ME416 Homework 2 Report

Jilin Zheng

March 18, 2024

Problem 1: Reference Frames and Coordinates

Question 1.1

$$ {}^{B_1}x = {}^{B_1}R_{B_2}\,{}^{B_2}x + {}^{B_1}T_{B_2} \tag{1} $$

Question 1.2

$$ {}^{W}x = {}^{W}R_{B_1}\,{}^{B_1}x + {}^{W}T_{B_1} = {}^{W}R_{B_1}\,{}^{B_1}x \tag{2} $$

Question 1.3

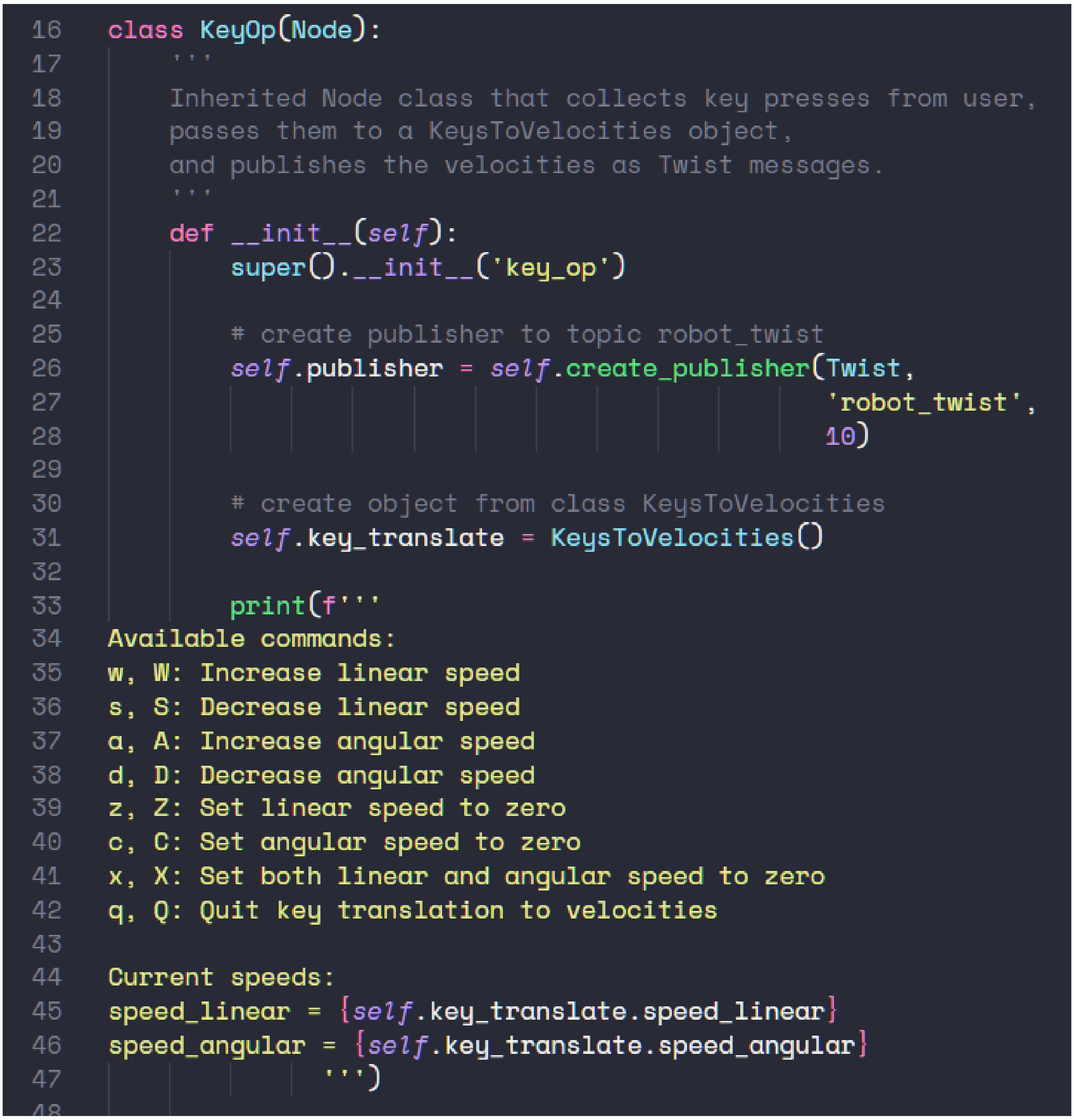
$$
\begin{align}
{}^{W}x &= {}^{W}R_{B_1}\,{}^{B_1}x + {}^{W}T_{B_1} \tag{3} \\
&= {}^{W}R_{B_1}\,({}^{B_1}R_{B_2}\,{}^{B_2}x + {}^{B_1}T_{B_2}) + {}^{W}T_{B_1} \tag{4} \\
&= {}^{W}R_{B_1}\,({}^{B_1}R_{B_2}\,{}^{B_2}x + {}^{B_1}T_{B_2}) \tag{5}
\end{align}
$$

