vdots \quad = \mathcal{A}(\mu(t))x(t), \quad t \in \mathbb{T}. && (3) \end{aligned}$$

The "expc" function is defined as

$$\operatorname{expc}(X) := I + \frac{1}{2}X + \frac{1}{6}X^2 + ... + \frac{1}{n!}X^{n-1} + ..., \quad (4)$$

a convergent power series that behaves similarly to an exponential. It has several interesting and useful

properties:

P1: $\text{expc}(X) = (e^X - I)X^{-1}$ when X^{-1} exists,
P2: $\text{expc}(j\omega) = e^{j\omega/2}\,\text{sinc}(\omega/2)$
P3: $\text{expc}(X) \to I$ as $X \to 0$

Property P1 and P2 give the the function its name (the "exponent cardinal"). Property P3 shows that, in the special case of continuous time with $\mathbb{T} = \mathbb{R}$ and $\mu \equiv 0$, the expression $\text{expc}(\mu(t)A)$ reduces to an identity matrix, giving $\dot{x} = (A + BK)x$ as expected in (3).One can now see that the graininess $\mu(t)$ – the distance from the current sampling time to the next sampling time – is akin to the sample period and may vary dynamically, depending on the determinism of the system.

It is useful at this point to revisit some fundamental results regarding the stability of linear systems on arbitrary time scales. The authors of [6] prove that time-invariant linear systems, or a time-varying linear systems that are Jordan reducible, will be exponentially stable if and only if every system eigenvalue belongs to the *set of exponential stability* $\mathcal{S}(\mathbb{T})$. A subset of $\mathcal{S}(\mathbb{T})$, of interest here, is defined as

$$\begin{aligned}\mathcal{S}_{\mathbb{C}}(\mathbb{T}) &= \{\eta \in \mathbb{C} : \alpha = \\ &\limsup_{t\to\infty} \frac{1}{t-t_0}\int_{t_0}^{t} \frac{\ln|1+\mu(\tau)\eta(\tau)|}{\mu(\tau)}\Delta\tau < 0\}\end{aligned} \tag{5}$$

where ln is the natural logarithm. Looking at (5), it becomes apparent that the integrand must be negative on average. Furthermore, under the condition that $\mu(t) > 0$ on a time scale with no limit points, as in this paper, it is shown in the author's previous work [5] that a proper subset of $\mathcal{S}_{\mathbb{C}}(\mathbb{T})$ is the set of all $\eta \in \mathbb{C}$ with

$$H_k(t) := \prod_{\tau=\rho^k(t)}^{t} |1 + \mu(\tau)\eta(\tau)| < 1. \tag{6}$$

In effect, (6) conservatively approximates the infinite integral of (5) over a moving window of k points.

The region defined by $|1 + \mu\eta| < 1$ in (5) is termed the *Hilger Circle*, a circle in the left-hand complex plain that passes through the origin, with center at $-\frac{1}{\mu}$. Note that, for systems of non-constant graininess, the Hilger Circle changes radius dynamically, and thus one interpretation of (5) is that system eigenvalues must reside in the circle "most" of the time, on average. However, as with continuous-time systems, when the system matrix is time varying, simple eigenvalue placement is not sufficient to deduce any conclusions about the overall system stability. For this reason the following Lyapunov analysis is introduced. Take Lyapunov functional

$$V = x^T P x \tag{7}$$

where P solves the equation

$$\mathcal{A}^{*T}P + P\mathcal{A}^* + \mu^*\mathcal{A}^{*T}P\mathcal{A}^* = -I \tag{8}$$

with $\mathcal{A}^* := \mathcal{A}(\mu^*)$ as per the definition in (3). Graininess μ^* is the nominal (or expected) constant sample period, and is selected small enough to guarantee system stability, i.e. the eigenvalues of $\mathcal{A}(\mu^*)$ are strictly within the Hilger Circle. Equation (8) is termed the generalized Lyapunov equation. DaCunha [3] proves that a unique solution P exists given that $\mathcal{A}^*$ has eigenvalues in the Hilger Circle.

When the fixed μ^* is replaced by the time-varying $\mu(t)$, equation (8) then becomes

$$\begin{aligned}&\mathcal{A}^T P + P\mathcal{A} + \mu\mathcal{A}^T P\mathcal{A} \\ =\ & -I + Q(\mu(t))\end{aligned} \tag{9}$$

where the explicit time-dependence of $\mathcal{A}$ and μ is dropped for readability. Note that $Q(\mu(t))$ consists of

$$\begin{aligned}Q(\mu(t)) &= (\mathcal{A}-\mathcal{A}^*)^T P + P(\mathcal{A}-\mathcal{A}^*) \\ &\quad -\mu^*\mathcal{A}^{*T}P\mathcal{A}^* + \mu\mathcal{A}^T P\mathcal{A},\end{aligned} \tag{10}$$

with $Q(\mu^*) = 0$. Delta-differentiating V using the time scale product rule gives

$$\begin{aligned}V^\Delta &= x^{\Delta T}Px^\sigma + x^T P x^\Delta \\ &= x^T[\mathcal{A}^T P + P\mathcal{A} + \mu\mathcal{A}^T P\mathcal{A}]x \\ &= x^T[-I + Q]x \\ &\le \frac{-1 + \lambda_{\max}(Q)}{\lambda_{\min}(P)} V := \eta(\mu)V.\end{aligned} \tag{11}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ and maximum and minimum eigenvalues. Examining the formulation of $\eta(\mu)$, it is clear that $\eta < 0$ in a neighborhood of μ^*, suggesting that the system will remain stable over any range of sampling periods within that neighborhood. However, it is actually not necessary to restrict μ to the region $\eta(\mu) < 0$ all of the time. To see this, a theorem in [4] shows that, if $\eta \in \mathcal{S}_{\mathbb{C}}(\mathbb{T})$, then there exists a positive constant $K \ge 1$ such that

$$V \le Ke^{-\alpha t}, \tag{12}$$

and therefore the state $x(t)$ is exponentially bounded as well. Herein lies the power of time scale analysis: the set $\mathcal{S}_{\mathbb{C}}(\mathbb{T})$ clearly allows for $\eta > 0$ as long as η does not remain positive for too long. In particular, one or even a string of "long" sample periods will not affect the overall system stability as long as $H_k(t) < 1$ from equation (5). The choice of window length in (5) is associated with system performance. If $k = 1$, then the system is required to be "pointwise stable", i.e. $\eta(\mu(t)) < 0$ at every $t \in \mathbb{T}$. This corresponds to the best possible system performance on a given

time scale. On the other hand, if $k > 1$, then performance may degrade somewhat but exponential stability is maintained as long as the time scale does not have so many long sample periods that $H_k(t) \geq 1$. (Note that $H_k < 1$ is a sufficient but not necessary condition for stability; there may exist some $k' > k$ for which $H_k \geq 1$ but $H_{k'} < 1$. This would imply that the system may require a longer period of time to stabilize.)

3 Discussion

Perhaps the most important (and common) metric for real-time determinism is interrupt latency. As mentioned earlier, real-time operating systems frequently juggle many tasks, including file access, network access and multi-thread scheduling. Typically, internal or external timers generate periodic interrupts signaling when it is time to sample sensor values, perform computations, etc. Interrupt latency is the amount of time from the point at which the interrupt is generated to the point at which the computer begins performing the task associated with that interrupt. Interrupt latency depends on a host of factors including the task priority, context switching time, the presence of other interrupt routines, etc. If a control network is involved, communication latency – the time required to communicate with sensors and/or actuators – also impacts overall system determinism. For the purposes of this discussion, the term "latency" will encompass both of these types of latencies, as well as any other delay that prevents the controller from executing a given task at some desired constant sampling rate.

Since the factors that impact latency may be unique and specific to a given platform, it is difficult to give a reliable statistical distribution for timing delays. However, as with many statistics that derive from temporal measurements, latency h may reasonably be viewed as a Poisson process with an associated exponential density function,

$$f(h) = \beta e^{-\beta h}, \qquad h \geq 0. \tag{13}$$

(This corresponds roughly with the author's experience.) Here, we assume that once an interrupt has been serviced, the next interrupt is set to occur in μ^* seconds, so that $\mu(t) = \mu^* + h(t)$. Consequently, very large values of β yield timescales "near" $\mu^*\mathbb{Z}$, a set of uniformly spaced points. A more sophisticated time scale model would assume that interrupts are generated uniformly, whether or not the previous interrupt was serviced. This is more realistic, but more difficult to model.

