ITM Web of Conferences **3**, 02001 (2014)
DOI: 10.1051/itmconf/20140302001

An n–th Order Generalization of the Activity Measure for Continuous Systems

Rodrigo Castro[1,2,a] and Ernesto Kofman[3,b]

[1] *Department of Computer Science, FCEN, Universidad de Buenos Aires and CONICET, Argentina.*
[2] *Department of Environmental Systems Sciences, ETH Zurich, Switzerland.*
[3] *Department of Control, FCEIA, Universidad Nacional de Rosario and CIFASIS-CONICET, Argentina.*

Abstract. In this work we present an analytical expression that generalizes the definition of activity measure in continuous time signals. We define the activity of order n and show that it allows to estimate the number of sections of polinomials up to order n that are needed to represent that signal with certain accuracy. We apply this concept to obtain a lower bound for the number of steps performed by quantization–based integration algorithms in the simulation of ordinary differential equations. We performed a practical analysis over a first order example system, computing the activity of order n and comparing it with the number of steps required integration methods of different orders. We corroborated the theoretical predictions, which indicate that the activity measure can be used as a reference for assessing the suitability of different algorithms depending on how close they perform in comparison with the theoretical lower bound. Finally, a discussion is provided which indicates that further research is needed in order to test the results presented in this work in the context of stiff systems.

1 Introduction

The concept of activity associated to a signal was introduced by Zeigler and Jammalamadaka [1] in order to measure the rate of change of a signal, which in turn allows to estimate the number of steps performed by quantization–based integration methods in the simulation of continuous time systems [2].

The original definition of activity takes only into account the change in the signal value. Thus, for a monotonically increasing or decreasing signal, the activity only measures the distance between the final and the initial value, without using at all the information about how it goes from the initial to the final value. Using this formulation, a signal that growth with a straight ramp contains the same activity than a signal that grows with more complex functions of time.

When the signals are not monotonic, the activity can be computed as the sum of the activity of the different monotonic sections.

In either case, the formal definition of the activity for a signal $x_i(t)$ between an initial time t_0 and a final time t_f is given by

[a] e-mail: rcastro@dc.uba.ar
[b] e-mail: kofman@cifasis-conicet.gov.ar

$$A_{x_i(t_0,t_f)} \triangleq \int_{t_0}^{t_f} \left| \frac{dx_i(\tau)}{d\tau} \right| \cdot d\tau \tag{1}$$

When a continuous time signal $x_i(t)$ is the input of a *zero–order* quantization function, the corresponding output trajectory $q_i(t)$ results piecewise constant as shown in Figure 1.

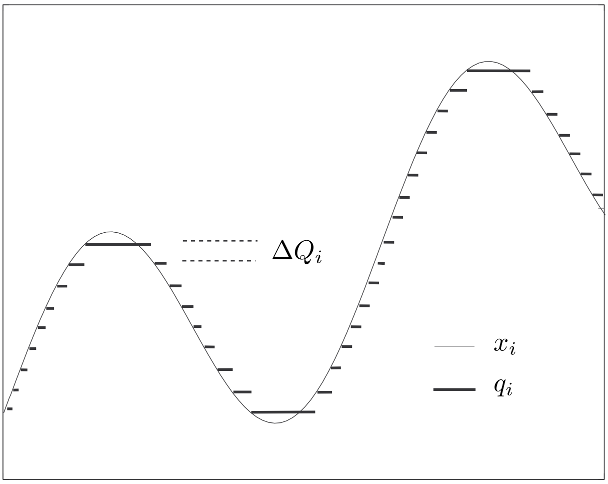

Figure 1. Zero order quantization.

The number of discontinuities of the output trajectory $q_i(t)$ is closely related to the activity of the input $x_i(t)$. Notice that for each interval of time at which $x_i(t)$ is monotonic, the number of quantum crossings is about $A_j/\Delta Q_i$, where A_j is the amplitude (i.e., the activity) of the segment and ΔQ_i is the quantum size.

