polar_plot

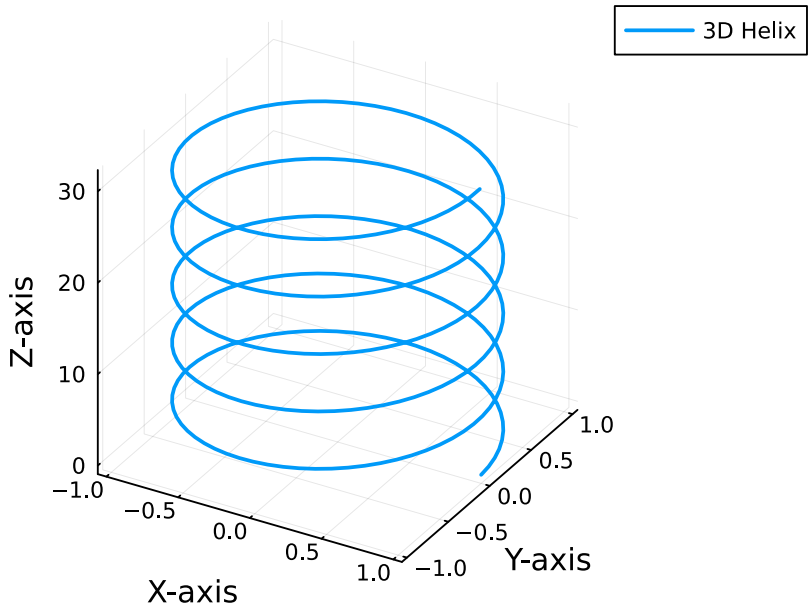
December 9, 2024

[1]:
```
using FFTW, LinearAlgebra
```

[2]:
```
include("modules/operations.jl");
```

[3]:
```
using Plots

# Define parameters
t = 0:0.1:10   # Range for the parameter
x = cos.(t)  # X-coordinates (helix)
y = sin.(t)  # Y-coordinates (helix)
z = t        # Z-coordinates (height of the helix)

# Create 3D plot
plot(x, y, z
    # , proj=:circle
    , linewidth=2
    , label="3D Helix"
    , title="3D Helix Example"
    , xlabel="X-axis", ylabel="Y-axis", zlabel="Z-axis"
)
```

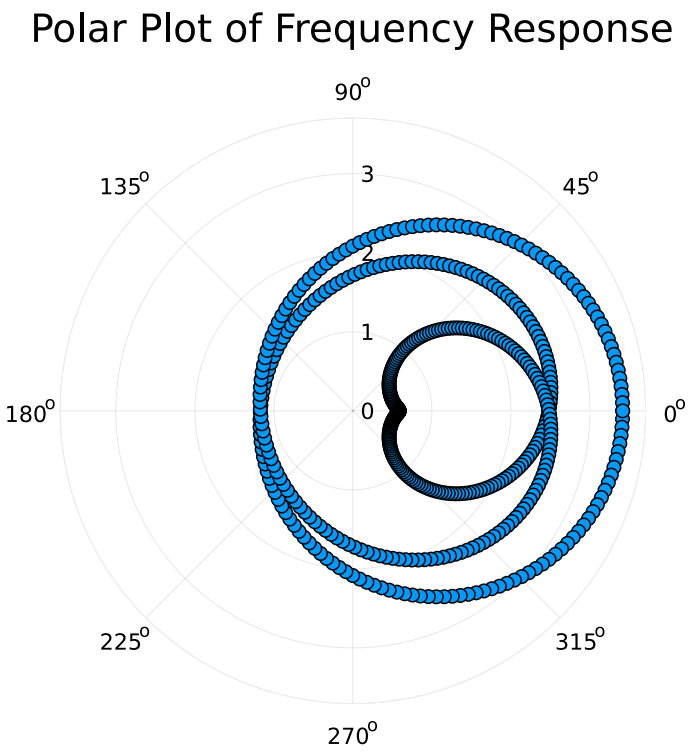
[3]:



[4]:
```julia
using Plots

# Define the transfer function H(e^jw) for the discrete-time system
function H( )
    # Complex exponential terms
    z  =  ^( j *  )  # e^(j )
    z  =  ^(2j *  )  # e^(j2 )

    # Transfer function
    numerator = z ^2 - √(2) * z  + 1  # (z^2 - sqrt(2)z + 1)
    denominator = z ^2  # (z^2)
    return numerator / denominator
end

# Frequency range (from -  to   for discrete systems)
  = range(- ,  , length=500)

# Magnitude and phase response
magnitude = abs.(H.( ))
phase = angle.(H.( ))  # Phase in radians

# Polar plot (magnitude and phase)
plot(phase, magnitude
```

```
    , proj=:polar, marker=:circle
    , title="Polar Plot of Frequency Response", legend=false
)
```

[4]:



[ ]: