An efficient deep memory algorithm for computing fractional order operators

Steven Dorsher and Gary W. Bohannan

June 11, 2013

Abstract

This paper outlines a method to achieve effective bandwidth of five or more decades in the approximation of a fractional order derivative. This constitutes an increase of two decades or more over current algorithms, such the Infinite Impulse Response (IIR) method based on continued fraction expansion, while demanding only slightly more computational steps and processor memory.

1 Introduction

Interest in the application of the fractional calculus has been growing at an ever increasing rate. Of long term and continued interest is in the use of fractional order (FO) operators, such as integrators and differentiators, in motion control applications. [1] While wide bandwidth analog controllers have been successfully demonstrated [2], fractional order analog circuit elements are not generally available and the prototypes that have been demonstrated do not have the ability to be retuned for a specific desired phase. [3]

Unfortunately, the existing techniques for digital approximation of FO operators are limited in bandwidth to on the order of three and a half decades of frequency response while nonlinear effects in motion control systems can span five decades or more. This places a severe constraint on the design and implementation of digital FO controllers, i.e. how to set the sampling frequency to meet the high speed requirements necessitated by the Nyquist sampling rate while at the same time providing enough deep memory to get low offset error. Additionally, the infinite impulse response type of implementation cannot guarantee stability due to the limitations of finite precision arithmetic. See e.g. [4].

Given the current necessity to implement FO controls in digital form, it is desirable to obtain the most efficient algorithm to compute a fractional order operator while maximizing the numerical stability of the algorithm. Efficiency to be measured in both memory utilization and number of computations per time step. This paper outlines a computational method inspired by the Riemann-Liouville integral definition and the Grünwald algorithm. [5] The essential concept is the rescaling of time by successive accumulation of older data into increasing size bins for deeper memory.

Outline

We will first briefly describe the current state of the art in fractional order operator approximation and then describe a novel approach based on successive binning of older data.

- 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 definitions

2.1 Partitioning the Grunwald history into averaged bins

The technology most widely used today in digital fractional order circuitry is the continued fraction expansion (CFE) of the Tustin approximation to s^α. Its benefits are that it was a flat phase response over approximately two and a half decades in phase for a 9th order expansion (10 registers of input signal memory). It would be desirable to find an algorithm with an even broader flat phase frequency bandwidth and with a comparably small amount of memory required. We take the Grunwald algorithm as a starting point. As the signal history retained in the Grunwald sum grows longer, the bandwidth of flat phase response grows broader; however, the memory required also increases with input signal history length. To reduce this memory requirement and retain a long signal history, we propose partitioning the input signal history into bins that are longer further into the past. Since the Grunwald weights decrease rapidly moving further back in time, the older and longer bins will sum together many small contributions, never exceeding the total from the more recent and shorter bins. The presence of short bins at recent times maintains sensitivity to high frequencies, while the inclusion of long bins at past times adds sensitivity to low frequencies that would not usually be present in a Grunwald sum with the same number of terms.

2.1.1 Modified Grunwald

The Grunwald form of the fractional integral can be written

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

where $f(t)$ is the input signal at time t, the jth value of the input signal history is $x_j = f\left(t - \frac{jt}{N_t}\right)$, and the jth Grunwald weight is

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

TO INCLUDE MORE DISTANT HISTORY AT LOW COMPUTATIONAL COST, We modify the Grunwald sum of Equation 1 by partitioning its history into N_b bins. In each bin k, the input signal history x_j is represented by its average value over that bin, X_k.

Since we make the assumption that each value is well represented by its average within a bin, we can define a value for the "bin coefficient" by summing the Grunwald coefficients within that bin.

$$W_k = \sum_{j=p_{k-1}+1}^{p_k} w_j \tag{3}$$

where w_j is summed from the lowest index of the input data history within bin k to the highest index p_k within that bin. There is an additional factor that goes into $\bar{W}$ that will be discussed in Section 2.1.2.

With these definitions, the modified Grunwald differ-integral can be written

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

where Δt is the interval between time samples.

2.1.2 Updating the average history

When a new input data element is read, 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 virtual element should be shifted out of that bin if the bin is full. It shifts into the next bin, and pushes a virtual element out of that one, until a bin which is partially full or empty is reached. To update the average data stored in the bins, we take the weighted average obtained by adding one virtual element from the $(k-1)$th bin to the b_k elements in the kth bin.

During start-up, it will be necessary to consider bins that have some set size b_k, but are not filled to that capacity. In that case, it is the current occupation number c_k of each bin that enters the calculation. If the kth bin initially contains c_k elements, updating the history either leaves c_k ($c_k\prime = c_k = b_k$) or increments the number of elements in the bin such that $c_k\prime = c_k + 1$ if the bin is not yet at capacity. Either way, the updated average of the value of the kth bin, $X_k\prime$, is given by

$$X_k' = \frac{c_k' - 1}{c_k'} X_k + \frac{1}{c_k'} X_{k-1}. \tag{5}$$

During start-up, these partially full bins may also factor into the Grunwald weights. 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,

$$\bar{W}_k = \frac{c_k W_k}{b_k}. \tag{6}$$

3 Results

3.1 Analysis of computational resources

4 Conclusions

References

[1] Ying Luo and Yang-Quan Chen. *Fractional Order Motion Controls*. John Wiley & Sons, Chichester, West Sussex, 2013.

[2] G. Bohannan. Analog fractional order controller in temperature and motor control applications. *Journal of Vibration and Control*, 14(9-10):1487–1498, 2008.

[3] Concepción A. Monje, YangQuan Chen, Blas M. Vinagre, Dingyü Xue, and Vicente Feliu. *Fractional-order Systems and Controls: Fundamentals and Applications,* a monograph in the *Advances in Industrial Control* Series. Springer, Berlin, 2010.

[4] Yang-Quan Chen, Blas M. Vinagre, and Igor Podlubny. Continued fraction expansion approaches to discretizing fractional order derivatives – an expository review. *Nonlinear dynamics*, 38:155–170, 2004.

[5] K. Oldham and J. Spanier. *The Fractional Calculus.* Academic Press, New York, 1974.