

TITLE ME

AUTHOR ME

March 9, 2013

**Abstract**

Who is writing me? Later anyway...

## **1 Introduction**

Professor Bohannon is writing this ...

**Outline** remove or fix this paragraph later– this is here for our reference now **Do we intend to discuss the exponential or recursive algorithms in this paper?**

- Section 2 will contain algorithm definitions.
- Section 3 will contain:
  - the amplitude and phase response of these algorithms for a large and small number of bins in the partition.
  - an analysis of the computational resources required for the larger versus the smaller number of bins in the partition.

## **2 Algorithm definiitions**

### **2.1 The Average-Shift algorithm: partitioning the history into averaged bins**

To obtain a digital differ-integrator that both is computationally efficient and contains contributions from deep history, we modify the Grunwald integral by partitioning the history into binned intervals in time. Within each bin, the history of the input signal at every time step is represented by its average value. The motivation is to select bins such that time is rescaled— a greater number of points contribute to a bin’s sum far into the past where the weights are weak than near the present time. With this binning algorithm, we store and use only the averages of the bins rather than the full history of the input signal, conserving memory and improving speed.

## 2.2 Modified Grunwald

The Grunwald form of the fractional integral can be written

$$D_t^\alpha f(t) = \lim_{N \rightarrow \infty} \left( \frac{t}{N_t} \right)^{-\alpha} \sum_{j=0}^{N_t-1} w_j f\left(t - \frac{jt}{N_t}\right) \quad (1)$$

where  $f(t)$  is the input signal at time  $t$

$$w_j = \frac{\Gamma(j - \alpha)}{\Gamma(j + 1)\Gamma(-\alpha)} \quad (2)$$

To extend the Grunwald sum of Equation 1, we simply sum the coefficients within each bin. We define

$$W_k = \sum_{bin} w_j \quad (3)$$

where  $w_j$  is summed over the historical time steps within the  $k$ th bin and

$$D_t^\alpha = (\Delta t)^{-\alpha} \sum_{k=0}^{N_b} \bar{W}_k X_k \quad (4)$$

where  $\Delta t$  is the interval between time samples. During start-up, some bins may be empty and one bin may be partially filled. To handle bins that are partially full, we weight the binned Grunwald weights by the ratio of the bin occupation number  $c_k$  to its capacity  $b_k$ ,

$$\bar{W}_k = \frac{c_k W_k}{b_k}. \quad (5)$$

### 2.3 Updating the average history

When the history is updated, the new data element is shifted into the first bin through a weighted average. Since data elements represent time steps, they should be incompressible—when one element is shifted into a bin, another element should be shifted out of that bin if the bin is full. To update the remaining bins, we take the weighted average obtained by adding one sample from the more recent bin to all minus one of the samples in the current bin. During start-up, it will be necessary to consider bins that have some set size, but are not filled to that capacity. In that case, it is the current occupation number of each bin that enters the calculation.

Consider a digitally sampled data stream of points,  $x_j$ . As these data points enter, they could be stored in the history from most recent at  $x_0$  to most distant past at some  $x_{N_t}$ . Then, as a new data point entered,  $x_i \rightarrow x_{i+1}$  and the new element would become  $x_0$ . However, storing the full history of data points presents a large memory and computational burden. To avoid this, we partition the history into  $N_b$  bins of sizes  $b_k$  with the  $k = 0$  bin containing the  $j = 0$  data point after the first time step.

Each time step, the average input signal  $X_k$  is updated through a weighted average, assuming the value of each element of the  $k$ th bin is represented by the average value of the input signal in the  $k$ th bin. If the  $k$ th bin initially contains  $c_k$  elements, adding another element either leaves  $c_k$  unchanged ( $c_k' = c_k = b_k$ ) or increments the number of elements in the bin such that  $c_k' = c_k + 1$  if the bin is not yet at capacity. Either way, the updated average of the value of the  $k$ th bin,  $X_k'$ , is given by

$$X_k' = \frac{c_k' - 1}{c_k'} X_k + \frac{1}{c_k'} X_{k-1} \quad (6)$$

where  $X_k$  is the value of the  $k$ th bin.

**3 Results**

**4 Conclusions**

**References**