Autonomous Underwater Vehicles: A From-Scratch Perception & Control Approach

Sreeganesh Valathara Rajendran

January 4, 2025

Contents

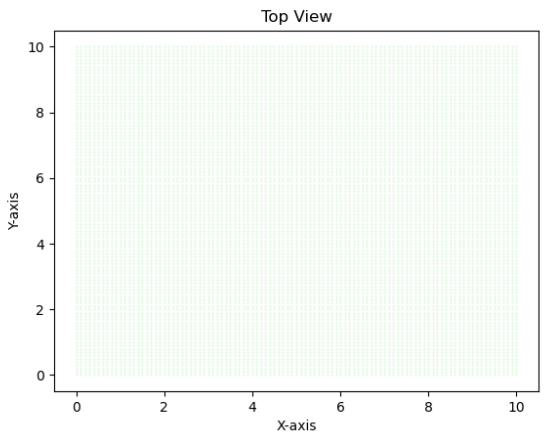
Chapter 1

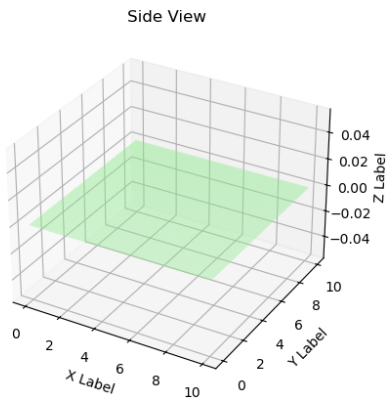
Setup

1.1 Topology Setup

The sea-floor is represented in a discrete manner. That is, the sea-floor is represented by a number of points that has some coordinate and some reflectance value. These two attributes are stored in two separate tensors as follows:

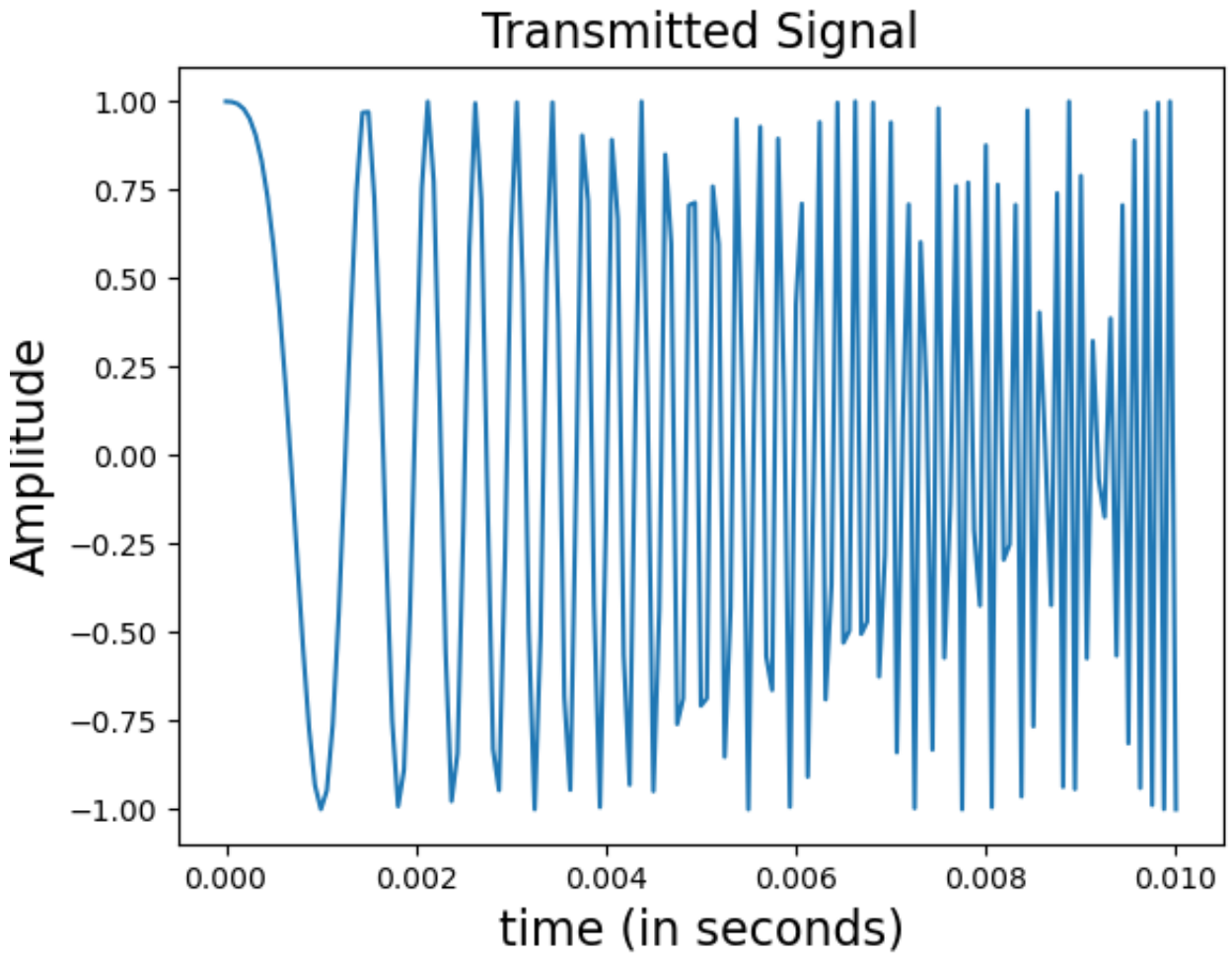
- *location tensor*: this tensor contains the location of all the scatters that are used to represent the sea-floor
- *reflectance tensor*: this tensor contains the reflectivity of the points representing the sea-floor.





1.2 Signal Simulation

The transmitted signal that we're using for this experiment is a chirp signal.


Transmitted Signal
Amplitude
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.000
0.002
0.004
0.006
0.008
0.010
time (in seconds)

Chapter 2

Software

2.1 Classes

2.1.1 Class: AUV

```
# class representing AUV
class AUV:
    # init function
    def __init__(self,
                 location,            # current location of AUV [tensor]
                 velocity,            # velocity of AUV [tensor]
                 acceleration,        # acceleration of AUV [tensor]
                 pointing_direction): # direction in which AUV is pointed
                     [tensor]
        """
        Initializing parameters related to AUV
        """
        # fundamental attributes
        self.location            = location            # current location of AUV
        self.velocity            = velocity            # velocity of AUV
        self.acceleration        = acceleration        # acceleration of AUV
        self.pointing_direction  = pointing_direction  # direction in which AUV
            is pointed

        # add-on attributes
        self.projector_starboard = None  # projector to the right
        self.projector_portside  = None  # projector to the left
        self.projector_fbls      = None  # projector to the front

        self.ula_portside        = None  # ULA mounted on the left
        self.ula_starboard       = None  # ULA mounted on the right

    def summarize(self):
        print(">location           = \n", self.location)
        print(">velocity           = \n", self.velocity)
        print(">acceleration       = \n", self.acceleration)
        print(">pointing_direction = \n", self.pointing_direction)
```

```
31          # print("\t>")
32          # print("\t>")
33          # print("\t>")
34          # print("\t>")
35
36
37      def update_timestep(self):
38          """
39          Updating the after each time step
40          """
41
42      def simulate_signal(self):
43          """
44          Aim: Simulate signals
45          Note:
46              > Project signals from projectors
47              > Return signals from scatters
48              > simulate signals received by ULAs
49          """
50          pass
```

2.1.2 Class: Scatter

2.1.3 Class: Projector

2.2 Function