Let the system parameters be

$$A = \begin{bmatrix} 0 & 1 \\ -4 & 1 \end{bmatrix}; \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \qquad K = [0, -5]. \tag{14}$$

The maximum control period $\mu_{\max}$ that leaves the eigenvalues of $\mathcal{A}$ within the Hilger Circle is approximately $\mu_{\max} = 0.215$. In effect, $\mu_{\max}$ is the absolute upper limit that will still allow the system to be stabilized using uniform sampling. In this example the nominal sampling period is chosen to be $\mu^* = 0.1$, so that the eigenvalues of $\mathcal{A}^*$ are $\{-1.67, -10.04\}$, well within the Hilger circle. As expected, the system step response is well behaved in figure 1.

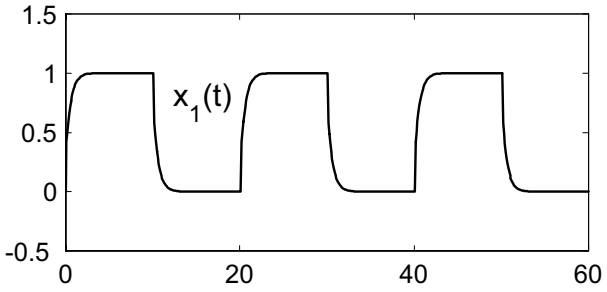

Figure 1: State variable $x_1(t)$ versus time under constant graininess $\mu^* = 1$. ($\beta = \infty$.)

Next, we examine a time scale with random latencies introduced ($\beta = 12$). As seen in figure 2, state x_1 exhibits slightly degraded performance with respect to the constant graininess case. However, the sampling periods are quite variable (figures 2 and 3), with a few at or above 5 times the intended duration. In fact, 76 of 332 control periods exceed $\mu_{\max}$, a level of determinism that would usually dismiss the simulated controller as a viable option for this system. In figure 4, we see that, with a window of $k = 1$, monitoring function $H_k(t)$ shows that the system is frequently "instantaneously unstable," i.e. $\eta > 0$ in equation (11). However, a window of $k = 20$ shows that the system is still stable in the time scale sense.

It may reasonably be argued that relaxing the hard deterministic timing constraints on a given real-time system still represents a risk. After all, in the example above it is probabilistically possible that another timescale will have a lengthy string of overly long sample periods, and after which $H_{20}(t) > 1$. Practitioners will recognize, however, that this example is extreme. Furthermore, in many cases it may be possible to exert some – if not complete – influence over the system timing. For example, in a distributed control network communication latency is partially a function of packet priority, which can be dynamically varied to prevent extremely long waits (a classic scheduling problem). Also, the risk may be acceptable occasionally in view of the possible savings in cost and/or complexity that a less deterministic platform might offer.

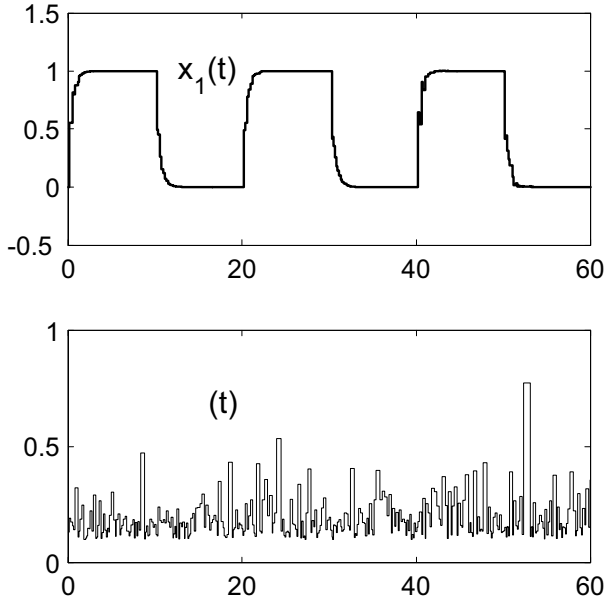

Figure 2: System step response on a time scale with a highly variable latency ($\beta = 12$).

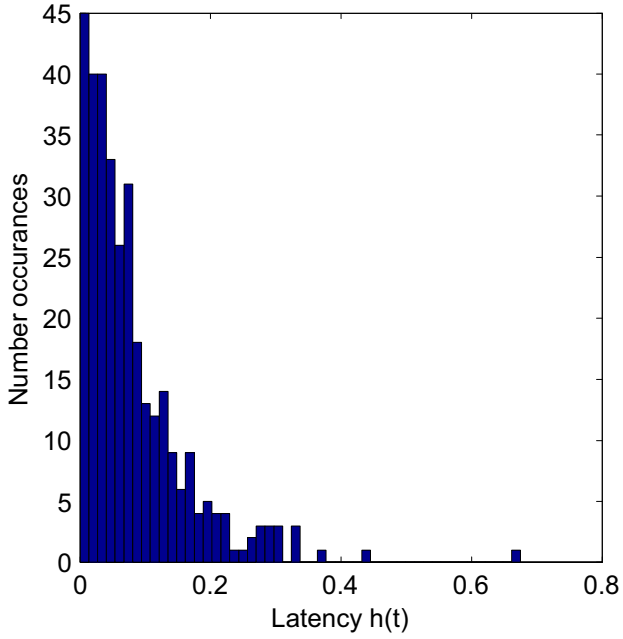

Figure 3: A histogram of timing latency for $\mu(t)$, showing the expected exponential distribution.

4 Conclusions

In summary, the paper discusses the effect of variable timing latencies in real time controllers. After developing a time scale system model for arbitrary discrete time domains, the discussion illustrates how an example system can perform reasonably well under a wide range of random timing latencies. The analysis supports the hypothesis that real-time control platforms may not always have to be as deterministic as often thought. Looser constraints on system determinism can lead to simpler and less expensive computation alternatives for real-time systems; these benefits could be significant given that a considerable portion of the cost and development time for complex automation processes is invested in the controller, software, and communication pathways.

Given the importance of reliable real-time performance in the broad fields of robotics and automation, it seems natural to explore the effects of timing variability. However, only recently have mathematical tools like time scale theory matured enough to support a thorough analysis of the behavior of dynamic systems on variable arbitrary time domains. Future contributions in this area should focus on the application of time scale theory to the types of nonlinear systems of interest to automation and robotics engineers. While this work focuses on the asymptotic stability of a variable-timing controller, it may be argued that determinism is in fact more relevant to a system's transient performance. Furthermore, the analysis here neglects other timing issues such as phase lags between the availability of sensor readings.

This work is supported by a Baylor University Research Committee grant and NSF grant CNS-0410685.

5 Appendix

A thorough introduction to dynamic equations on time scales is beyond the scope of this appendix; however, Bohner and Peterson [2] have written an excellent overview of the subject. In short, the theory springs from the doctoral dissertation of S. Hilger in 1988. Starting the early 1990's, the theory began to grow until in 2001 a complete treatise on the subject

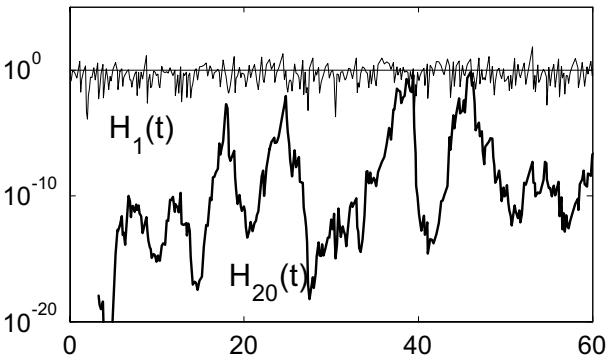

Figure 4: Plots of $H_k(t)$ for $k = 1$ and $k = 20$. Note that $H_{20}(t)$ never exceeds 1.

appeared [2].

A *time scale* $\mathbb{T}$ in this context is defined as an arbitrary non-empty closed subset of the real numbers. Thus time scales can be any of the usual integer subsets (e.g. $\mathbb{Z}$ or $\mathbb{N}$), the entire real line ($\mathbb{R}$) or any combination of discrete points unioned with continuous intervals. The bulk of engineering systems theory to date rests on two time scales, $\mathbb{R}$ and $\mathbb{Z}$ (or more generally $h\mathbb{Z}$, meaning discrete points separated by distance h). However, as this paper illustrates, there are occasional instances when necessity or convenience dictates the use of an alternate time scale. The question of how to how to approach the study of *dynamic systems on time scales* then becomes relevant, and in fact the majority of research on time scales so far has focused on expanding and generalizing the vast suite of tools available to the differential and difference equation theorist.

The paper makes use of a few essential definitions and theorems from this body of work, which are discussed now.

Definition 1 *The* ***forward jump operator*** $\sigma(t) : \mathbb{T} \to \mathbb{T}$ *and the* ***backward jump operator*** $\rho(t) : \mathbb{T} \to \mathbb{T}$ *are given by*

$$\sigma(t) = \inf_{s \in \mathbb{T}} \{s > t\}, \quad \rho(t) = \sup_{s \in \mathbb{T}} \{s < t\}. \tag{15}$$

The ***graininess function*** $\mu(t) : \mathbb{T} \to [0, \infty)$ *is given by*

$$\mu(t) = \sigma(t) - t. \tag{16}$$

Evidently, since the forward jump operator returns the next point in the time scale, the graininess can be visualized as the step size. Note that, for a closed interval of $\mathbb{R}$, $\sigma(t) = t$ except at the rightmost point, and therefore $\mu(t) = 0$ except at the rightmost point. Thus it becomes clear that time scales consist of collections of two types of elements, *scattered points* (i.e. points where $\rho(t) \neq t$ and $\sigma(t) \neq t$) and *dense points* (i.e. points where $\rho(t) = t$ or $\sigma(t) = t$.) At the endpoints of continuous intervals, we have left- and right-dense points as well. Points that are right- and left-scattered are termed *isolated*. If a function is evaluated at a forward or backward jump point, e.g. $f(\sigma(t))$, it is often denoted $f^{\sigma}(t)$. Compositions of forward or backward jumps are written $\sigma^n(t)$ or $\rho^n(t)$, respectively.

Definition 2 *For some function $f : \mathbb{T} \to \mathbb{R}$, the* ***delta derivative*** *of $f(t)$, designated $f^{\Delta}(t)$, is the number (when it exists) with the property that there is some neighborhood U of t where*

$$\begin{aligned} \left|[f(\sigma(t)) - f(s)] - f^{\Delta}(t)[\sigma(t) - s]\right| &\leq \epsilon \left|\sigma(t) - s\right| \\ \forall s &\in U, \end{aligned} \tag{17}$$

and $\epsilon > 0$.

Function f is termed *delta differentiable* provided the delta derivative f^{Δ} exists for all $t \in \mathbb{T} - \{\max(\mathbb{T})\} := \mathbb{T}^{\kappa}$, i.e. the timescale minus its rightmost point if that point exists. For simplicity in this paper we omit the $\mathbb{T}^{\kappa}$ notation because of the convention that, if the time scale does not have a maximum, $\mathbb{T}^{\kappa} = \mathbb{T}$. Not surprisingly, the condition for existence of the delta derivative is simply that f be continuous over all closed, continuous intervals, if there are any (i.e. all subsets of $\mathbb{R}$). If this is the case, the delta derivative is well defined by the equality

$$f^{\Delta} = \frac{f(\sigma(t)) - f(t)}{\mu(t)}. \tag{18}$$

Needless to say, the generalizations of the usual rules of differentiation are not always as simple as their continuous cousins (e.g. $[x^3(t)]^{\Delta} \neq 3x^2(t)$). There is, however, a straightforward product rule for differentiation,

$$(fg)^{\Delta} = f^{\Delta}g + f^{\sigma}g^{\Delta} = f^{\Delta}g^{\sigma} + fg^{\Delta}. \tag{19}$$

This rule easily extends to higher order multiples as well [2].

Of course, along with differentiation one would like to have integration. For this, some mild technical conditions are required [2], including that f be *regulated*, meaning that its right- and left-sided limits exist at any right- and left-dense points in $\mathbb{T}$.

