Politechnika
Wrocławska

Sygnay i obrazy cyfrowe
Laboratorium 1 - Aliasing 2D

Informatyczne Systemy Automatyki

Wykonujący:
Igor Potyrała - 272518

Prowadzący - Przemysław Śliwiński

Data laboratoriów: 11 października 2023

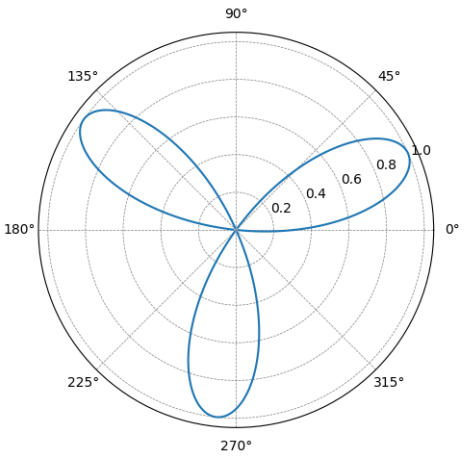
1 Zadania

1.1 Sekwencja obrazów

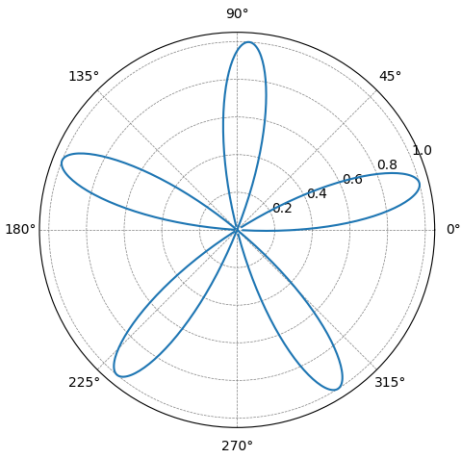
Celem zadania pierwszego było wygenerowanie sekwencjii $M = 64$ obrazów przedstawiających kręcacę się śmigło, z $n = 3, 5$ łopatkami. Wykonać to można byłO za pomocą współrzędnych biegunowych oraz funkcji:

$$f(x) = \sin(nx + \frac{m\pi}{RPM})$$

m - aktualna klatka, gdzie $m \subset < -32; 32 >$
RPM - liczba obrotów na minute, u nas $RPM = 10$
x - wartość X współrzędnych biegunowych
n - liczba łopatek



Obraz 1. Kręcacę się śmigło z $n = 3$ łopatkami.



Obraz 2. Kręcacę się śmigło z $n = 5$ łopatkami.

Następnie by wygenerować kręcącę się śmigło skorzystamy z funkcji FuncAnimation oraz PillowWriter biblioteki matplotlib.

```
def animate_propeller(frame):
    r = np.sin(BLADES * x + (frame * np.pi / RPM))
    plot.set_data(x, r)
```

Kod 1. Generacja obrazu śmigła.

```
# first_fig - polar axis / plot
# animate_propeller - function that will be called %FRAMES times
# np.range(....) - args of the function (animate_propeller)
# FRAMES - M = <-64, 64>
# FPS - number of frames per second showed in the .gif file
FuncAnimation(first_fig, animate_propeller, np.arange(-FRAMES / 2 , FRAMES / 2, 1)).save(
    "propeller_animation.gif", writer=PillowWriter(fps=FPS))
```

Kod 2. Animacja obrazu.

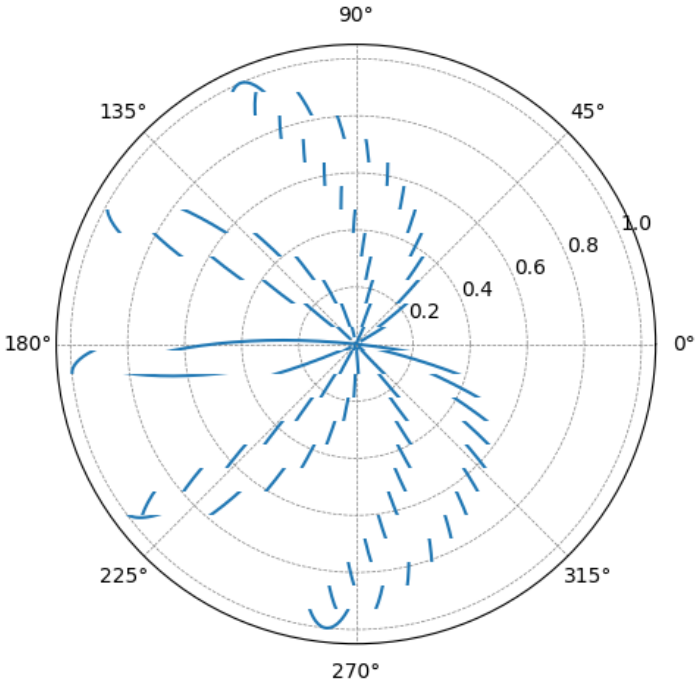
1.2 Sensor

Drugie zadanie wymagało sprawdzenia szybkości sensora mającego rozdzielczość 256 x 256 pikseli. Sensor był w stanie odczytać l = 1, ... , 16 linii. Następnie należało utworzyć film uruchamiający sekwencję obrazów "w kółko". Można było tego dokonać robiąc zrzuty ekranu i wklejając je po kolei od góry obrazu zależnie od zmiennej l = 1, ..., 16.

```
# Shutter Effect
shutter_effect = Image.new('RGB', (512, 512))
def shutter_effect(frame):
    r = np.sin(BLADES * x + (frame * np.pi / RPM))
    plot.set_data(x, r)

    global temp
    plt.savefig('frame.png')
    shutter_effect.paste(Image.open('frame.png').crop((0,
        temp * JUMP, 512, temp * JUMP + JUMP)), (0, temp * JUMP))
    temp += 1
```

Kod 3. Funkcja imitująca sensor.



Obraz 3. Obraz śmigła wygenerowany przez funkcje powyżej, $n = 5$.

2 Wnioski

Zakłócenia ukazujące się w obrazie 3 wynikają z niespełnienia warunków twierdzenia o próbkowaniu. Obiekt porusza się zbyt szybko, by macierz sensora zarejestrowała wszystkie piksele. Przykładowe sposoby na zniwelowanie tego efektu:

- Unikanie ruchu podczas filmowania / trzymanie kamery nieruchomo zminimalizuje zniekształcenia,
- Zwiększenie prędkości odczytu sensora, by była większa od prędkości obracania się łopatek.

* Uniwersalną funkcją może ta ,którą użyliśmy w poprzednich zadaniach, $f(x) = sin(nx + \frac{m\pi}{RPM})$, wystarczy zmieniać tylko parametr n w zależności od tego ile śmigieł chcemy. Przy większej ilości łopatek warto będzie, także zwiększyć wielkość generowanego obrazu.