**Tustin**

The Tustin transform is an implementation of the difference equation show in figure ---

It makes use of two buffers, xt[] and yt[] which hold the history of input and outputs, respectively, where the 0th index represents the newest sample, and increasing indices indicate samples from further back in time.

The algorithm begins by shifting the past inputs back in the buffer to make room in xt[0], which is initialised to the incoming sample. The filter is then applied using the coefficients calculated earlier and assigned to the start of the output buffer. That value is also assigned to the next element in the buffer for use in the next iteration of the filter, and is finally assigned to the output.

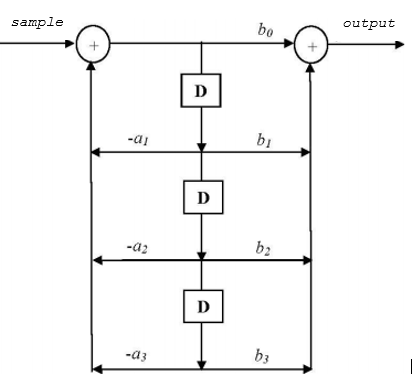


**Direct Form 2**

To implement the direct form 2 filter, we consider each adder shown in figure ----- as two separate variables, left and right.

For every new sample that is received, the existing samples in the buffer are shifted down, and the relevant filter coefficients are applied, the results of which are added to the respective variable: left for the a coefficients, and right for the b coefficients.

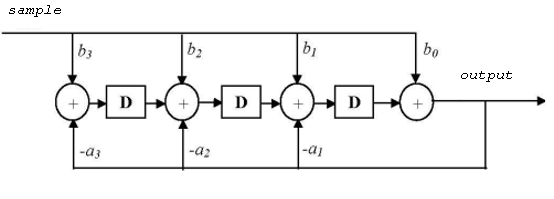
The new sample is then added to the left adder, which is multiplied by b[0] and added to right to create the final value of the output. Lastly, the current value of left is assigned to the first element of the buffer, to be ready for the next cycle.



**Direct Form 2 Transpose**

This form works by summing the chained outputs of previous adders.

The output is initialised to be the sum of the first output and the first b coefficient applied to the new sample, as shown in figure ---. The for loop in the function sets the value of the output of each adder (represented by x[i]) as the sum of the output of the previous adder, and the ith a and b coefficients applied to the new sample. The reason it only runs until BUFSIZE-2 is that the last element, i.e. the last adder (position BUFSIZE-1 in the buffer), has no adder previous to it, so its output is calculated slightly differently: the last b coefficient applied to the new sample, minus the last a coefficient applied to the output.



**Source:** https://bb.imperial.ac.uk/bbcswebdav/pid-591062-dt-content-rid-2714545\_1/courses/DSS-EE3\_19-15\_16/lab5.pdf