Theorem 3 *Let f be regulated. Then there exists a function F and region of differentiation D such that*

$$F^{\Delta}(t) = f(t), \quad t \in D \tag{20}$$

Definition 4 *A function $F : \mathbb{T} \to \mathbb{R}$ is called an* ***antiderivative*** *of f provided*

$$F^{\Delta}(t) = f(t) \quad \forall t \in \mathbb{T}^{\kappa}. \tag{21}$$

Theorem 5 *Every right-dense continuous function has an antiderivative. If $t_0 \in \mathbb{T}$, then*

$$F(t) = \int_{t_0}^{t} f(\tau)\Delta\tau \quad t \in \mathbb{T}. \tag{22}$$

As one would hope, the theorems above reveal that, in the continuous case $\mathbb{T} = \mathbb{R}$, delta antiderivatives and integrals are the usual antiderivatives and definite integrals from standard calculus. When $\mathbb{T} = \mathbb{Z}$, these quantities correspond to indefinite and definite sums often seen in the study of difference equations. Without further exposition, the usual properties of integrals hold as well, including linearity and homogeneity. However, as with derivatives, the usual integration "rules of thumb" do not hold. These definitions lead to a foundational theorem of time scale calculus, which states:

Theorem 6 *If f is right-dense continuous, then*

$$\int_t^{\sigma(t)} f(\tau) = \mu(t) f(t). \tag{23}$$

This theorem, along with linearity of the time scale integral, is what equates the integral to a sum in the case that $\mathbb{T}$ consists only of isolated points.

From the definitions above, the next obvious step is to investigate linear and the non-linear time scale differential equations, e.g. systems of the form $x^{\Delta\Delta}(t) + a(t)x^{\Delta}(t) + b(t)x(t) = f(t)$ and beyond. Since the time scale itself is often allowed to be arbitrary (or occasionally must adhere to mild assumptions), the theoretical foundations that underpin the study of dynamic equations on time scales are extremely broad. The types of time scales in this paper are relatively "tame" in comparison to the variety that are possible, but nevertheless, time scale theory provides a rigorous and holistic technique by which to study non-uniform sampling problems with relative ease.

References

[1] Audsley, N.; et. al; "Applying new scheduling theory to static priority pre-emptive scheduling," J. Software Engineering, Sept. 1993, pp. 284-292.

[2] Bohner, Martin; Peterson, Allan; *Dynamic Equations on Time Scales*, Birkhauser Boston, 2001.

[3] Dacunha, Jeffrey; "Stability for time varying linear dynamic systems on time scales," J. Computational and Applied Mathematics (in press), available online Sept. 2004

[4] Gravagne, Ian; Davis, John; Dacunha, Jeffrey; "High-Gain Adaptive Control on Time Scales," submitted, Systems and Control Letters

[5] Gravagne, Ian; Davis, John; Dacunha, Jeffrey; "Bandwidth Reduction for Controller Area Networks Using Adaptive Sampling," Proc. IEEE Int. Conf. Robotics and Automation, New Orleans, LA, April 2004, pp. 5250-5255.

[6] Potzsche, C.; Siegmund, S.; Wirth, F.; "A spectral characterization of exponential stability for linear time-invariant systems on time scales," J. Discrete and Continuous Dynamical Systems, v.9(5), Sept. 2003, pp. 1223-1241.

[7] Tindell, Ken; Burns, Alan; "Guaranteeing message latencies on control area network (CAN)," Proc. 1st International CAN Conference, 1994.

[8] Wargui, M.; Tadjine, M.; Rachid, A.; "A scheduling approach for decentralized mobile robot control system," IEEE Int'l Conf. Intelligent Robots and Systems, v.2, 1997, pp. 1138-1143.

[9] Wargui, M.; Rachid, A.; "Application of controller area network to mobile robots," Proc. Mediterranean Electromechanical Conf., v.1, 1996, pp. 205-207.

[10] Wiesspeiner, W.; Windischbacher, E.; "Distributed intelligence to control a stair-climbing wheelchair," Proc. IEEE Int. Conf. Engineering in Medicine and Biology, v.17(2), 1995, pp. 1173-1174.

[11] Zuberi, Khawar; Shin, Kang; "Scheduling messages on controller area network for real-time CIM applications," IEEE Trans. Robotics and Automation, v.13(2), April 1997, pp. 310-314.