Thus, the total number of discontinuities in $q_i(t)$ can be directly computed as

$$k \approx \frac{A_{x_i(t_0,t_f)}}{\Delta Q_i} \tag{2}$$

Zero–order quantization functions like that of Fig.1 are used in some Quantized State Systems (QSS) numerical integration algorithms, such as the first order accurate QSS1 method [3], the Backward QSS (BQSS) method [4], and the first order accurate Linearly Implicit QSS (LIQSS1) algorithm [5].

Thus, Eqs.(1)–(2) can be used to establish a lower bound for the number of steps needed by those methods to simulate a given system with a given quantum. That way, the concept of activity can predict the minimal computational costs required to simulate a system with a given accuracy.

However, there exist higher order quantization functions that are also used in QSS methods of higher order. For instance, the QSS2 method [6] uses *first order* quantization functions that produce piecewise linear output trajectories as shown in Fig.2.

In these cases, nor Eq.(1) neither Eq.(2) provide any help to estimate the number of discontinuities in the output trajectory.

Firstly, as it was already mentioned, the activity definition of Eq.(1) obtains the same result for a straight ramp than for a more complex increasing signal provided that both signals have the same

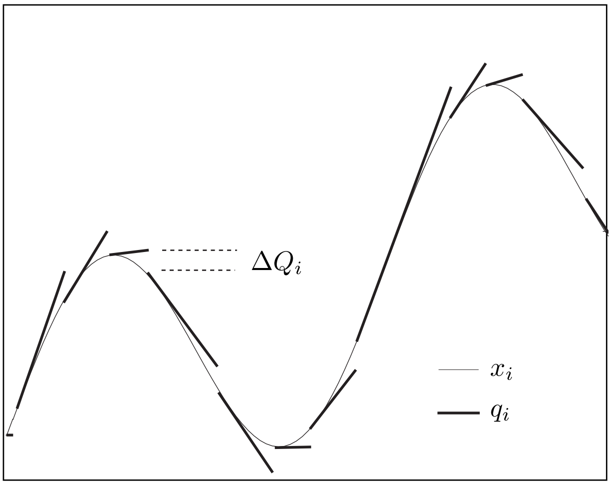

Figure 2. First order quantization.

amplitude. However, a straight ramp can be represented by a single section of a piecewise linear trajectory while a more complex signal, depending on the quantum, would require more segments. Evidently, this difference is not captured by Eq.(1)

Secondly, it is known that in second order accurate methods like QSS2 the number of steps varies with the square root of the quantum ΔQ_i [5, 6]. However, Equation (2) shows a linear dependence, which is clearly wrong.

These facts motivated the need of generalizing the concept of activity so that it can be still applied in presence of higher order quantization.

In this work we study and develop the idea of *activity of order n*.

The paper is organized as follows. In Section 2 we provide a review of quantization schemes up to order three, and synthesize how they are used in algorithms for quantization–based simulation of continuous systems. In Section 3 we first derive the expression for *n*–th order quantization, and based on that, we introduce the definition of activity of order *n*. Then, in Section 4 we apply the new definitions in a first order (non–stiff) system. We then analyse the correlation between theoretical and practical results obtained through simulation, including a discussion about stiff systems. Finally, in Section 5 we present the conclusions and provide hints about follow up steps stemming from the concepts introduced in this work.

2 Background

In this section we review a particular family of quantization functions which are used in the context of quantization–based integration of ordinary differential equations (ODEs).

These quantization functions approximate a continuous time input signal $x_i(t)$ by a piecewise polynomial output signal $q_i(t)$, so that they do not differ more than the quantum ΔQ_i. This is, they ensure that.

$$|x_i(t) - q_i(t)| \leq \Delta Q_i \tag{3}$$

2.1 Zero–order quantization

In zero–order quantization [3, 7] the approximating polynomial segments are of order zero, i.e., the quantized signal $q_i(t)$ is piecewise constant.

