In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
```

Homework 7

Eugeniu Vezeteu - 886240

Define the robot dynamic and Euler method

In [2]:

```

def robot_dynamic(t,x,u):
    return np.array([u[0]*np.cos(x[2]), u[0]*np.sin(x[2]), u[1]])

def euler(f,t_now, x_now,u_now, dt):
    return x_now + f(t_now, x_now,u_now)*dt

def euler_propagate(f,t, x_init,u,dt):
    x_res = np.zeros((u.shape[0],x_init.shape[0]))
    x_res[0] = x_init
    for i in range(x_res.shape[0]-1):
        x_res[i+1] = euler(f,t[i], x_res[i],u[i],dt)
    return x_res
```

In [81]:

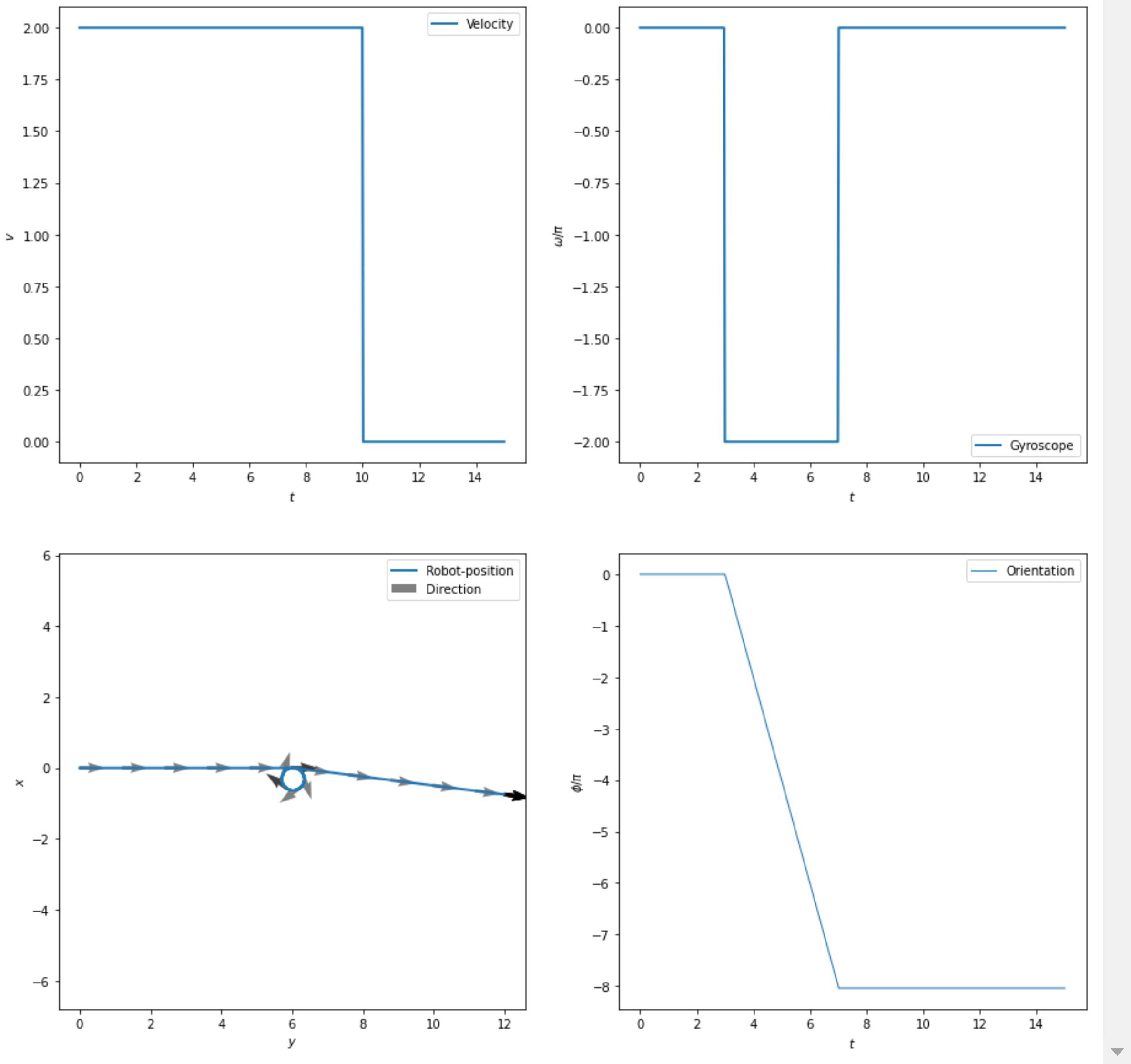
```
robot_init = np.array([0.,0.,0.]) #Task a), start at 0,0,0
t_robot = np.linspace(0.,15.,501)
dt_robot = t_robot[1]-t_robot[0]
u_robot = np.zeros((t_robot.shape[0],2))

#Task b) - Construct speed and gyroscope signals
for i in range(t_robot.shape[0]):
    if 0<= t_robot[i] <= 10:
        u_robot[i,0] = 2.
    else:
        u_robot[i,0] = 0

    if 3<= t_robot[i] <= 7:
        u_robot[i,1] = -2*np.pi #360-degree turn clockwise
    else:
        u_robot[i,1] = 0

#Task 3 - Solution with Euler method
x_robot_euler = euler_propagate(robot_dynamic, t_robot, robot_init, u_robot, dt_rob
```

In [84]:

```
#Visualize the solution
f, ax = plt.subplots(2,2, figsize=(15,15))
skip = 20
ax[0,0].plot(t_robot,u_robot[:,0], label='Velocity', linewidth=2)
ax[0,0].set_xlabel('$t$')
ax[0,0].set_ylabel('$v$')
ax[0,0].legend()

ax[0,1].plot(t_robot,u_robot[:,1]/np.pi, label='Gyroscope', linewidth=2)
ax[0,1].set_xlabel('$t$')
ax[0,1].set_ylabel('$\omega / \pi$')
ax[0,1].legend()

ax[1,0].plot(x_robot_euler[:,0],x_robot_euler[:,1], label='Robot-position', linewid
ax[1,0].quiver(x_robot_euler[::skip,0],x_robot_euler[::skip,1],
               np.cos(x_robot_euler[::skip,2]),np.sin(x_robot_euler[::skip,2]),
               label='Direction', linewidth=0.5, alpha=0.5)
ax[1,0].axis('equal')

ax[1,0].set_xlabel('$y$')
ax[1,0].set_ylabel('$x$')
ax[1,0].legend()

ax[1,1].plot(t_robot,x_robot_euler[:,2]/np.pi, label='Orientation', linewidth=1)
ax[1,1].set_xlabel('$t$')
ax[1,1].set_ylabel('$\phi / \pi$')
ax[1,1].legend()
```

Out[84]:

```
<matplotlib.legend.Legend at 0x7f807d182898>
```


Velocity
v
t
Gyroscope
ω/π
t
Robot-position
Direction
x
y
Orientation
ϕ/π
t

In [ ]: