Topic 10

Coordinate Spaces and Transforms

It depends on your perspective

Topic 10

Coordinate Spaces and Transforms

It depends on your perspective

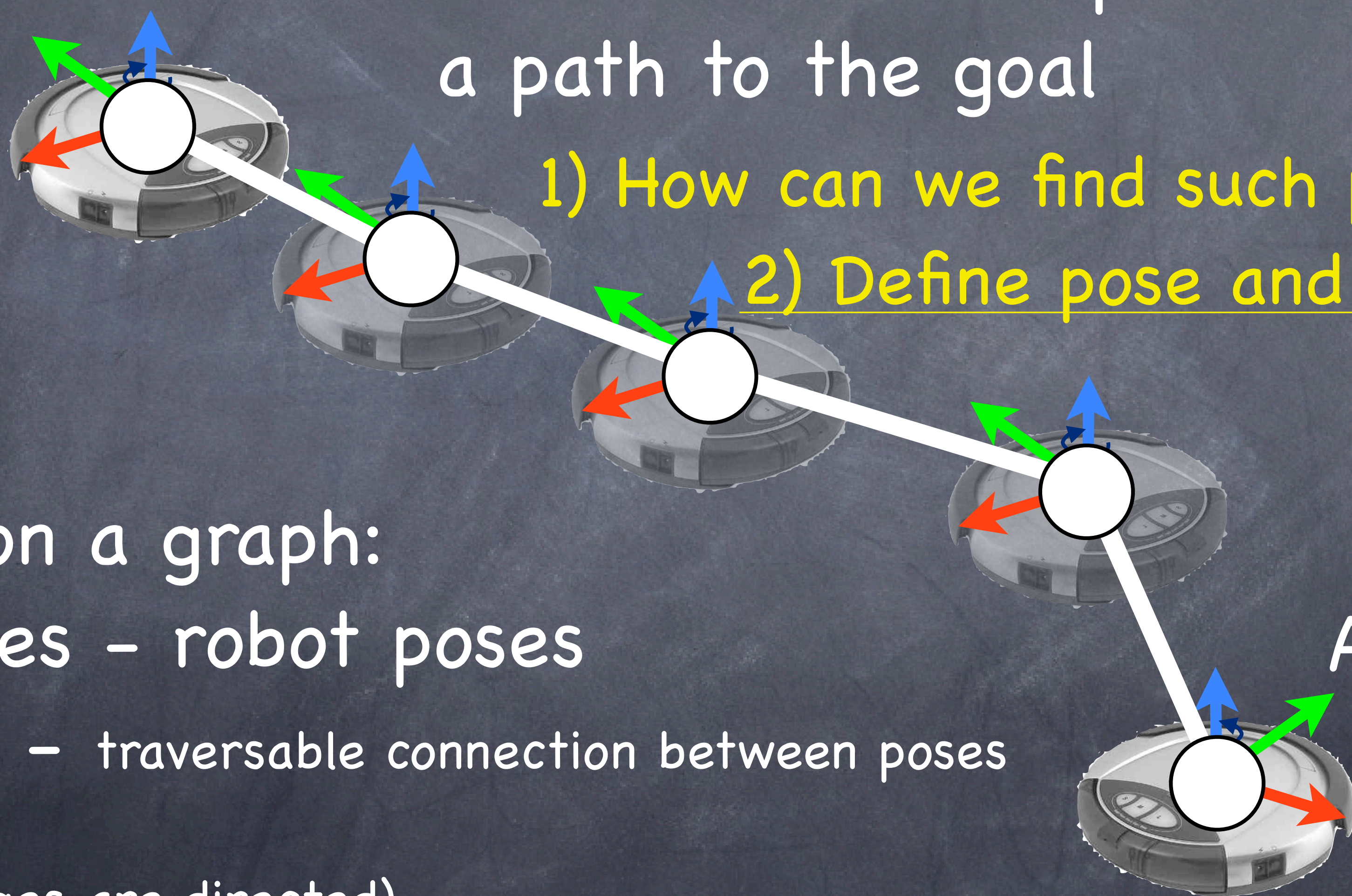
Path Planning

B: Goal

Find intermediate poses forming a path to the goal

1) How can we find such paths?

