



Busan science high school

2023 Ocean ICT Festival

2023 BOIF

Youtube 영상 QR

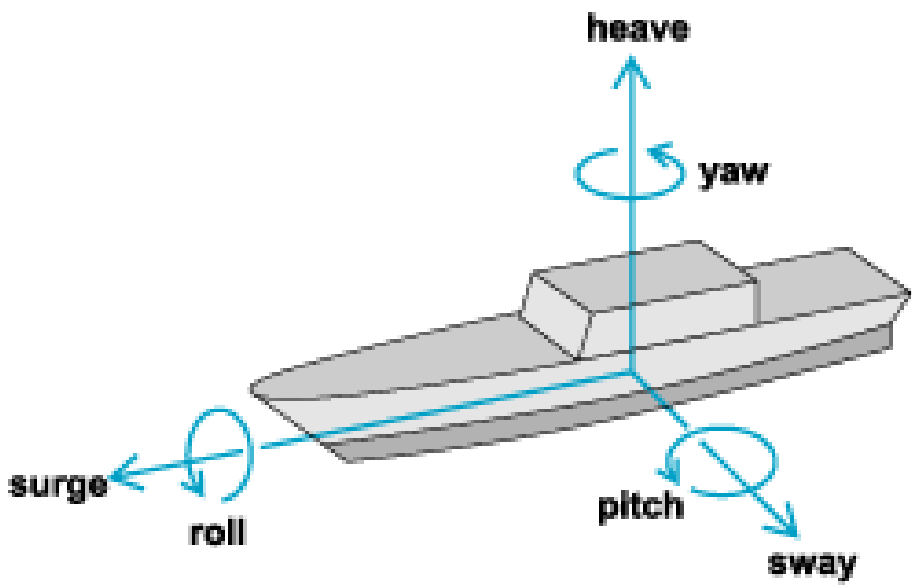
선박 안정화 프로그램 설계

#탐구동기



선박을 운영하는 과정에서 파도 등의 이유로 선박의 안정성이 빠르게 깨져 전복사고가 발생할 수 있다. 그래서 우리는 자동으로 배를 안정화시키고 조종사의 부주의로 인한 사고를 방지하고자, 기울어진 정도 등 외부정보를 기반으로 선박을 안정화할 수 있는 프로그램을 제작해보기로 하였다.

#이론적 배경



1. 배의 운행 3요소

배의 운행 3요소에는 roll, pitch, yaw가 있는데, roll, pitch, yaw는 각각 x,y,z축을 기준으로 선박이 기울어진 각도를 의미한다. 우리는 배의 길이와 roll, pitch, yaw값을 이용하여 배 양끝 4점의 좌표를 계산해 그래프로 나타내고자 했다.

