Here is an example code in Matlab to apply a shift in time to a 2D Fourier transform:

% Load the signal and calculate its 2D Fourier Transform

signal = imread('example\_image.png');

fourier\_transform = fft2(signal);

% Define the desired shifts in the x and y directions (in pixels)

delta\_x = 10;

delta\_y = 20;

% Create a meshgrid of the same size as the Fourier transform matrix

[N, M] = size(fourier\_transform);

[x, y] = meshgrid(0:M-1, 0:N-1);

% Calculate the phase shift for each point in the meshgrid

phase\_shift = -2 \* pi \* (delta\_x \* x / M + delta\_y \* y / N);

% Create a complex exponential matrix using the phase shift values

complex\_exponential = exp(1j \* phase\_shift);

% Multiply the Fourier transform matrix by the complex exponential matrix element-wise

shifted\_fourier = fourier\_transform .\* complex\_exponential;

% Calculate the inverse Fourier Transform of the shifted Fourier transform matrix

shifted\_signal = ifft2(shifted\_fourier);

% Show the original and shifted signals side by side

figure;

subplot(1, 2, 1);

imshow(signal);

title('Original signal');

subplot(1, 2, 2);

imshow(abs(shifted\_signal));

title('Shifted signal');

In this code, signal is a 2D signal (e.g., an image) that you want to shift in time. The code calculates the 2D Fourier Transform of the signal using the fft2() function.

The desired shifts in the x and y directions are specified using the delta\_x and delta\_y variables.

The code then creates a meshgrid of the same size as the Fourier transform matrix using the meshgrid() function. The phase shift for each point in the meshgrid is calculated using the equation I gave you earlier. A complex exponential matrix is created using these phase shift values.

The Fourier transform matrix is multiplied by the complex exponential matrix element-wise to apply the phase shift in the frequency domain. The inverse Fourier Transform of the shifted Fourier transform matrix is then calculated using the ifft2() function to obtain the shifted signal in the time domain.

Finally, the original and shifted signals are shown side by side using the imshow() function.