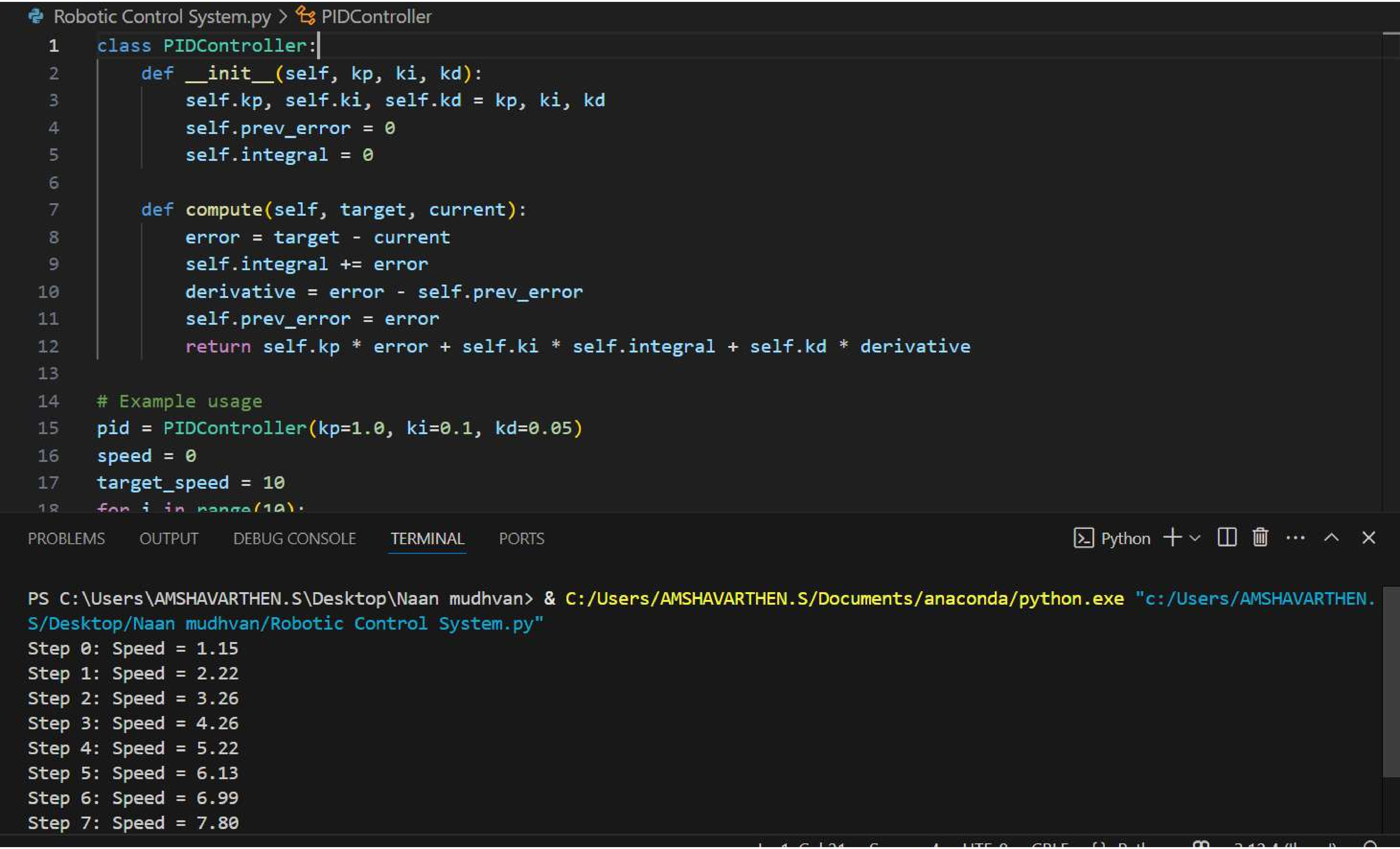
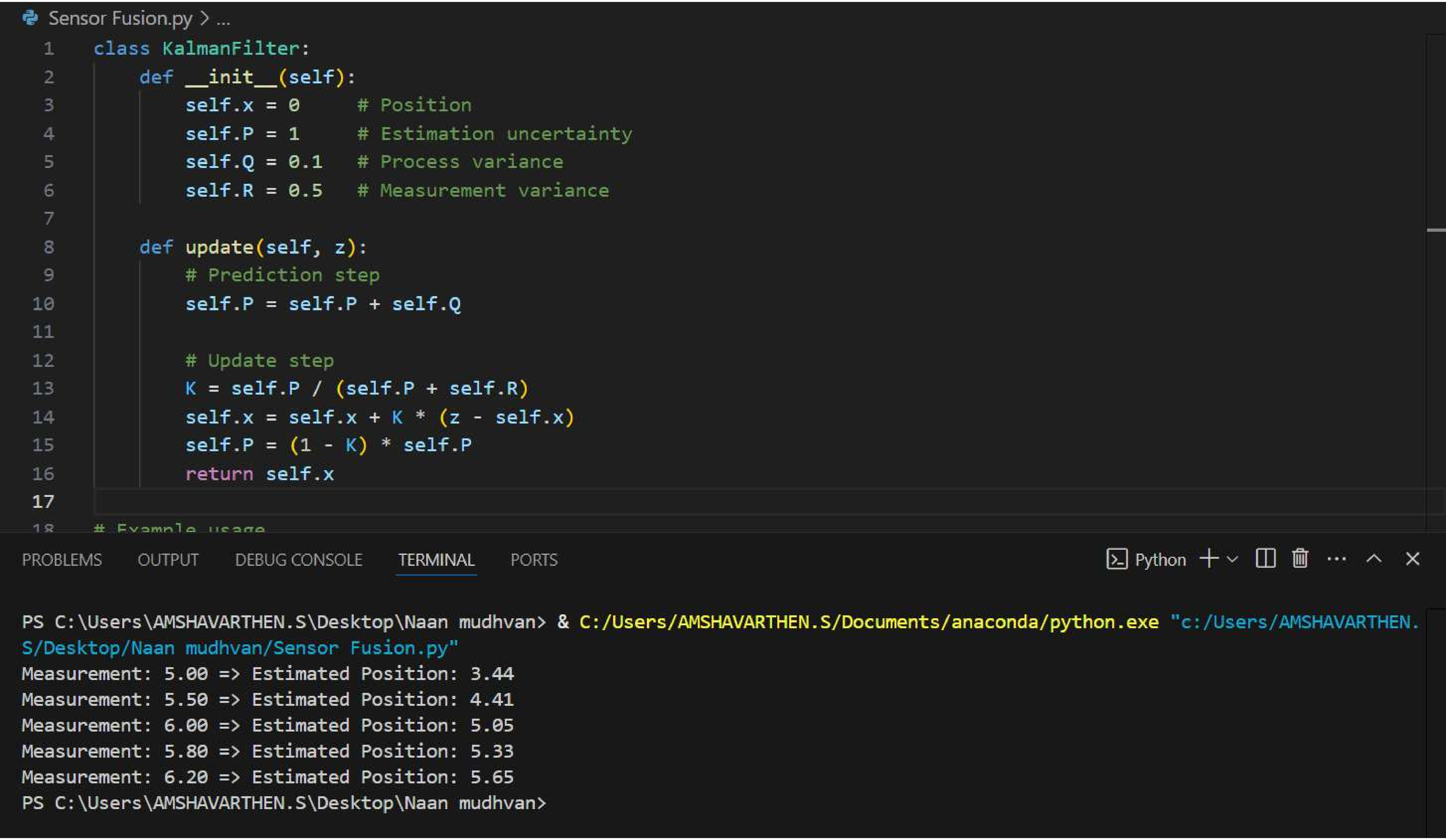


```python
class PIDController:
    def __init__(self, kp, ki, kd):
        self.kp, self.ki, self.kd = kp, ki, kd
        self.prev_error = 0
        self.integral = 0

    def compute(self, target, current):
        error = target - current
        self.integral += error
        derivative = error - self.prev_error
        self.prev_error = error
        return self.kp * error + self.ki * self.integral + self.kd * derivative

# Example usage
pid = PIDController(kp=1.0, ki=0.1, kd=0.05)
speed = 0
target_speed = 10
for i in range(10):
```

```
PS C:\Users\AMSHAVARTHEN.S\Desktop\Naan mudhvan> & C:/Users/AMSHAVARTHEN.S/Documents/anaconda/python.exe "c:/Users/AMSHAVARTHEN.S/Desktop/Naan mudhvan/Robotic Control System.py"
Step 0: Speed = 1.15
Step 1: Speed = 2.22
Step 2: Speed = 3.26
Step 3: Speed = 4.26
Step 4: Speed = 5.22
Step 5: Speed = 6.13
Step 6: Speed = 6.99
Step 7: Speed = 7.80
```