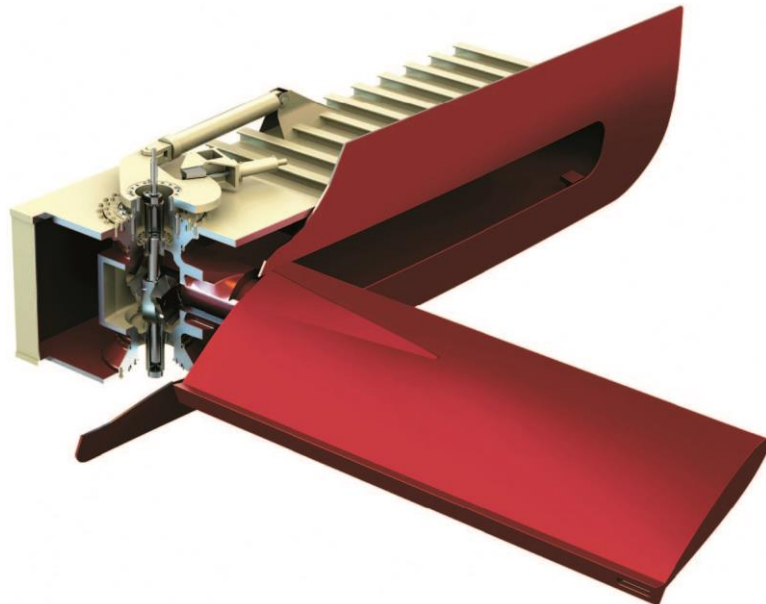
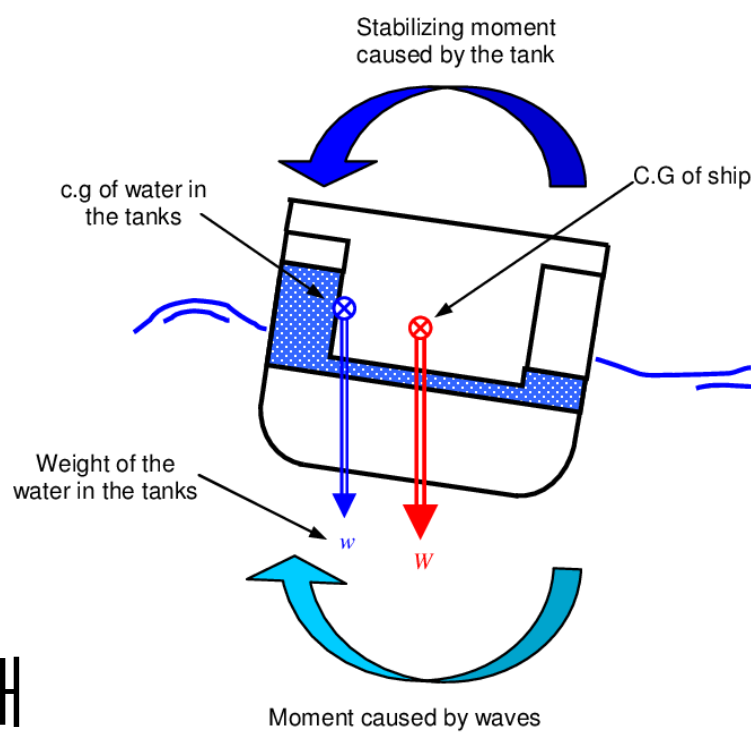
2. 선박 안정화 장치

-Anti-rolling Tank

큰 질량을 가진 물탱크를 이동시켜 질량중심을 옮겨 배를 안정화

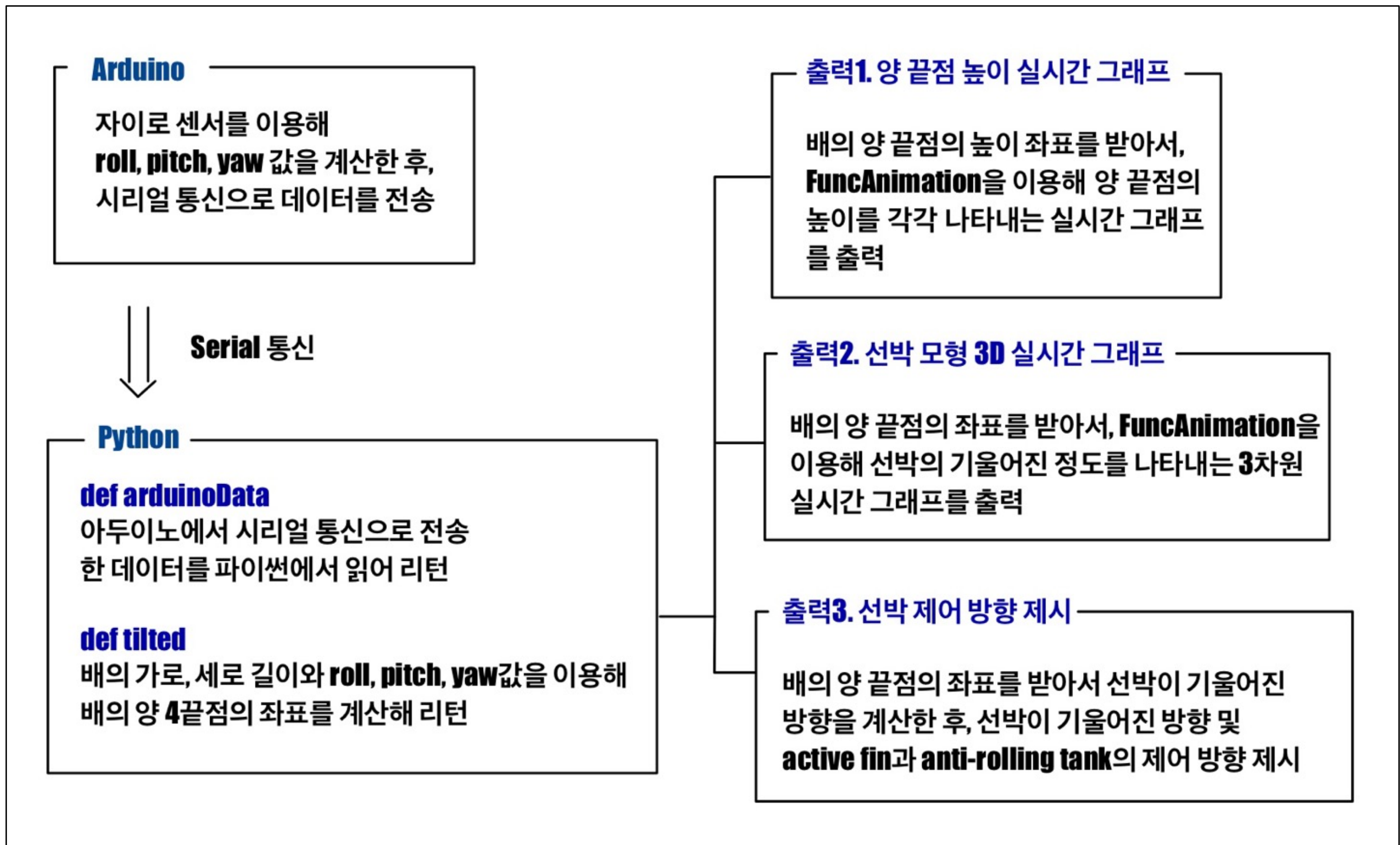
-Active Fin

배의 오른쪽과 왼쪽에 달려있는 fin 2개를 사용해 배를 회전시키는 토크를 발생시켜 배를 안정화



#코드 설명

<전체 흐름도>



#선박 제어방향 출력

```
def suggestDir(f_r,f_l,b_r,b_l):
    if f_r + f_l - b_r - b_l > 10:
        f_b = 1
        # 앞쪽이 들려있음
    elif f_r + f_l - b_r - b_l < -10:
        f_b = -1 # 뒷쪽이 들려있음
    else: f_b = 0

    if f_r + b_r - f_l - b_l > 10:
        r_l = 1 # 오른쪽이 들려있음
    elif f_r + b_r - f_l - b_l < -10:
        r_l = -1 # 왼쪽이 들려있음
    else: r_l = 0
    return f_b,r_l

def guideperson(f_b,r_l):
    c=""
    c+= "앞쪽이 들려있습니다." if f_b==1 else "뒤쪽이 들려있습니다." if f_b==-1 else " "
    c+= "오른쪽이 들려있습니다." if r_l==1 else "왼쪽이 들려있습니다." if r_l==-1 else " "
    return c

def tank_wing(f_r,f_l,b_r,b_l):
    d=suggestDir(f_r,f_l,b_r,b_l)
    f_b=d[0]
    r_l=d[1]
    wing='down' if f_b==1 else 'up' if f_b==-1 else 'stable'
    wing2='left' if r_l==1 else 'right' if r_l==-1 else 'stable'
    wt1='front' if f_b==1 else 'behind' if f_b==-1 else 'stable'
    wt2='left' if r_l==1 else 'right' if r_l==-1 else 'stable'
    return wing, wing2, wt1, wt2

while True:
    r,p,y=arduinoData()
    height=tilted(r,p,y)[2]
    calculated= suggestDir(height[0], height[1], height[2], height[3])
    calculated2=tank_wing(height[0], height[1], height[2], height[3])
    print(f'guideperson(calculated[0], calculated[1])<w>wing(calculated2[0], calculated2[1]))<w>water tank(calculated2[2], calculated2[3]))')
```

아두이노로부터 받은 각 지점의 높이를 이용해서 현재 배의 상황을 파악한다. 또한 그 결과를 이용해 배의 상황에 따라 제어하는 방향을 제시하고, 편을 제어하기 위한 변수를 반환한다. 출력 예시) '앞쪽이 들려있습니다. 오른쪽이 들려있습니다.'

#실시간 그래프출력_선박 3D그래프

```
fig = plt.figure()
ax = plt.axes(projection='3d')
X, Y = np.meshgrid(np.linspace(-a, a, 50), np.linspace(-b, b, 50))
Z = np.zeros_like(X)
line=ax.plot_surface(X, Y, Z,alpha=0.5, color='blue')
ax.set_title('ship')

def update(frame):
    x,y,z,fr,fl,br,bl= tilted(arduinoData(frame))
    points = [[X[0], y[0], z[0]],
               [X[1], y[1], z[1]],
               [X[2], y[2], z[2]],
               [X[3], y[3], z[3]]]
    v1 = np.array(points[1]) - np.array(points[0])
    v2 = np.array(points[2]) - np.array(points[0])
    normal = np.cross(v1, v2)
    x_min = min(x)
    x_max = max(x)
    y_min = min(y)
    y_max = max(y)
    xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.1), np.arange(y_min, y_max, 0.1))
    zz = (-normal[0] * xx - normal[1] * yy - points[0][2]) / normal[2]
    line.remove()
    line=ax.plot_surface(xx,yy,zz,alpha=0.5, color='blue')
    ax.relim()
    ax.autoscale_view()

ani=FuncAnimation(fig, update, frames=range(500), interval=200, repeat=False)

plt.show()
```

3D 그래프 출력을 위해 Numpy 기능을 이용해 각 지점의 x, y, z 값을 묶어서 행렬로 만들어 준다. 또 출력을 할 기반이 되는 3차원 평면에 격자점을 선안해주고 x,y 에 대해 최고점, 최저점까지 그래프를 출력하게 설정하여 보기 쉽도록 하였다. 반복없이 0.2초마다 한 번씩 출력되도록 인터벌을 조정하여 출력한다.

