The algorithm works like this:

- Starting with the time-series data, x_n , form a Hankel matrix by "striping" the data into columns, with the elements in each successive column shifted by one. For example, if the original time series data is $x=[x_{1,} x_{2,} x_{3,} \dots , x_{10}]$ then a Hankel matrix formed from the data is

$$H=\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ x_2 & x_3 & x_4 & x_5 \\ x_3 & x_4 & x_5 & x_6 \\ x_4 & x_5 & x_6 & x_7 \\ x_5 & x_6 & x_7 & x_8 \\ x_6 & x_7 & x_8 & x_9 \\ x_7 & x_8 & x_9 & x_{10} \end{pmatrix}$$

- Note that the Hankel matrix is not necessarily square (although it can be). Also a given vector x can be used to create many different Hankel matrices of different shapes. The number of rows/columns can be varied depending upon the requirements of the problem at hand.
- Next, take the SVD of the Hankel matrix, $[U, S, V]=svd(H)$
- Then zero out the unwanted, higher-order singular values, $S(N:end)=0$
- Then reform the matrix using the new *S*, $H_n=USV^T$
- Although this matrix is not necessarily a Hankel matrix, you can treat it as such. Unwrap H_n back into a vector by extracting the first column, then the last row of the matrix. This is the "denoised" time series.

This procedure works well when a periodic signal has been corrupted with additive white Gaussian noise since the periodic signal is well captured by the largest few singular values, and the unwanted noise is spread out over many small singular values.