Understanding vx, vy, and ω in 3-Wheel Drive Robots

Let's break down the basics in simple language, step by step, so you can really understand how your robot moves and how inverse kinematics helps you control it.

What Do vx, vy, and ω Mean?

- **vx**: This is the robot's speed moving *forward or backward* (along the X-axis of the robot).
- **vy**: This is the robot's speed moving *sideways* (left or right, along the Y-axis of the robot).
- **ω (omega)**: This is the robot's *rotation speed* (how fast it spins or turns around its center).

Think of your robot like a hockey puck on a table:

- If you push it straight ahead, that's **vx**.
- If you push it sideways, that's **vy**.
- If you twist it to spin, that's **ω**.

Why Do These Change?

- **vx, vy, and ω** change based on what you want your robot to do. If you want it to go forward, you set a positive vx. If you want it to go right, you set a positive vy. If you want it to spin, you set ω.
- You can combine these: move forward and spin at the same time, or move diagonally by setting both vx and vy.

How Does the Robot Use These?

Your robot has three wheels, each powered by a motor. To move in any direction or rotate, you need to figure out how fast each wheel should spin. This is where **inverse kinematics** comes in.

Inverse Kinematics: Turning Motion Into Wheel Speeds

Imagine you tell your robot: "Go forward at 0.2 m/s, go right at 0.1 m/s, and spin at 0.1 rad/s."

- The robot uses math to figure out how fast each wheel needs to turn to make that happen.
- Each wheel is placed at a different angle (usually 0°, 120°, and 240°).

The math looks like this (don't worry, you don't need to memorize it):

$$
\begin{aligned}
\omega_1 &= \frac{v_y + R \cdot \omega}{r} \\
\omega_2 &= \frac{0.866 v_x - 0.5 v_y + R \cdot \omega}{r} \\
\omega_3 &= \frac{-0.866 v_x - 0.5 v_y + R \cdot \omega}{r}
\end{aligned}
$$

- $ r $: wheel radius
- $ R $: distance from robot center to wheel
- $ \omega_1, \omega_2, \omega_3 $: speed for each wheel

Why Do We Need This?

- If you just set all wheels to the same speed, the robot will only go forward or backward.
- By using inverse kinematics, you can make the robot move in *any* direction and rotate at the same time.

Visualizing the Motions

- **Forward (vx > 0, vy = 0, ω = 0):** All wheels work together to move straight.
- **Sideways (vx = 0, vy > 0, ω = 0):** Wheels spin in a way that the robot slides sideways.
- **Rotate (vx = 0, vy = 0, ω > 0):** All wheels spin to rotate the robot in place.
- **Diagonal (vx > 0, vy > 0, ω = 0):** Robot moves diagonally.

How Do You Use This in Code?

You set the values for vx, vy, and ω based on what you want the robot to do. The code calculates the wheel speeds and sends them to the motors. If you change vx, vy, or ω, the robot's movement changes instantly.

Would you like to see a simple code example for Teensy 4.1, or do you want to try setting vx, vy, and ω yourself and see what happens? Let me know how you want to continue!

⁂

1. https://gjeta.com/sites/default/files/GJETA-2023-0202.pdf
2. https://modernrobotics.northwestern.edu/nu-gm-book-resource/13-2-omnidirectional-wheeled-mobile-robots-part-1-of-2/
3. https://www.roboticsbook.org/S42_logistics_actions.html
4. https://gregwar.github.io/omnidirectional-wheeled-robots
5. https://ijiset.com/vol2/v2s11/IJISET_V2_I11_84.pdf
6. https://journal.umy.ac.id/index.php/jrc/article/download/8889/5613/33638
7. https://www.sciencedirect.com/science/article/abs/pii/S1474667016335807