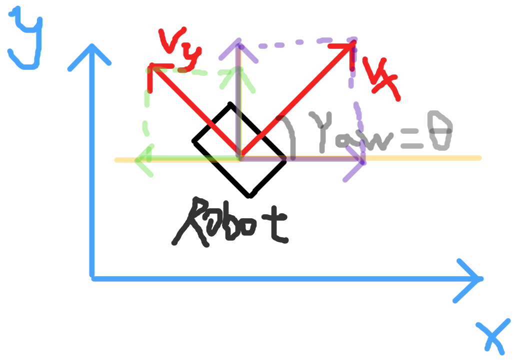
✅ 1. System Model (Vehicle Kinematics)

📌 Control Inputs (in body frame):

- v_x: forward velocity
- v_y: lateral velocity (should be minimized)
- ω: angular (yaw) velocity



📌 State Variables (in global/world frame):

- x: position (X)
- y: position (Y)
- θ: heading angle (yaw)

📌 Discrete-Time Kinematic Model:

The state at the next time step is predicted as:

$$
\begin{aligned}
x_{k+1} &= x_k + dt \cdot (v_{x,k} \cdot \cos\theta_k - v_{y,k} \cdot \sin\theta_k) \\
y_{k+1} &= y_k + dt \cdot (v_{x,k} \cdot \sin\theta_k + v_{y,k} \cdot \cos\theta_k) \\
\theta_{k+1} &= \theta_k + dt \cdot \omega_k
\end{aligned}
$$

This model transforms control inputs from the robot's local frame to the world frame.

✅ 2. MPC Cost Function (Objective)

At each control step, the MPC solves an optimization problem over a prediction horizon N. The cost function penalizes:

- Position tracking error
- Heading alignment error
- Control effort

📌 Full Cost Function:

$$
J = \sum_{k=0}^{N-1} \left[Q_x \cdot (x_k - x_k^{ref})^2 + Q_y \cdot (y_k - y_k^{ref})^2 + Q_\theta \cdot \left(\text{atan2}\left(\sin(\theta_k - \theta_k^{ref}), \cos(\theta_k - \theta_k^{ref}) \right) \right)^2 + R_{vx} \cdot v_{x,k}^2 + R_{vy} \cdot v_{y,k}^2 + R_\omega \cdot \omega_k^2 \right]
$$

Where:

- (x_k^{ref}, y_k^{ref}): reference point at step k
- θ_k^{ref}: reference heading direction computed as:

$$
\theta_k^{ref} = \text{atan2}(y_{k+1}^{ref} - y_k^{ref},\ x_{k+1}^{ref} - x_k^{ref})
$$

- The heading error is wrapped via $\text{atan2}(\sin(\Delta), \cos(\Delta))$ to ensure smooth rotation (avoid 360° spins).

✅ 3. MPC Constraints

🔸 Dynamics Constraints:

$$X_{k+1} = f(X_k, U_k) \quad \text{(using the kinematic model)}$$

🔸 Control Bounds:

$$\begin{aligned} 0 \leq v_x \leq 2.0 \quad \text{(no reverse)} \\ -2.0 \leq v_y \leq 2.0 \\ -2.0 \leq \omega \leq 2.0 \end{aligned}$$

🔸 Initial Condition:

$$X_0 = \text{current state}$$

✅ 4. MPC Execution Strategy (Receding Horizon)

At every control step:

1. Identify the closest point on the reference trajectory;
2. Extract $N + 1$ reference points ahead;
3. Compute reference heading angles;
4. Set up the optimization problem with:
   - Cost function (errors + control penalties)
   - Dynamics constraints
   - Input bounds
5. Solve using IPOPT via CasADi;
6. Apply only the **first step** of the optimal control sequence;
7. Repeat the process at the next time step.

✅ 5. Key Design Decisions in Your Code

- **Heading alignment penalty** (Q_θ) is high → enforces face-forward movement;
- **Lateral speed penalty** (R_{vy}) is high → suppresses sliding motion;
- **No reverse constraint**: $v_x \geq 0$;
- **Wrapped angle error** avoids long rotations.