2) <u>Define pose and controls?</u>



A: Start

Path on a graph:

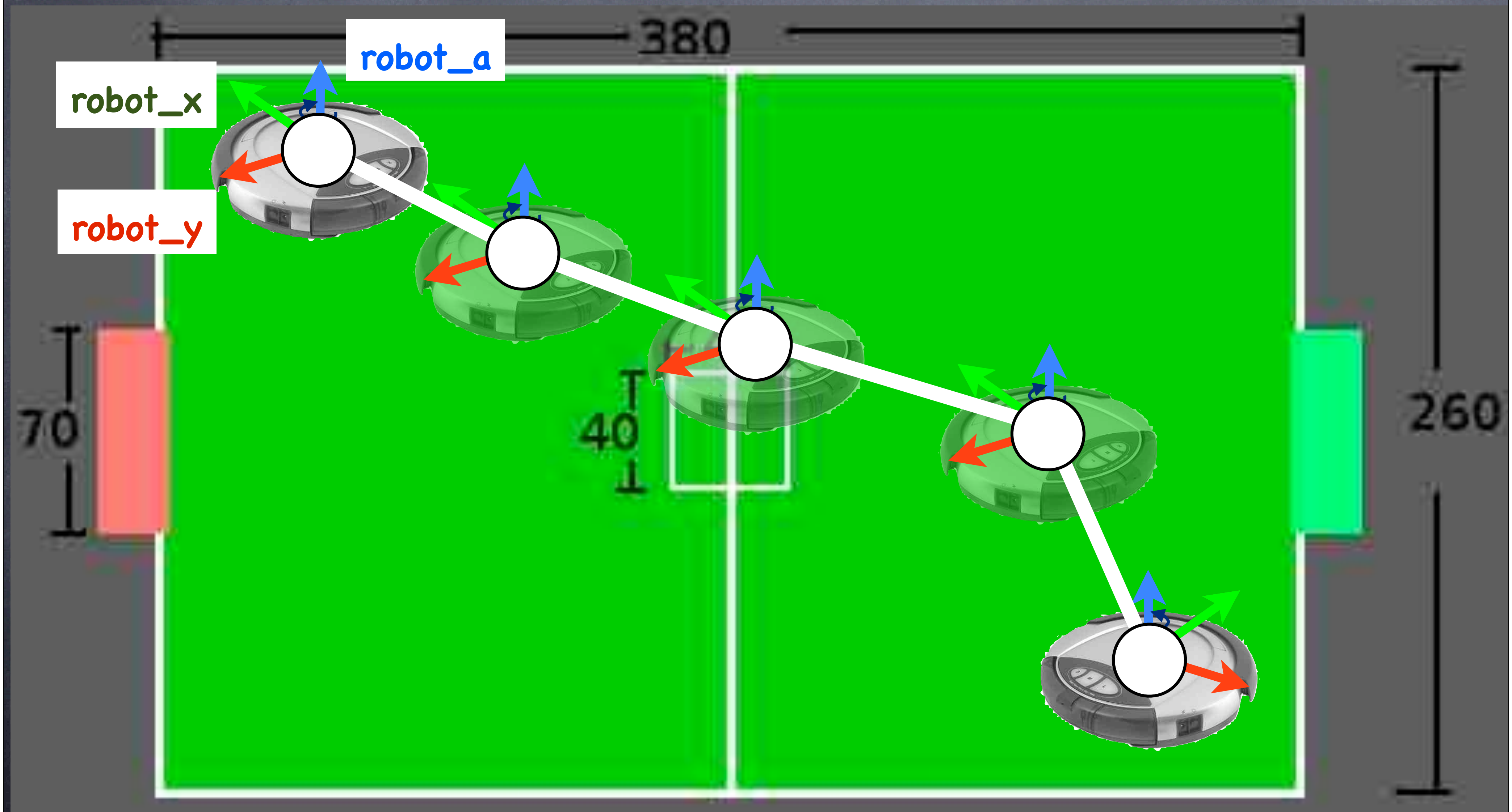
vertices - robot poses

edges - traversable connection between poses

(note edges are directed)