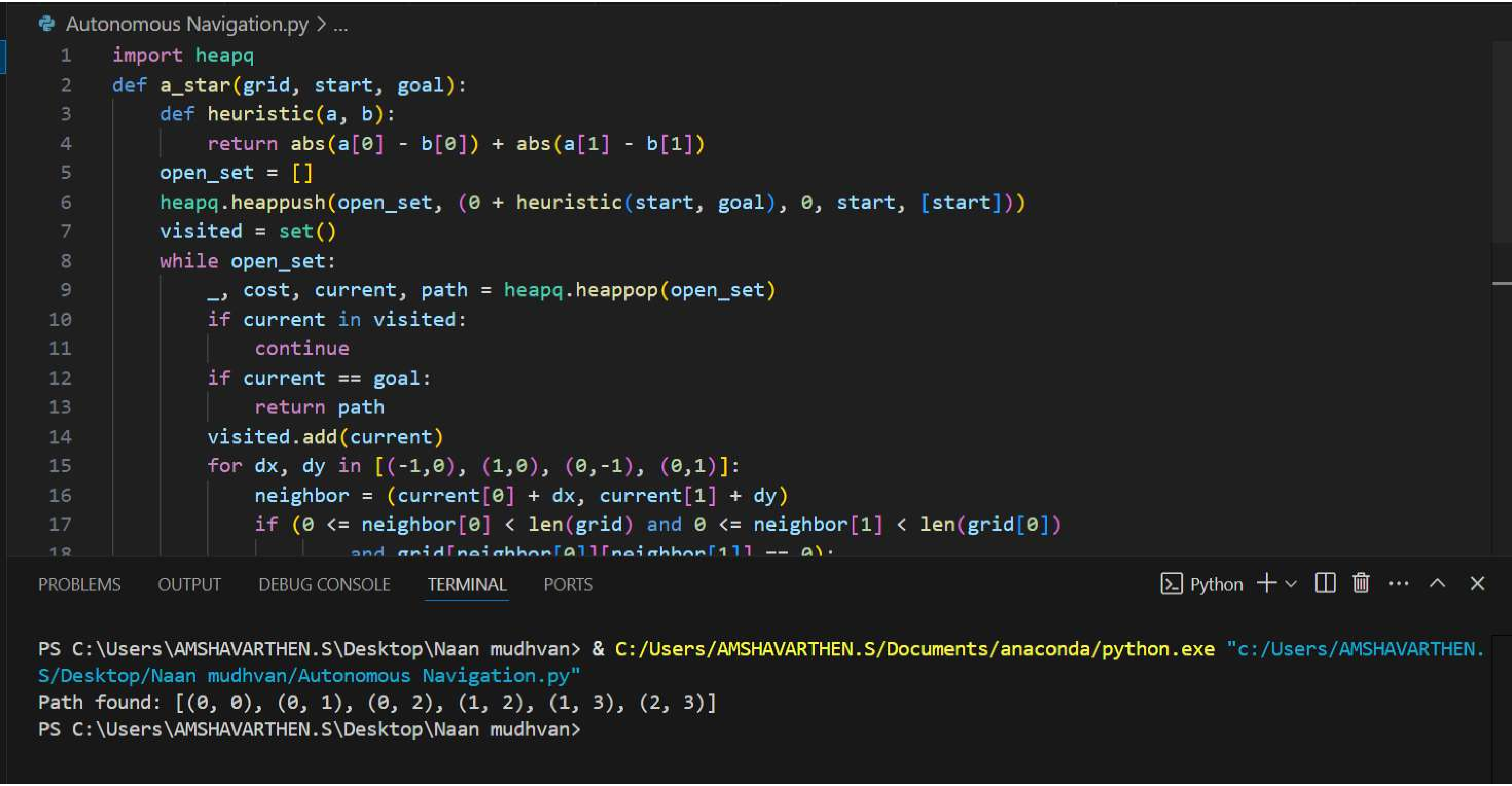
```python
class KalmanFilter:
    def __init__(self):
        self.x = 0     # Position
        self.P = 1     # Estimation uncertainty
        self.Q = 0.1   # Process variance
        self.R = 0.5   # Measurement variance

    def update(self, z):
        # Prediction step
        self.P = self.P + self.Q

        # Update step
        K = self.P / (self.P + self.R)
        self.x = self.x + K * (z - self.x)
        self.P = (1 - K) * self.P
        return self.x

# Example usage
```

```
PS C:\Users\AMSHAVARTHEN.S\Desktop\Naan mudhvan> & C:/Users/AMSHAVARTHEN.S/Documents/anaconda/python.exe "c:/Users/AMSHAVARTHEN.
S/Desktop/Naan mudhvan/Sensor Fusion.py"
Measurement: 5.00 => Estimated Position: 3.44
Measurement: 5.50 => Estimated Position: 4.41
Measurement: 6.00 => Estimated Position: 5.05
Measurement: 5.80 => Estimated Position: 5.33
Measurement: 6.20 => Estimated Position: 5.65
PS C:\Users\AMSHAVARTHEN.S\Desktop\Naan mudhvan>
```



```python
import heapq
def a_star(grid, start, goal):
    def heuristic(a, b):
        return abs(a[0] - b[0]) + abs(a[1] - b[1])
    open_set = []
    heapq.heappush(open_set, (0 + heuristic(start, goal), 0, start, [start]))
    visited = set()
    while open_set:
        _, cost, current, path = heapq.heappop(open_set)
        if current in visited:
            continue
        if current == goal:
            return path
        visited.add(current)
        for dx, dy in [(-1,0), (1,0), (0,-1), (0,1)]:
            neighbor = (current[0] + dx, current[1] + dy)
            if (0 <= neighbor[0] < len(grid) and 0 <= neighbor[1] < len(grid[0])
                    and grid[neighbor[0]][neighbor[1]] == 0):
```

```
PS C:\Users\AMSHAVARTHEN.S\Desktop\Naan mudhvan> & C:/Users/AMSHAVARTHEN.S/Documents/anaconda/python.exe "c:/Users/AMSHAVARTHEN.
S/Desktop/Naan mudhvan/Autonomous Navigation.py"
Path found: [(0, 0), (0, 1), (0, 2), (1, 2), (1, 3), (2, 3)]
PS C:\Users\AMSHAVARTHEN.S\Desktop\Naan mudhvan>
```