Formally, given an input signal $x_i(t)$ and a piecewise constant output signal $q_i(t)$, we say that they are related by a *zero–order quantization function* Q_0 with quantum ΔQ_i if they satisfy

$$q_i(t) = q_i(t_j) \quad \text{for } t_j \leq t < t_{j+1} \tag{4a}$$

with the sequence t_j defined by

$$t_{j+1} = \min_{t>t_j} t \text{ subject to } |q_i(t_j) - x_i(t)| = \Delta Q_i \tag{4b}$$

Notice that $q_i(t)$ follows a piecewise constant trajectory that only changes its value when the difference between $q_i(t)$ and $x_i(t)$ becomes equal to the quantum. After each recalculation of the quantized variable it results that $q_i(t) = x_i(t)$. This behavior is depicted in Figure 1.

One consequence of this approach is that a regular grid of evenly spaced *quantization thresholds* can be imagined superimposed to the input and output trajectories offering an intuitive visual perception of the quantization process: new values of $q_i(t)$ are produced as $x_i(t)$ hits the thresholds that verify $|q_i(t_j) - x_i(t)| = \Delta Q_i$.

Unfortunately, as we shall see shortly after, this grid–oriented hint is only possible in the zero order case; such an evenly spaced set of adjacent thresholds will lack any meaning in higher order schemes, starting already with the first–order quantization case.

2.2 First and second order quantization

The same idea presented for zero–order quantization is followed in first–order quantization [6], but this time around resorting to piecewise *linear* segments for constructing $q_i(t)$ rather than piecewise constant as in the preceding case.

Formally, given an input signal $x_i(t)$ and a piecewise linear output signal $q_i(t)$, we say that they are related by a *first–order quantization function* Q_1 with quantum ΔQ_i if they satisfy

$$q_i(t) = q_i(t_j) + c_{1,j} \cdot (t - t_j) \quad \text{for } t_j \leq t < t_{j+1} \tag{5a}$$

with the sequence t_j defined by

$$t_{j+1} = \min_{t>t_j} t \text{ subject to } |q_i(t_j) + c_{1,j} \cdot (t - t_j) - x_i(t)| = \Delta Q_i \tag{5b}$$

and $c_{1,j}$ computed as:

$$c_{1,j} = \frac{\mathrm{d}x_i}{\mathrm{d}t}(t_j) \tag{5c}$$

The result of this approach is that $q_i(t)$ follows a piecewise linear trajectory that experiences discontinuities at time instants $t = t_j$ when the difference between $q_i(t_j)$ and $x_i(t_j)$ is equal to que quantum ΔQ_i.

Along the same lines, given an input signal $x_i(t)$ and a piecewise parabolic output signal $q_i(t)$, we say that they are related by a *second–order quantization function* [8] Q_1 with quantum ΔQ_i if they satisfy

$$q_i(t) = q_i(t_j) + c_{1,j} \cdot (t - t_j) + c_{2,j} \cdot (t - t_j)^2 \quad \text{for } t_j \leq t < t_{j+1} \tag{6a}$$

with the sequence t_j defined by

$$t_{j+1} = \min_{t>t_j} t \text{ subject to } |q_i(t_j) + c_{1,j} \cdot (t - t_j) + c_{2,j} \cdot (t - t_j)^2 - x_i(t)| = \Delta Q_i \tag{6b}$$

and $c_{1,j}$, $c_{2,j}$ computed as:

$$c_{1,j} = \frac{\mathrm{d}x_i}{\mathrm{d}t}(t_j); \quad c_{2,j} = \frac{1}{2!}\frac{\mathrm{d}^2 x_i}{\mathrm{d}t^2}(t_j) \tag{6c}$$

which results in $q_i(t)$ following a piecewise parabolic trajectory, changing their polynomial coefficients only at time instants $t = t_j$ when the difference between q_i and x_i becomes equal to que quantum ΔQ_i.

The behaviour of a second order quantization function is depicted in Figure 3.

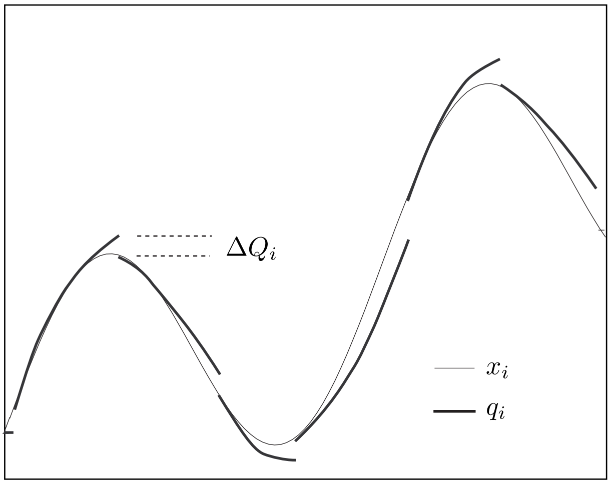

Figure 3. Second order quantization.

2.3 Quantization–Based Integration

Continuous Time Systems are typically represented by ODEs. Except for very simple cases, these ODEs lack of analytical solutions and they must be approximated by numerical integration algorithms in order to be solved. Classic numerical integration algorithms are based on the discretization of the time variable [7].

In recent years, a new class of ordinary differential equation solvers has been developed that replaces the time discretization by the state quantization [7]. These algorithms, based on Zeigler's idea of representing quantized system as DEVS models [9, 10], are called *Quantized State Systems* (QSS) methods.

A QSS numerical solver operates naturally in an asynchronous mode, i.e., the instants t_j belong to the set of positive real numbers and are not confined to any synchronized pattern of time instants.

Each state variable carries its own simulation clock. If the states of a subsystem change very little, the model equations capturing the dynamics of that subsystem will be executed rarely (or equivalently, their activity will be very low).

In the context of QSS–based simulations a dormant model does not slow down the simulation, as its equations will not get executed (i.e., it will experience null activity).

The quantization schemes presented above are those employed within QSS methods that shall be described in the next section.

2.3.1 First Order QSS1 Method

Given the system:

$$\dot{\mathbf{x}}_{\mathbf{a}}(t) = \mathbf{f}(\mathbf{x}_{\mathbf{a}}(t), t) \tag{7}$$

with analytical solution $\mathbf{x}_{\mathbf{a}}(t)$, the first order QSS1 method approximates it by

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{q}(t), t) \tag{8}$$

Here, $\mathbf{q}$ is the *quantized state vector*. Its entries $q_i(t)$ are componentwise related with those of the state vector $x_i(t)$ by a *hysteretic quantization function*, defined as in Eq.(4).

2.3.2 Other First Order QSS Methods

Besides QSS1, the Backward QSS (BQSS) [4], Centered QSS (CQSS) [4] and Linearly Implicit QSS (LIQSS1) [5] perform first order approximations. They differ from QSS1 in the definition of the quantization function given by Eq.(4), however in all these methods the quantized variables $q_i(t)$ follow piecewise constant trajectories.

BQSS and LIQSS1 were conceived to efficiently simulate stiff systems, while CQSS was proposed to simulate marginally stable systems.

2.3.3 Higher Order QSS Methods

The accuracy of the simulation is directly related to the quantum ΔQ_i [6]. Thus, if we want to improve the accuracy by a factor of e.g. 100, the quantum must be reduced 100 times. Then, *any* first order QSS method will perform 100 times more steps. This is a serious limitation of first order schemes, since accurate results require performing lots of steps with the corresponding increment in the computational costs.

To overcome this difficulty, higher order methods were developed like the second order QSS (QSS2) [6] and the third order QSS (QSS3) [8].