#실시간 그래프출력_끝점의 높이

```
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

fig=plt.figure()
ax=fig.add_subplot()
line_fr = ax.plot([], [], 'r-')
line_fr = ax.plot([], [], 'b-')
line_fl = ax.plot([], [], 'g-')
line_br = ax.plot([], [], 'y-')
height_fr=[]
height_fr=[]
height_fl=[]
height_br=[]

def update(frame):
    global height_fr, height_fr, height_fl, height_br
    roll, pitch, yaw=arduinoData()
    inputvalue=tilted(roll, pitch, yaw)
    height_fr.append(inputvalue[4][2])
    height_fr=height_fr[-20:]
    height_fr.append(inputvalue[3][2])
    height_fr=height_fr[-20:]
    height_fl.append(inputvalue[6][2])
    height_fl=height_fl[-20:]
    height_br.append(inputvalue[5][2])
    height_br=height_br[-20:]

    line_fr.set_data(x*(0.1 for x in range(frame-len(height_fr)+1,frame+1)),height_fr)
    line_fr.set_data(x*(0.1 for x in range(frame-len(height_fr)+1,frame+1)),height_fr)
    line_fl.set_data(x*(0.1 for x in range(frame-len(height_fl)+1,frame+1)),height_fl)
    line_br.set_data(x*(0.1 for x in range(frame-len(height_br)+1,frame+1)),height_br)
    ax.relim()
    ax.autoscale_view()

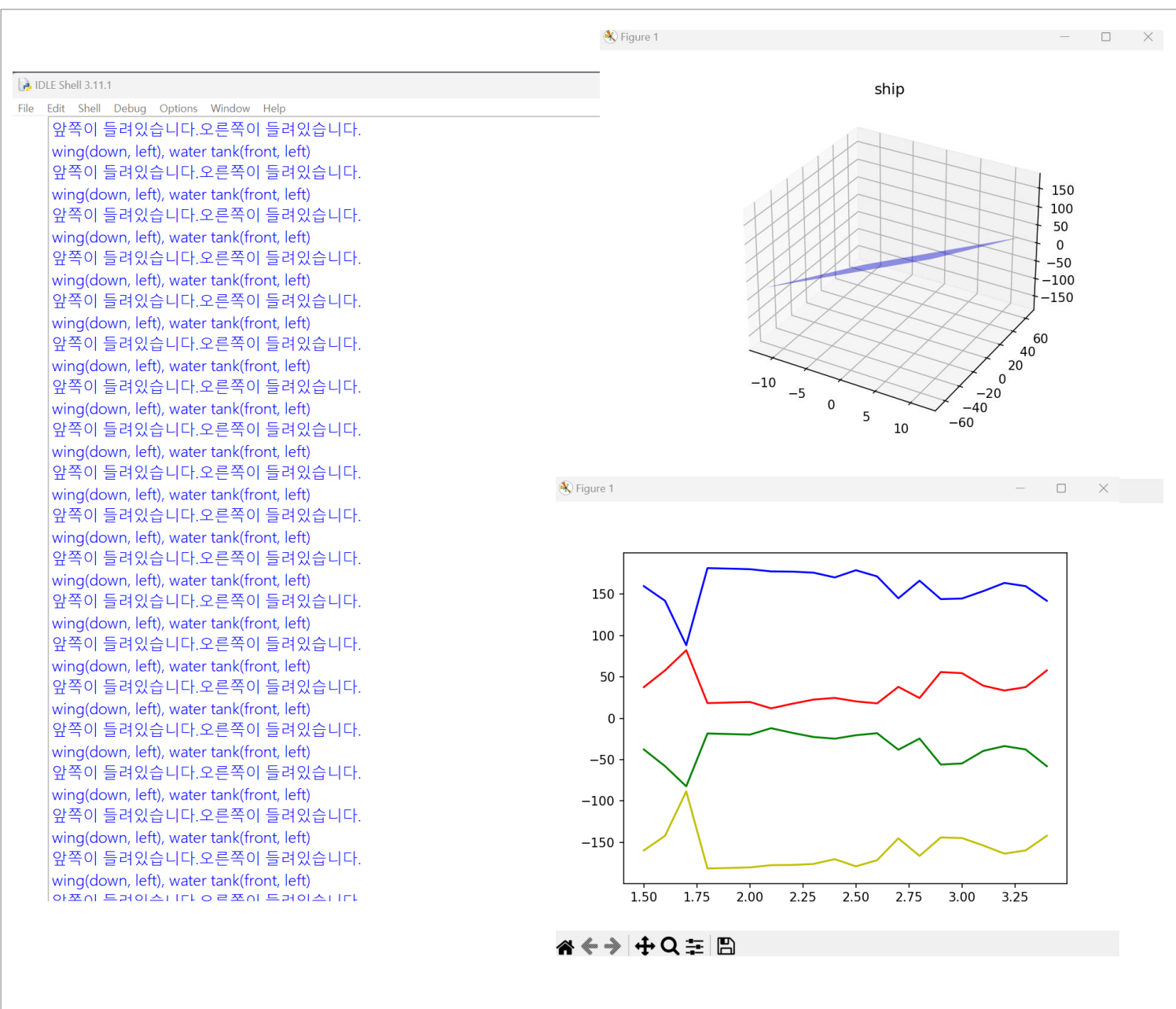
    return line_fr, line_fr, line_fl, line_br

ani=FuncAnimation(fig, update, frames=range(500), interval=200, repeat=False)

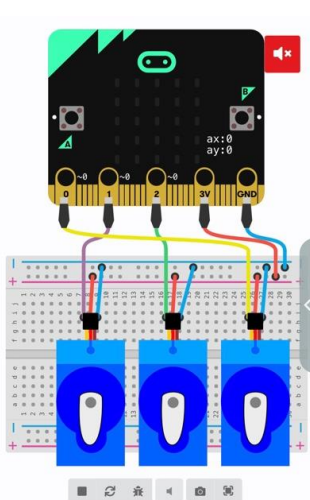
plt.show()
```

Matplotlib의 Pyplot 기능과 animation 기능을 활용하여 0.2초마다 실시간으로 그래프를 출력하는 코드이다. 배의 각 지점을 각각 다른 색으로 동시에 출력한다. 사용하는 데이터는 앞에서 계산한 지점별 높이값 리스트이다. 초당 프레임수와 함숫값, x축, y축을 설정한 뒤, Pyplot을 이용해서 그래프를 출력한다.

#실제 작동화면



결론



본래의 계획은 아두이노로 배의 상태를 출력하고 서보모터를 조작하여 배를 안정화시키는 것이었지만, 아두이노를 만드는 과정에서 문제가 생겨, 아두이노는 배의 상태를 출력하는 역할로만 사용하고 배의 안정화를 위한 조작은 마이크로비트를 사용하여 Active fin, Water tank, Gyro Stabilizer를 조작하여 배가 스스로 중심을 잡게 만들어, 배가 안정해지도록 하였다.