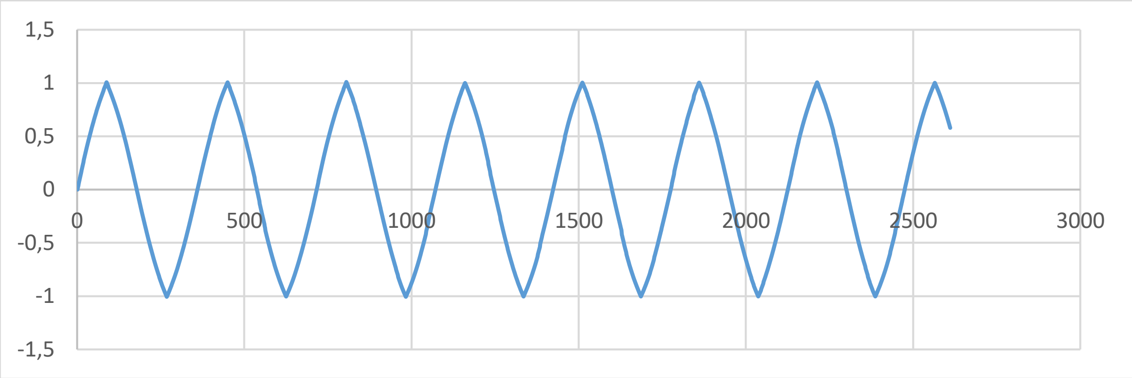
We have started with RF = 1Hz and WT = 1000us, and as so we run two experiments:

We have decreased the WT while the RF was fixed.

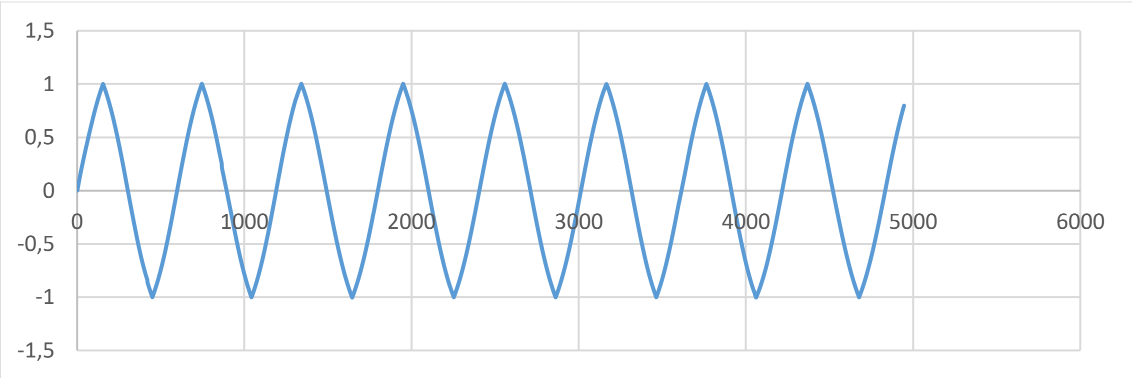
We have increased the RF while WT was fixed.

So with RF = 1:

WT = 1000

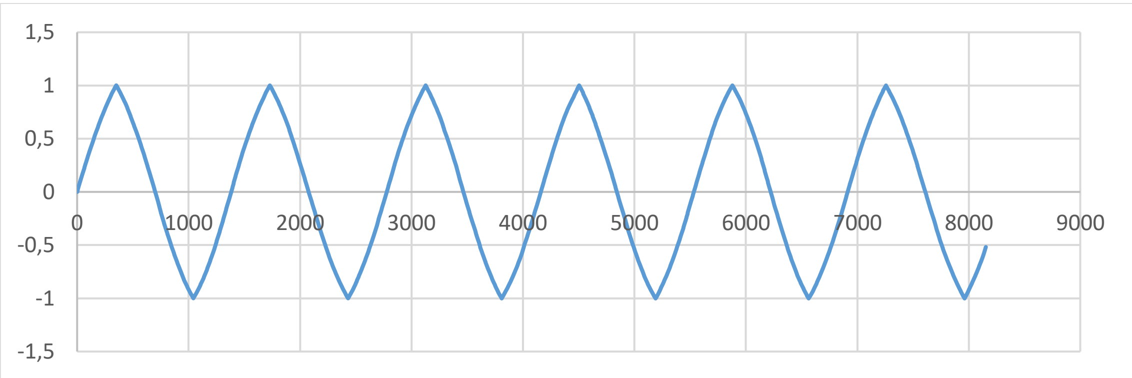


WT = 500

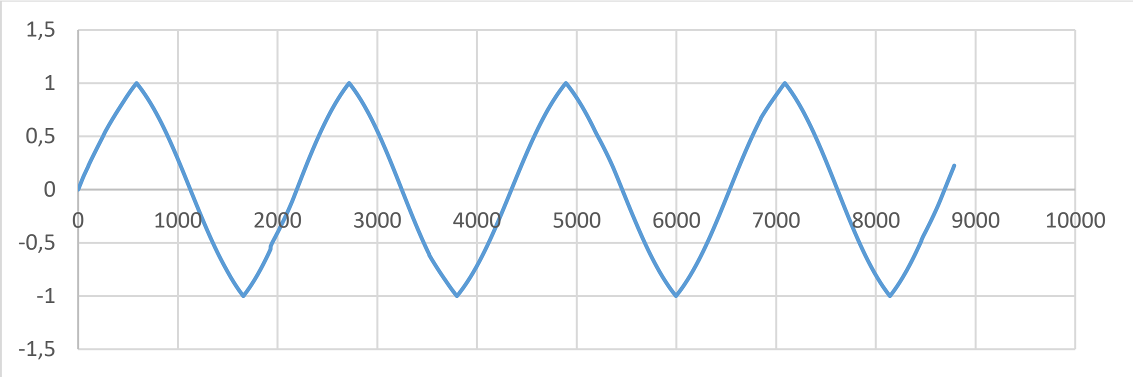


Since this point down, for matter of precision, we try to gather between 8.000 and 10.000 measurements:

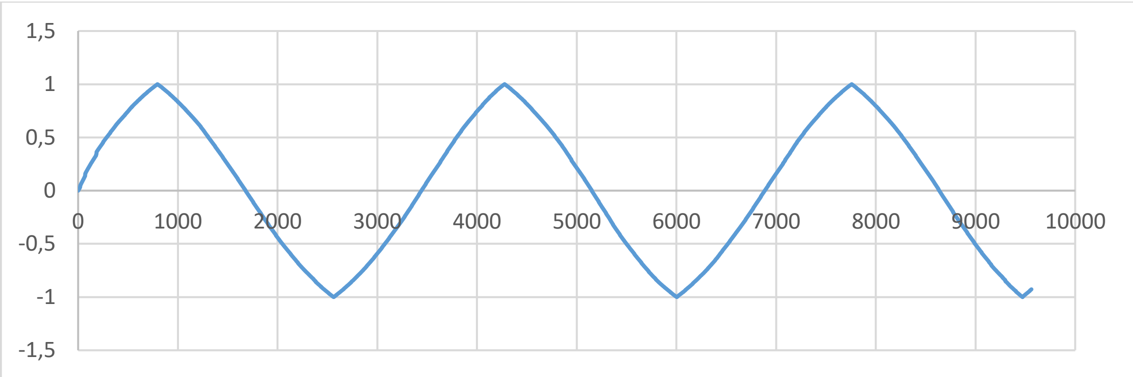
WT = 200



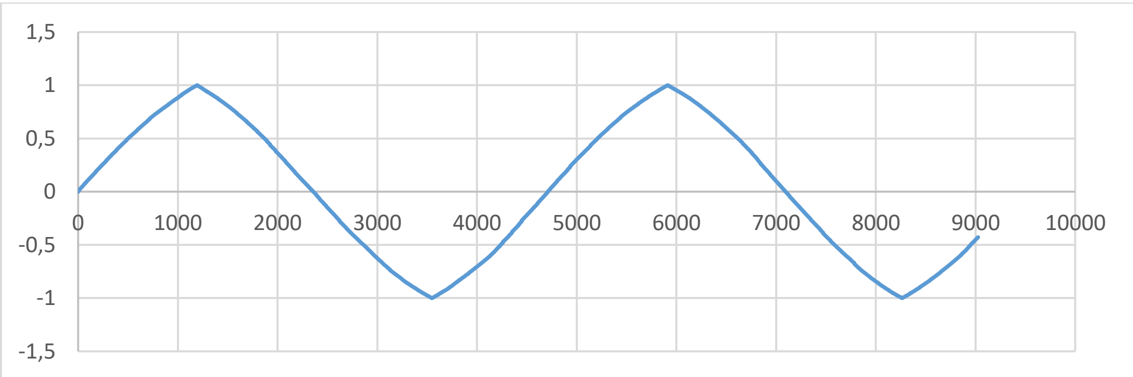
WT = 100



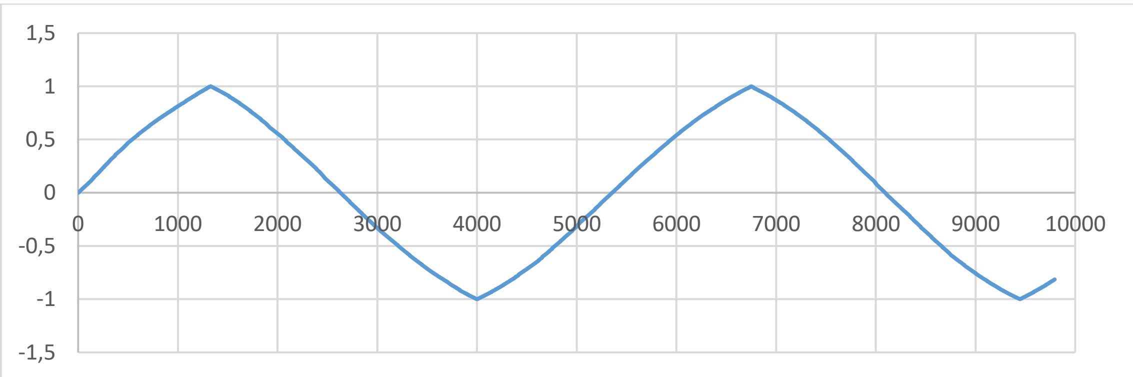
WT = 50



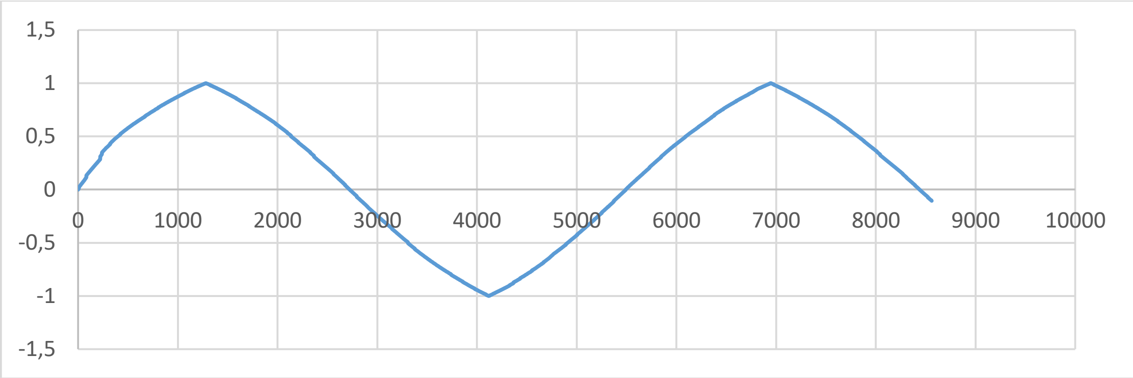
WT = 20



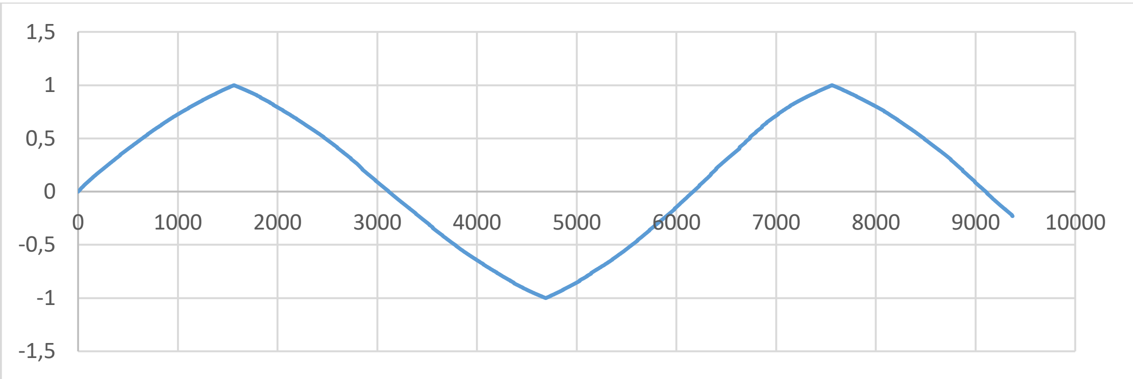
WT = 10



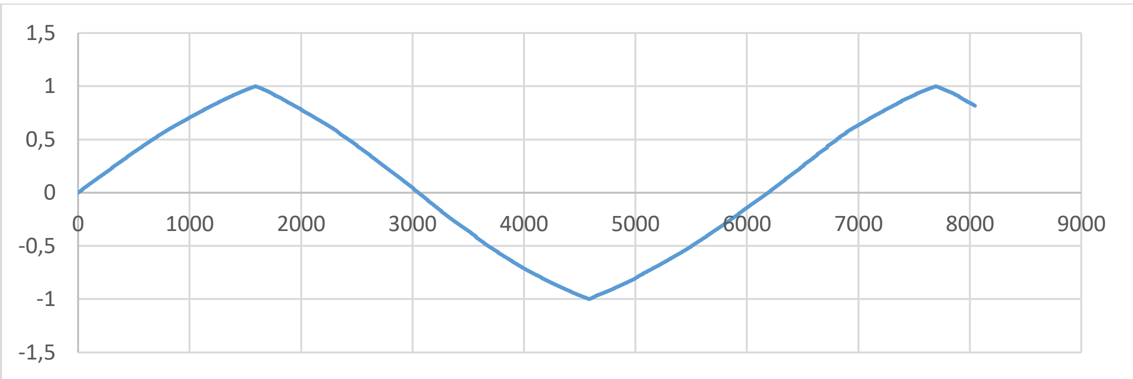
WT = 5



WT = 2



WT = 1

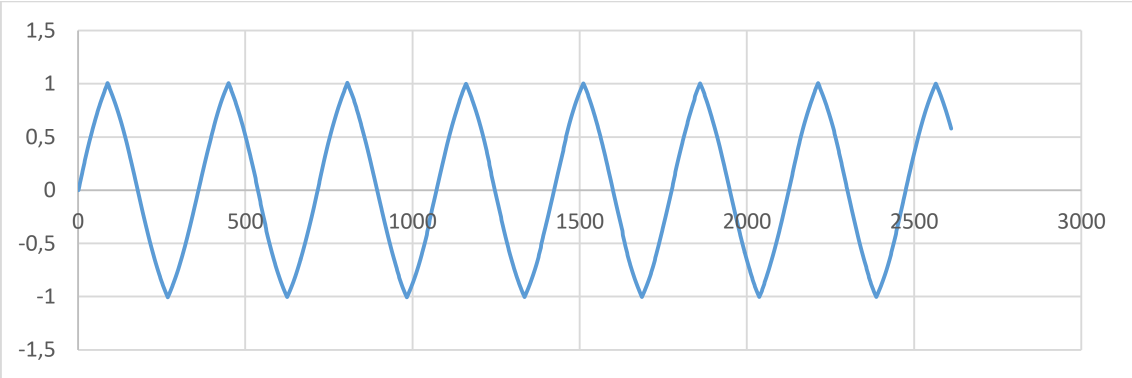


It was impossible for us to print in a file a sine wave with WT = 0. The amount of measurements taken was so huge that made the pc crash when trying to print them in a file.

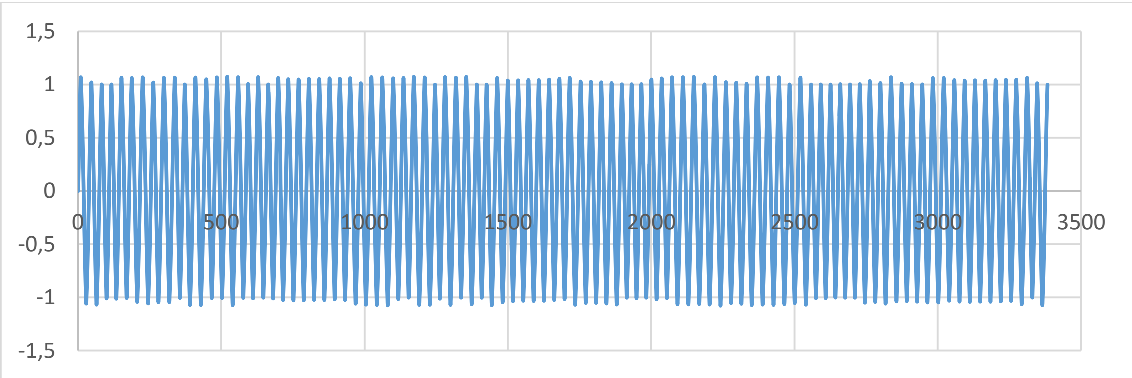
Also problems appeared when WT had huge values, which could make our DT increase to the point our system went to infinite.

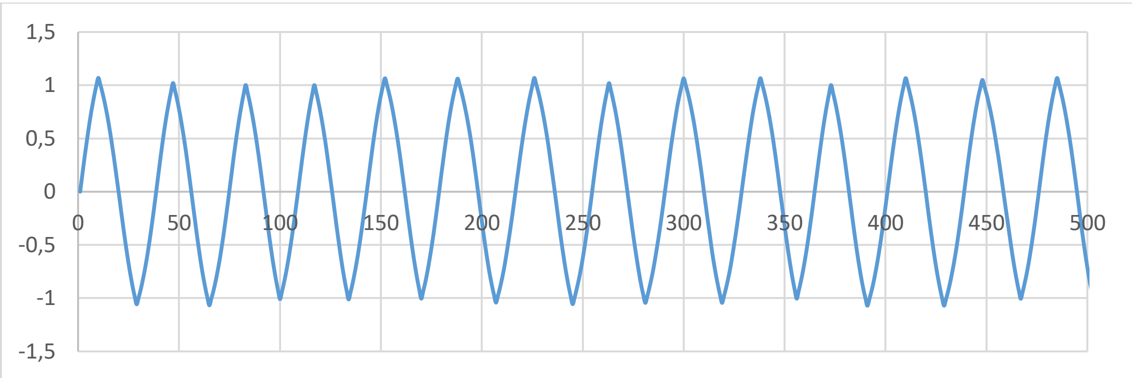
Now with WT = 1000

RF = 1

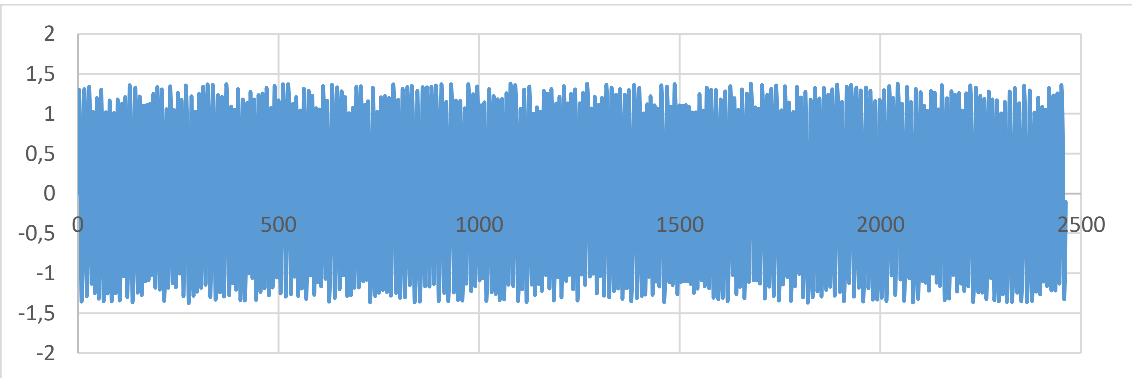


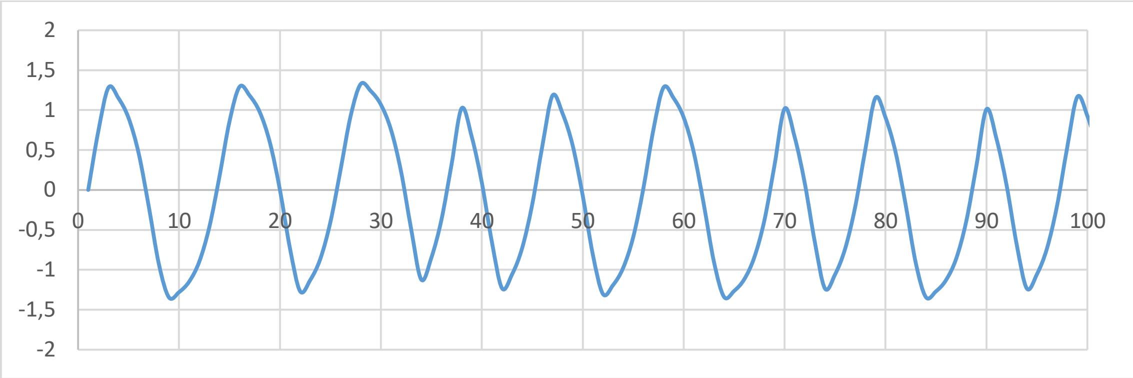
RF = 10





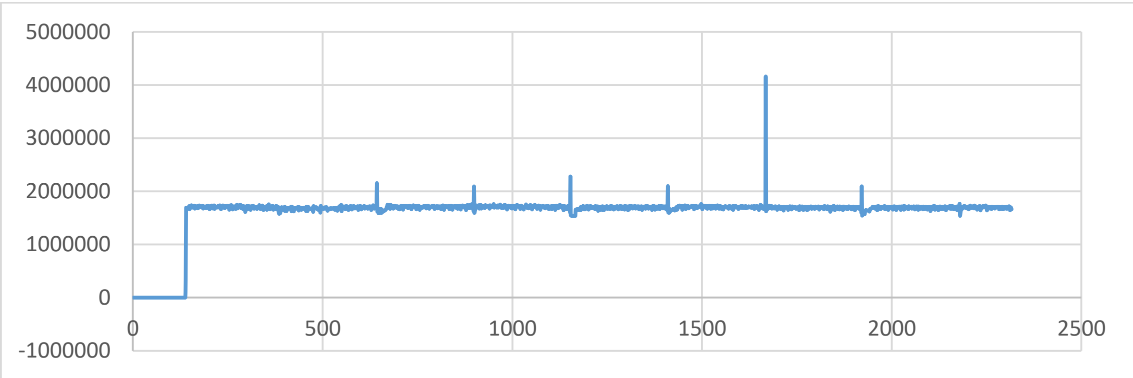
RF = 50





As we can see, the higher the frequency, the harder it is for the process to keep the shape of the sine wave. The lack of points in the peaks makes it hard to keep up with the amplitude of the wave.

We have tried several plots with a RF of 50 and we have had satisfying results like the previous one and other not so satisfying as the following one:



Where in a given moment, our system became uncontrollable and the equation raised to very high numbers.

So we can say that our higher frequency is more or less 50 Hz.

I have tried to modify the discrete sine equation, and instead of using the original formula:

$$Y[k] = Y[k-1] - DT \cdot \left(1 - \frac{(Y[k-1])^2}{2}\right) \cdot 2 \cdot pi \cdot RF$$

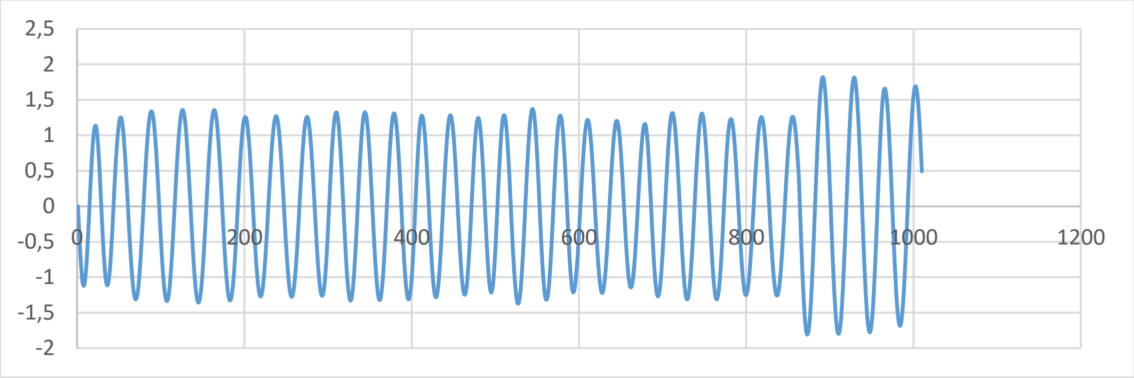
We have obtained the difference equation of a sine wave:

$$Y[k] = U[k-1] \cdot sin(2 \cdot pi \cdot RF \cdot DT) - U[k] \cdot sin(2 \cdot pi \cdot RF \cdot DT) - Y[k-2] + Y[k-1] \cdot 2 \cdot cos(2 \cdot pi \cdot RF \cdot DT);$$

Where $U[k]$ is the equation of an input step (a value that starts with 0 and changes to 1 at the first iteration)

Using this second formula we have been able to achieve higher frequencies before it becomes uncontrollable:

RF = 80



The previous formula has two problems:

It relies on knowing two previous points, not only one (not a very big problem).

The DT has to be constant. In this simulation it wasn't, that's why the amplitude keeps changing. Each time we send data through the socket and we receive it back, etc. the DT is slightly different, that's why this phenomenon happens.

So if we implemented a deterministic system where DT always had to remain the same, we could achieve a perfect sine wave with a constant amplitude and frequency.