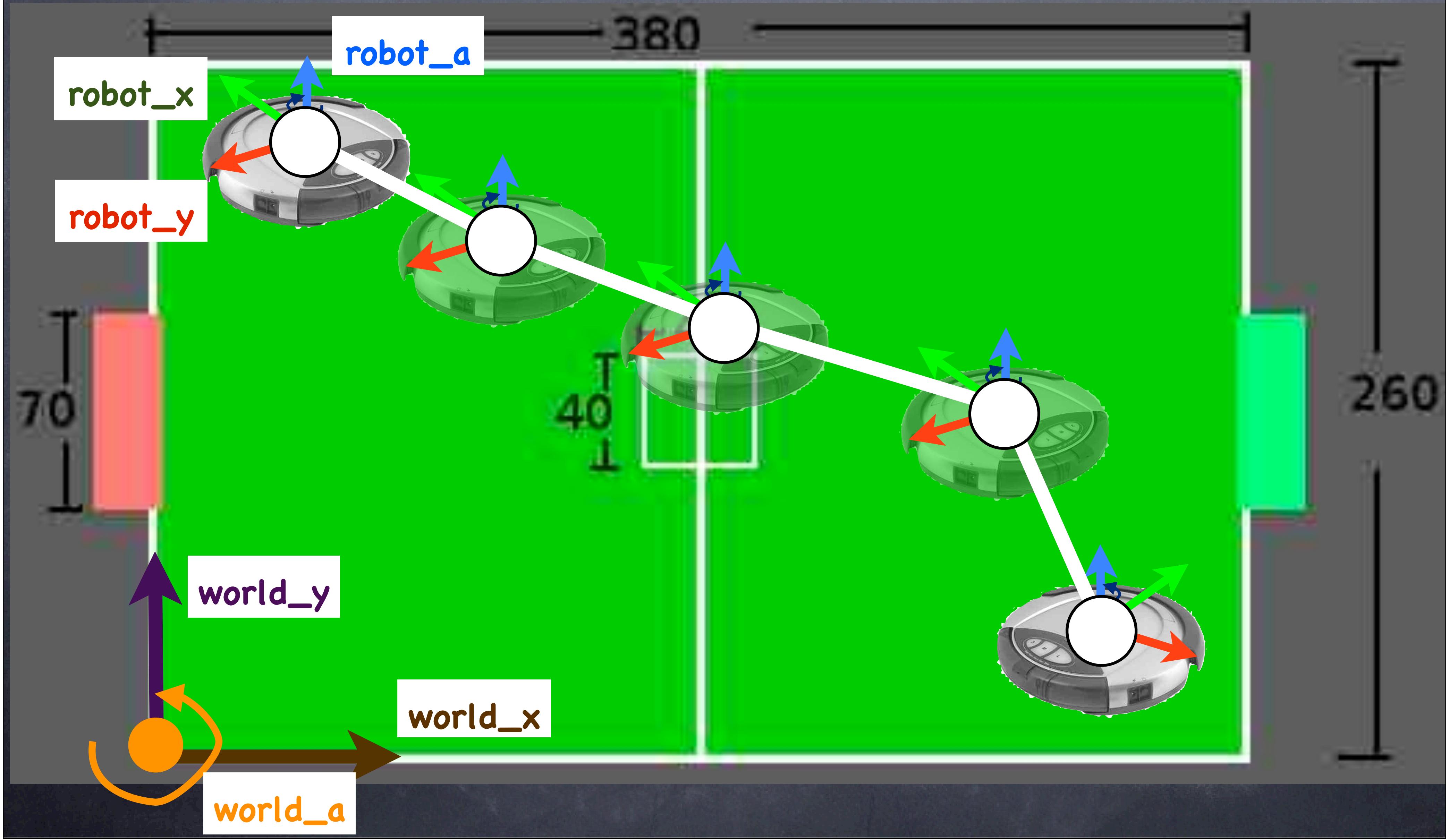
1) path planning: sequence of intermediate poses

2) Define pose and controls?



