Node Test:
Position and velocity approximator

Node test set up and prerequisites:

Rclcpp, geometry_msgs, colcon, gtest

Test1 : PublishesPositionVelocity
Tester : Melissa van Leeuwen
Date : 29/11/2025

Short description test: This test verifies that the node correctly processes incoming IMU acceleration data and updates its internal position and velocity by integrating the IMU data.
Start conditions: - Node is initialized with default starting position (0,0) and zero velocity. - No previous IMU messages were processed Input test: - acceleration : x = 0.0 and y = 1.0 - yaw_z = 0.0 - timestamp = now()
Expected result : - pos_y_ > old_y (the new calculated position has increased compared to the old position)
Result test: - The calculated position has increased. The node correctly processes IMU data and integrates the acceleration to calculate the position and velocity.

Test2 : ResetCallbackSetsPose
Tester : Melissa van Leeuwen
Date : 29/11/2025

Short description test: This test verifies whether the reset_callback() of the node correctly updates the node's internal position when receiving a new position. - The node should update its internal position (x,y) and yaw orientation when a new position message is received from the "position determinator" node.

- When reset occurs, all velocities (vx, vy, ωz) must be reset to zero.
- Integration must restart from this new position.

Start conditions:
- Node is initialized with default pose (0,0) and yaw 0 rad.
- All velocities are initially zero.

Input test:
A PoseStamped message with:
Position:
x = 5.0
y = -3.0

Orientation:
yaw = 90° = π/2 rad
Encoded as quaternion:
- z = sin(π/4)
- w = cos(π/4)

Expected result:

Position updated:
pos_x_ = 5.0
pos_y_ = -3.0

Yaw extracted correctly:
yaw_ ≈ π/2 (within tolerance 0.05 rad)

Velocities reset:
vx_ = 0
vy_ = 0
omega_z_ = 0

Result Test :
- Node's internal position updated to (5.0, –3.0).
- Extracted yaw approximately π/2 rad.
- Linear velocities and angular velocity correctly reset to zero.

Test3 : TrapezoidalIntegrationWorks
Tester : Melissa van Leeuwen
Date : 29/11/2025

Short description test:
This test verifies that the node correctly integrates IMU acceleration using the trapezoidal rule. Two IMU messages are sent with a small time difference and checks if the resulting velocity matches.

Input test:

First IMU message
- Acceleration y = 2.0 m/s²
- timestamp = now()
  (This initializes integration (dt = 0, no velocity change))

Second IMU message (after ~50 ms delay)
- Acceleration y = 2.0 m/s²
- timestamp = now()
  (Now a real Δt exists, so velocity should increase)

Expected result:
The node's internal vy_ should match:
$vy = 2.0 \cdot dt$ with tolerance ±0.05

Result Test :
- The node correctly integrated acceleration over the measured Δt.
- The calculated velocity matched the expected value.

Test4 : RotationTransformCorrect
Tester : Melissa van Leeuwen
Date : 29/11/2025

Short description test:
This test verifies whether the node correctly transforms robot-frame accelerations into map-frame accelerations using the current yaw angle.

IMU acceleration values are expressed in the robot frame, where +X = forward and +Y = left. Before integration, these values must be rotated into the global map frame using the robot's yaw angle. When yaw = 90° (π/2), robot-frame +X should point in the map-frame +Y direction.

This test ensures that the yaw-based rotation is applied correctly

Input test:
- Orientation = yaw = 90°

Two IMU messages are sent:
- First message initializes integration (dt = 0).
- Second message is sent after ~30 ms delay.
- Acceleration: x=1.0, y=0.0

Expected result:
With yaw = 90°:
- Robot frame +X = Map frame +Y
- Robot frame +Y = Map –X frame

So after integration, the map frame motion should be:
- Increase in map Y

- No significant change in map X
Result Test : - pos_y_ must increase (motion appears along map +Y). - pos_x_ should remain approximately 0, within ±0.05 tolerance.

Test5 : InvalidDtDoesNotUpdatePositionVelocity
Tester : Melissa van Leeuwen
Date : 29/11/2025

Short description test: This test verifies that the node correctly handles the case where two IMU messages arrive with identical timestamps, resulting in dt = 0. This test ensures that the node does not update velocity or position when such invalid data is received.
Input test: - acceleration: x = 0.0 and y = 5.0 - yaw = 0.0 - timestamp = now() After this message is published, the same message is published with identical timestamp.
Expected result: - pos_y_ remains equal to pos_y_before - vy_ remains equal to vel_y_before
Result Test : - No position change - No velocity change

```
melissa@ubuntu:~/Documents/GitHub/rmb_ws$ ./build/g425_assign4_pkg/utest_position_velocity_approximator
[==========] Running 5 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 5 tests from PositionVelocityTest
[ RUN      ] PositionVelocityTest.PublishesPositionVelocity
[INFO] [1764620338.837437421] [position_velocity_approximator]: PositionVelocityApproximator started.
[INFO] [1764620338.837820893] [position_velocity_approximator]: This test publishes an IMU message to verify if the position and velocity are updated.
[INFO] [1764620338.837952917] [position_velocity_approximator]: Position → x=0.000  y=0.000  yaw=0.000
[INFO] [1764620338.837970855] [position_velocity_approximator]: Velocity → vx=0.000  vy=0.000  wz=0.000
[       OK ] PositionVelocityTest.PublishesPositionVelocity (54 ms)
[ RUN      ] PositionVelocityTest.ResetCallbackSetsPose
[INFO] [1764620338.876803378] [position_velocity_approximator]: PositionVelocityApproximator started.
[INFO] [1764620338.877014734] [position_velocity_approximator]: This test calls reset_callback with a specific pose.
 After the call, the node's position, yaw, and velocities should match the reset values.
[INFO] [1764620338.877034566] [position_velocity_approximator]: Reset pose → x=5.000  y=-3.000  yaw=1.571
[       OK ] PositionVelocityTest.ResetCallbackSetsPose (6 ms)
[ RUN      ] PositionVelocityTest.TrapezoidalIntegrationWorks
[INFO] [1764620338.883380063] [position_velocity_approximator]: PositionVelocityApproximator started.
[INFO] [1764620338.883578497] [position_velocity_approximator]: This test verifies that the node correctly integrates IMU acceleration using the trapezoidal rule.
 Two IMU messages are send with a small time difference and check that the resulting velocity matches.
[INFO] [1764620338.883639661] [position_velocity_approximator]: Position → x=0.000  y=0.000  yaw=0.000
[INFO] [1764620338.883661354] [position_velocity_approximator]: Velocity → vx=0.000  vy=0.000  wz=0.000
[INFO] [1764620338.933921839] [position_velocity_approximator]: Position → x=0.000  y=0.008  yaw=0.000
[INFO] [1764620338.933972038] [position_velocity_approximator]: Velocity → vx=0.000  vy=0.101  wz=0.000
[       OK ] PositionVelocityTest.TrapezoidalIntegrationWorks (57 ms)
[ RUN      ] PositionVelocityTest.RotationTransformCorrect
[INFO] [1764620338.942251632] [position_velocity_approximator]: PositionVelocityApproximator started.
[INFO] [1764620338.942442634] [position_velocity_approximator]: This test sets the robot yaw to 90° and applies a robot-frame acceleration along +X.
 After integration, the motion should appear along +Y in the map frame, with X motion suppressed.
[INFO] [1764620338.942465619] [position_velocity_approximator]: Reset pose → x=0.000  y=0.000  yaw=1.571
[INFO] [1764620338.942525090] [position_velocity_approximator]: Position → x=0.000  y=0.000  yaw=1.571
[INFO] [1764620338.942541537] [position_velocity_approximator]: Velocity → vx=0.000  vy=0.000  wz=0.000
[INFO] [1764620338.972884399] [position_velocity_approximator]: Position → x=0.000  y=0.001  yaw=1.571
[INFO] [1764620338.972938678] [position_velocity_approximator]: Velocity → vx=0.000  vy=0.030  wz=0.000
[       OK ] PositionVelocityTest.RotationTransformCorrect (38 ms)
[ RUN      ] PositionVelocityTest.InvalidDtDoesNotUpdatePositionVelocity
[INFO] [1764620338.979361585] [position_velocity_approximator]: PositionVelocityApproximator started.
[INFO] [1764620338.979545282] [position_velocity_approximator]: This test publishes two IMU messages with the same timestamp.
 The second message should be ignored by the node, ensuring that position and velocity are not updated when dt = 0.
[INFO] [1764620338.979600842] [position_velocity_approximator]: Position → x=0.000  y=0.000  yaw=0.000
[INFO] [1764620338.979618403] [position_velocity_approximator]: Velocity → vx=0.000  vy=0.001  wz=0.000
[       OK ] PositionVelocityTest.InvalidDtDoesNotUpdatePositionVelocity (6 ms)
[----------] 5 tests from PositionVelocitvTest (164 ms total)
[  PASSED  ] 5 tests.
```