Question 1.4

$$
\begin{align}
{}^{W}x &= {}^{W}R_{B_1}\,({}^{B_1}R_{B_2}\,{}^{B_2}x + {}^{B_1}T_{B_2}) \tag{6} \\
&= {\color{blue}{}^{W}R_{B_1}\,{}^{B_1}R_{B_2}}\,{}^{B_2}x + {\color{red}{}^{W}R_{B_1}\,{}^{B_1}T_{B_2}} \tag{7} \\
&= {\color{blue}{}^{W}R_{B_2}}\,{}^{B_2}x + {\color{red}{}^{W}T_{B_2}} \tag{8}
\end{align}
$$

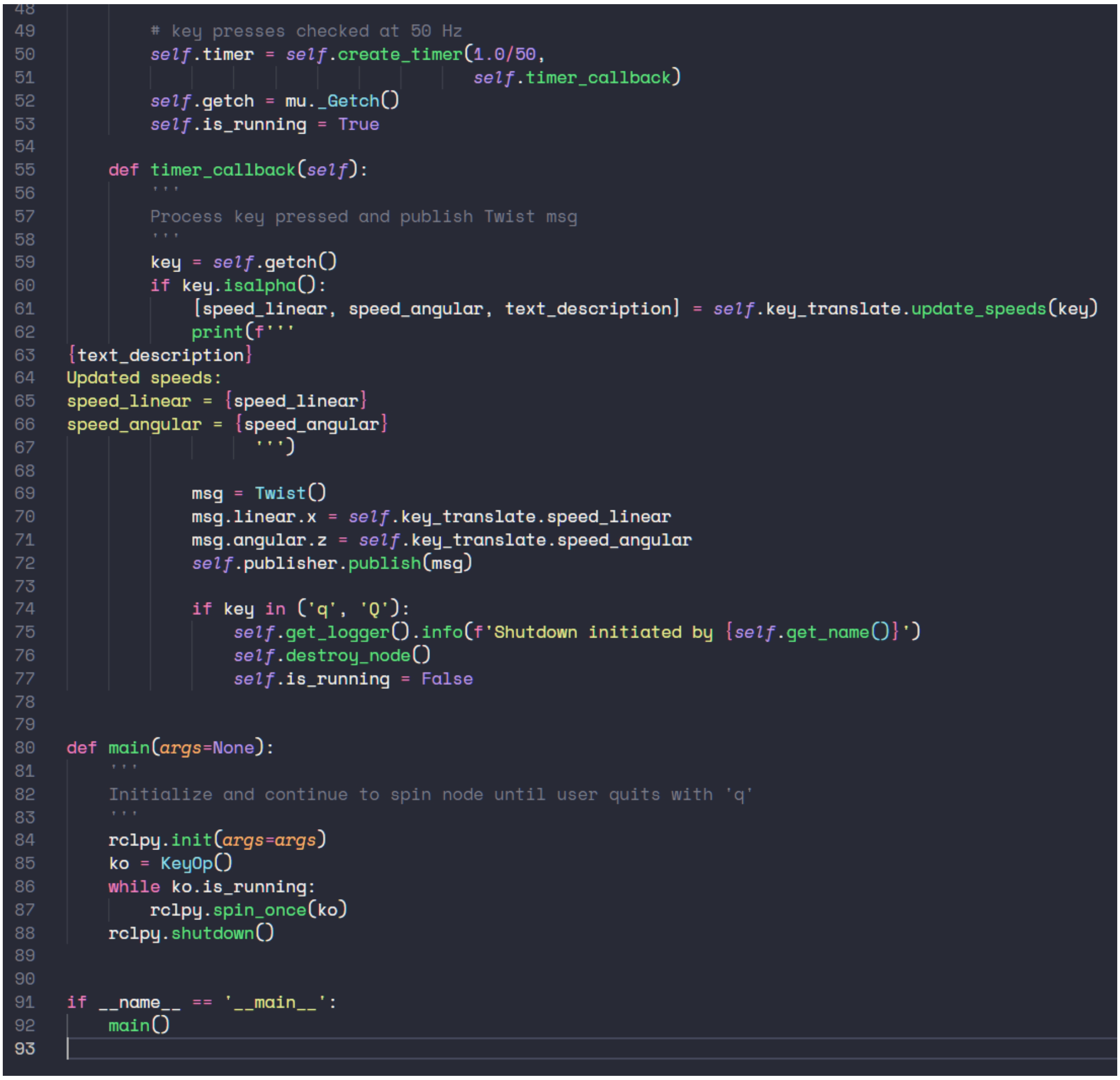
Problem 2: Keyboard Teleoperation

Question 2.1



```python
class KeyOp(Node):
    '''
    Inherited Node class that collects key presses from user,
    passes them to a KeysToVelocities object,
    and publishes the velocities as Twist messages.
    '''
    def __init__(self):
        super().__init__('key_op')

        # create publisher to topic robot_twist
        self.publisher = self.create_publisher(Twist,
                                               'robot_twist',
                                               10)

        # create object from class KeysToVelocities
        self.key_translate = KeysToVelocities()

        print(f'''
Available commands:
w, W: Increase linear speed
s, S: Decrease linear speed
a, A: Increase angular speed
d, D: Decrease angular speed
z, Z: Set linear speed to zero
c, C: Set angular speed to zero
x, X: Set both linear and angular speed to zero
q, Q: Quit key translation to velocities

Current speeds:
speed_linear = {self.key_translate.speed_linear}
speed_angular = {self.key_translate.speed_angular}
              ''')
```

Question 2.2



```
        # key presses checked at 50 Hz
        self.timer = self.create_timer(1.0/50,
                                       self.timer_callback)
        self.getch = mu._Getch()
        self.is_running = True

    def timer_callback(self):
        '''
        Process key pressed and publish Twist msg
        '''
        key = self.getch()
        if key.isalpha():
            [speed_linear, speed_angular, text_description] = self.key_translate.update_speeds(key)
            print(f'''
{text_description}
Updated speeds:
speed_linear = {speed_linear}
speed_angular = {speed_angular}
                  ''')

            msg = Twist()
            msg.linear.x = self.key_translate.speed_linear
            msg.angular.z = self.key_translate.speed_angular
            self.publisher.publish(msg)

            if key in ('q', 'Q'):
                self.get_logger().info(f'Shutdown initiated by {self.get_name()}')
                self.destroy_node()
                self.is_running = False


def main(args=None):
    '''
    Initialize and continue to spin node until user quits with 'q'
    '''
    rclpy.init(args=args)
    ko = KeyOp()
    while ko.is_running:
        rclpy.spin_once(ko)
    rclpy.shutdown()


if __name__ == '__main__':
    main()

```