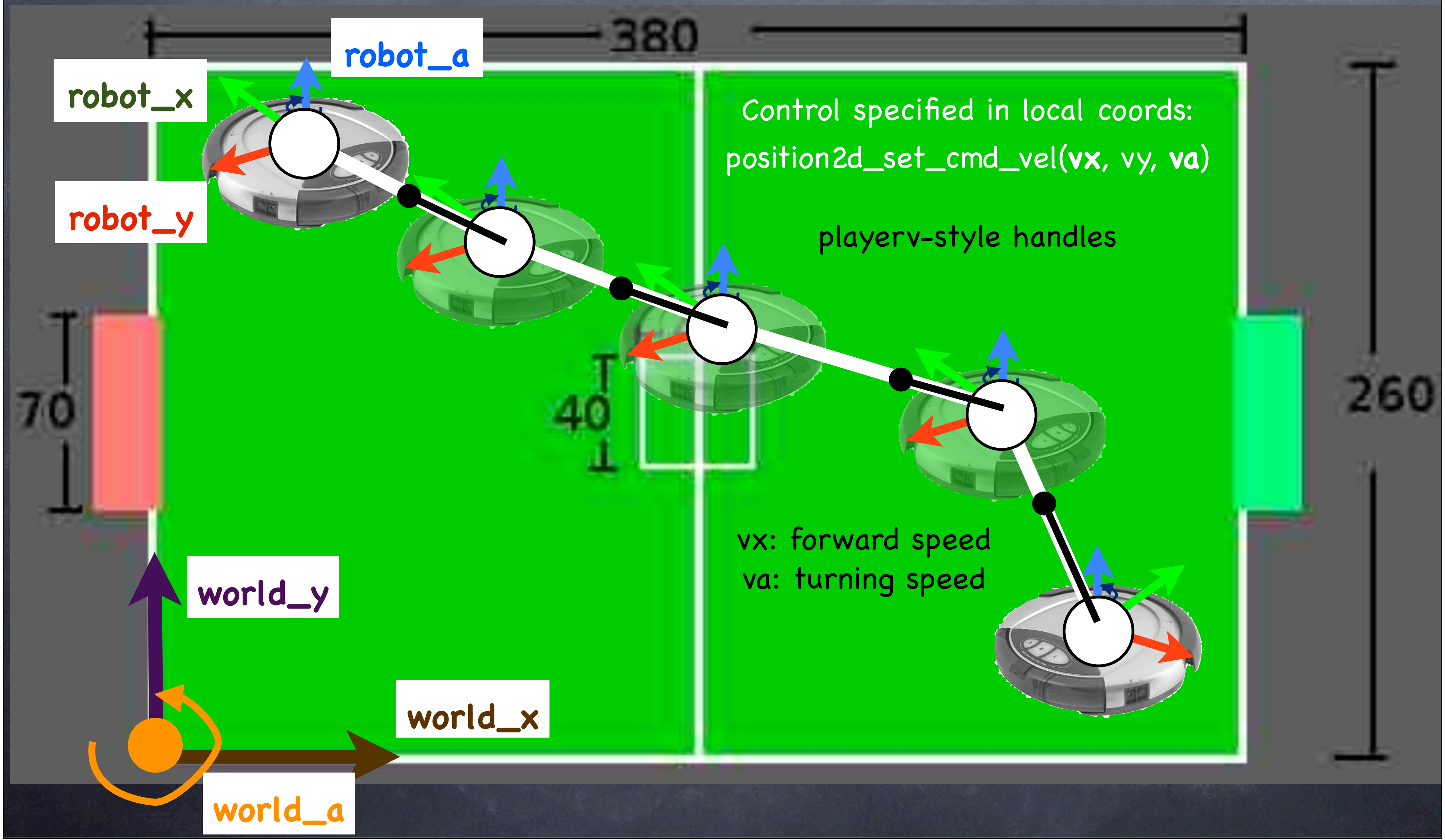
1) path planning: sequence of intermediate poses

2) Define pose (3DOF position/orientation on field) and controls?