The QSS2 method is based on the same principles as QSS1, approximating Eq.(7) by Eq.(8). However, it replaces the zero–order quantization function of Eq.(4) and Figure 1 by a first-order quantization function of Eq.(5) and Figure 2.

Consequently, the quantized state trajectories $q_i(t)$ are piecewise linear and each segment starts with a value and slope equal to those of the corresponding state $x_i(t)$. When both trajectories differ by ΔQ_i, a new segment of $q_i(t)$ starts.

It was shown that in QSS2, the number of steps grows with the square root of the accuracy. Thus, if we want to improve the accuracy by a factor of 100, QSS2 performs only 10 times more steps.

The third order QSS3 method is identical to QSS2, except that it replaces the first order quantization functionby the second order quantization function of Eq.(6) and Figure 3.

A second order quantization function generates an output piecewise parabolic trajectory , whose value, slope and second slope change when the difference between the output and input of the function becomes bigger than the quantum. Each output segment starts with the same value, slope and quadratic slope than the input.

It was shown that in QSS3, the number of steps grows with the cubic root of the accuracy. Thus, if we want to improve the accuracy by a factor of 1000, QSS3 performs only 10 times more steps.

Besides QSS2 and QSS3, there exist *linearly implicit* QSS methods of orders 2 (LIQSS2) and 3 (LIQSS3) which are particularly suitable to simulate stiff systems [5], while having definitions very similar to those of QSS2 and QSS3.

3 Activity of Order n

In this section we generalize the concept of activity so that it can be applied in the context of higher order quantization functions like those used in QSS2 and QSS3 methods.

Before introducing the new definition of activity of order n, we present a definition of n–th order quantization.

3.1 n–th order quantization

The ideas behind first and second order quantization functions can be generalized to define the n–th order quantization function $Q_n(t)$.

Given an input signal $x_i(t)$ and a piecewise polynomial output signal $q_i(t)$, we say that they are related by a *n–th–order quantization function Q_n* with quantum ΔQ_i if they satisfy

$$q_i(t) = q_i(t_j) + c_{1,j} \cdot (t - t_j) + c_{2,j} \cdot (t - t_j)^2 + \cdots + c_{n,t_j}.(t - t_j)^n \text{ for } t_j \leq t < t_{j+1} \tag{9a}$$

with the sequence t_j defined by

$$t_{j+1} = \min_{t>t_j} t \text{ subject to } |q_i(t_j) + c_{1,j} \cdot (t - t_j) + c_{2,j} \cdot (t - t_j)^2 + \cdots + c_{n,t_j}.(t - t_j)^n - x_i(t)| = \Delta Q_i \tag{9b}$$

and the coefficients $c_{m,j}$ computed as:

$$c_{m,j} = \frac{1}{m!} \frac{\mathrm{d}^m x_i}{\mathrm{d}t^m}(t_j) \tag{9c}$$

The latter requires all derivatives of $x_i(t)$ to exist at least up the order of the quantization scheme, i.e., n.

3.2 Activity of Order n

The original definition of activity [1] given by Eq.(1) integrates the rate of change $\left|\frac{\mathrm{d}x_i(t)}{\mathrm{d}t}\right|$ experienced by a continuous time signal $x_i(t)$ in a given interval of time.

When $q_i(t)$ is a piecewise constant approximation of $x_i(t)$, the rate of change $\left|\frac{\mathrm{d}x_i(t)}{\mathrm{d}t}\right|$ is also the rate at which the difference $|q_i(t) - x_i(t)|$ grows while $q_i(t)$ remains constant.

However, if $q_i(t)$ is obtained from a quantization function of order $n - 1$ with $n \geq 2$, the rate at which the difference $|q_i(t) - x_i(t)|$ grows follows a different law:

$$\Delta x_i(t) = x_i(t) - q_i(t) = x_i(t) - \left[x_i(t_j) + \frac{\mathrm{d}x_i(t_j)}{\mathrm{d}t} \cdot (t - t_j) + \cdots + \frac{\mathrm{d}^{n-1} x_i(t_j)}{\mathrm{d}t^{n-1}} \cdot \frac{(t - t_j)^{n-1}}{(n-1)!} \right]$$

where t_j is the time of the last discontinuity of $q_i(t)$.

Replacing $x_i(t)$ by its Taylor series expansion:

$$x_i(t) = x_i(t_j) + \frac{\mathrm{d}x_i(t_j)}{\mathrm{d}t} \cdot (t - t_j) + \cdots + \frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n} \cdot \frac{(t - t_j)^n}{n!} + \cdots$$

it results that

$$\Delta x_i(t) = \frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n} \cdot \frac{(t - t_j)^n}{n!} + \frac{\mathrm{d}^{n+1} x_i(t_j)}{\mathrm{d}t^{n+1}} \cdot \frac{(t - t_j)^{n+1}}{(n + 1)!} + \cdots$$

When the difference $t - t_j$ is small or when the n–th derivative of $x(t)$ is constant (as it happens in QSSn methods) the difference between $q_i(t)$ and $x_i(t)$ results:

$$\Delta x_i(t) \approx \frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n} \cdot \frac{(t - t_j)^n}{n!} \tag{10}$$

After $t = t_j$, the next discontinuity in $q_i(t)$ occurs at $t = t_{j+1}$, where $|\Delta x_i(t)| = \Delta Q_i$. Then, from Eq.(10) it results that

$$\Delta Q_i \approx \left| \frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n} \right| \cdot \frac{(t_{j+1} - t_j)^n}{n!}$$

Dividing the latter by ΔQ_i and computing the $1/n$ power at both sides, it results

$$1 \approx \left| \frac{\frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n}}{n!} \right|^{1/n} \cdot \left(\frac{1}{\Delta Q_i} \right)^{1/n} \cdot (t_{j+1} - t_j)$$

This equation holds for $j = 0, \cdots, k - 1$ in the interval (t_0, t_k). Then, we can compute the summatory for j at both sides:

$$\sum_{j=0}^{k-1} 1 \approx \sum_{j=0}^{k-1} \left| \frac{\frac{\mathrm{d}^n x_i(t_j)}{\mathrm{d}t^n}}{n!} \right|^{1/n} \cdot \left(\frac{1}{\Delta Q_i} \right)^{1/n} \cdot (t_{j+1} - t_j)$$

and approximating the summatory by the integral, we finally obtain

$$k \approx \left(\frac{1}{\Delta Q_i} \right)^{1/n} \int_{t_0}^{t_k} \left| \frac{\frac{d^n x_i(\tau)}{d\tau^n}}{n!} \right|^{1/n} \mathrm{d}\tau$$

which provides an expression for the number of discontinuities in $q_i(t)$ on the interval (t_0, t_k).

From this last expression, it makes sense to define the n–th order activity of the signal $x_i(t)$ on the interval (t_0, t_f) as

$$A^{(n)}_{x_i(t_0,t_f)} \triangleq \int_{t_0}^{t_f} \left| \frac{\frac{d^n x_i(\tau)}{d\tau^n}}{n!} \right|^{1/n} \mathrm{d}\tau \tag{11}$$

In that way, given a continuous time signal $x_i(t)$ we can estimate the *number of discontinuities* for an approximation of order n using a quantum ΔQ_i as:

$$k^{(n)}_{x_i(t_0,t_f)} \approx \frac{A^{(n)}_{x_i(t_0,t_f)}}{(\Delta Q_i)^{1/n}} \tag{12}$$

Notice that when $n = 1$ Eq.(11) coincides with the original definition of activity of Eq.(1) and the formulae for estimating the number of discontinuities given by Equations (12) and (2) become identical.

With Eqs.(11) we also extended to the n–order the concept that the activity measure is a property inherent to a signal, in contrast to the number of discontinuities computed by Eq.(12), wich depends on choices of the quantum size according the required accuracy.

4 Example: A first order linear system

The first order linear system:

$$\dot{x}(t) = a \cdot x(t)$$

has solution $x(t) = x(0) \cdot e^{a \cdot t}$.

The n–th order activity of the solution $x(t)$ is, according to Eq.(11),

$$A^{(n)}_{x(0,t_f)} = n \cdot |1 - e^{a \cdot t_f / n}| \cdot \left| \frac{x_0}{n!} \right|^{1/n} \tag{13}$$

When a is negative and $|a \cdot t_f| >> n$ it results that

$$A^{(n)}_{x(0,t_f)} \approx n \cdot \left| \frac{x(0)}{n!} \right|^{1/n}$$

Notice that the activity in this case is independent on the eigenvalue a.

Using the parameter $a = -1$, initial condition $x(0) = 1$ and a final time $t_f = 5$ the activities of order 1, 2, and 3, according to Eq.(13), result

$$A^{(1)}_{x(0,t_f)} = 0.993262; \quad A^{(2)}_{x(0,t_f)} = 1.298128; \quad A^{(3)}_{x(0,t_f)} = 1.339137 \tag{14}$$

We simulated the system with QSS1, QSS2 and QSS3 methods using in each case quanta from $\Delta Q = 10^{-2}$ down to $\Delta Q = 10^{-6}$ with decrements of one order of magnitude. Then, we compared the number of steps performed by each method with the number of steps predicted by Eq.(12). The results are shown in Table 1.

	QSS1		QSS2		QSS3	
	$k^{(1)} = \frac{A^{(1)}}{\Delta Q}$	Steps	$k^{(2)} = \frac{A^{(2)}}{\sqrt{\Delta Q}}$	Steps	$k^{(3)} = \frac{A^{(3)}}{(\Delta Q)^{1/3}}$	Steps
$\Delta Q = 10^{-2}$	99.3262	100	12.981	13	6.2157	12
$\Delta Q = 10^{-3}$	993.2620	994	41.0504	41	13.3913	15
$\Delta Q = 10^{-4}$	9932.62	9933	129.81	130	28.8508	28
$\Delta Q = 10^{-5}$	99326.205	99327	410.5040	411	62.1572	61
$\Delta Q = 10^{-6}$	993262	983881	1298.1	1298	133.914	133

Table 1. Theoretical and real number of quantum crossings.

4.1 Analysis of the results

The results agree with the theoretical predictions. There are only two cases in which the results do not match closely: the QSS1 simulation with a small quantum and the QSS3 simulation with a large quantum.

In the first case, the steps are very small (there are more than 980000 steps in 5 seconds). As a consequence, the round–off errors become significative.

In the second case, the difference is due to the fact that QSS3 starts with a first order approximation (in the first step it does not have information about the first derivative) and then it follows with a second order one. Only after the third step it really performs a third order approximation.

It is important to recall that QSS methods may introduce some spurious oscillations which provoke additional steps that can not predicted by the activity of the analytical solution. Thus, it must be remarked that the activity measure only offers a lower bound for the number of steps performed by an algorithm of order n.

A very relevant class of systems exist that strongly affect the number of computational steps performed by a numerical method, which cannot be predicted by the expression of activity presented as presented in this work if special methods are not used. This is the case of so–called stiff systems, which are briefly discussed in the next section.

4.2 The case of stiff systems

Stiff systems are a class of dynamical systems where slow and fast dynamics coexist. For stability reasons, stiffness imposes particular difficulties to non–stiff classic numerical solvers. Provided that the step size is adjusted so that the numerical solutions remain stable, they invariably end up performing an excessive number of computation steps for coping with spurious high frequency oscillations which are numerical artifacts introduced by the (non–stiff aware) numerical techniques.

Unfortunately, non–stiff quantization–based algorithms, such as QSS methods, experience similar difficulties. When a stiff system is solved by a QSS method, spurious high frequency oscillations appear in the numerical solution (which are not present at the analytical solution) [4, 5, 7].

Due to these oscillations, the activity of the numerical solution results much higher than the activity of the analytical solution. Thus, the number of steps performed by non–stiff QSS methods loose any relation with the theoretical figure predicted by (12).

However, a special branch of state-quantization based methods has been recently developed that efficiently simulate many stiff systems. Backward and Linearly Implicit QSS methods (BQSS, LIQSS) [4, 5] tend to eliminate the spurious oscillations, yielding a number of practical quantum crossings that, in principle, should match closer theoretical activity figures.

We can reasonably expect that the theoretical estimation provided by the activity of order n can be used as a theoretical lower bound for the number of steps performed by an algorithm of order n.

This lower bound can be compared with the actual number of steps produced by an algorithm. Thus, we could measure how suitable is that algorithm for simulating a given stiff system, not only by comparing raw values against other methods, but also by comparing their relative departures from a common theoretical absolute minimum.

5 Conclusions and next steps

We have presented a generalization of the concept of activity for continuous time signals. While the original definition of activity [1] measures the rate of change of the signal, the new definition of activity of n–th order takes into account the rate of change of its higher derivatives.

By doing so, this new concept allows to estimate the number of steps performed not only by first order quantization–based numerical integration algorithms such as QSS1, but also the number of steps performed by higher order methods.

This fact was analyzed in a simple example, where the number of steps performed by the QSSn algorithm and the theoretical estimations based on the activity of order n agreed in most cases for different orders and accuracy settings.

We remark that the results presented in this work are, in principle, mainly of theoretical value. The exact computation of the activity of order n (including the original case of $n = 1$) requires knowing the analytical solution of the system, which is impossible to obtain except for very simple cases.

However, the new concept formalizes the relationship between activity and quantization–based simulation of continuous systems for higher order algorithms. It also establishes a formal proof about the dependence between computational costs (which depend on the number of steps performed) and the accuracy of a simulation (which depends linearly on the quantum ΔQ_i) for a method of order n.

We are currently working on the following extensions and applications of the concept of n–th order activity:

- We are running experiments on the simulation of stiff systems using both stiff and non–stiff solvers, analyzing the results in light of the new framework of theoretical n–th order activity.
- Another current line of work stemming from the theoretical results herein is to explore how the knowledge of activity measures for each variable in a given system can be exploited to derive optimal model partitions into multiple parallel processing nodes (cores, processors, servers) in order to maximize speedups as compared against a serial (single node) simulation.

References

[1] R. Jammalamadaka, Master's thesis, The University of Arizona (2003)

[2] B.Z. A. Muzy, R. Jammalamadaka, J. Nutaro, Simulation: Transactions of the society of modeling and simulation international **87**, 449 (2011)

[3] E. Kofman, S. Junco, Simulation: Transactions of the Society for Modeling and Simulation International **18**, 123 (2001)

[4] G. Migoni, E. Kofman, F. Cellier, Simulation: Transactions of the Society for Modeling and Simulation International **88**, 387 (2012)

[5] G. Migoni, M. Bortolotto, E. Kofman, F. Cellier, Simulation Modelling Practice and Theory **35**, 118 (2013)

[6] E. Kofman, Simulation: Transactions of the Society for Modeling and Simulation International **78**, 76 (2002)

[7] F.E. Cellier, E. Kofman, *Continuous System Simulation* (Springer, New York, 2006)

[8] E. Kofman, Latin American Applied Research **36**, 101 (2006)

[9] B. Zeigler, J. Lee, *Theory of quantized systems: formal basis for DEVS/HLA distributed simulation environment*, in *SPIE Proceedings* (1998), pp. 49–58

[10] B. Zeigler, T. Kim, H. Praehofer, *Theory of Modeling and Simulation. Second edition* (Academic Press, New York, 2000)