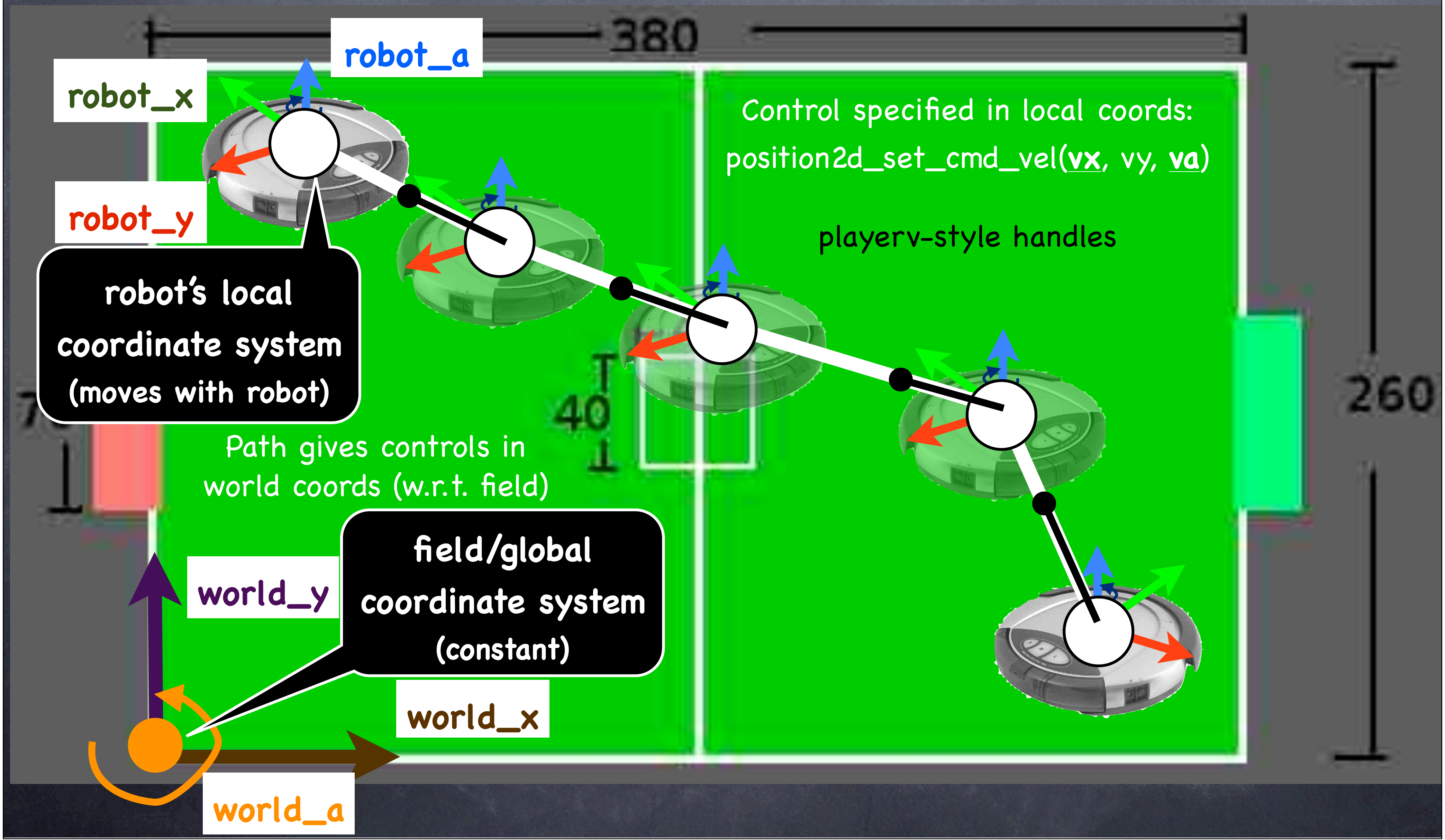
1) path planning: sequence of intermediate poses

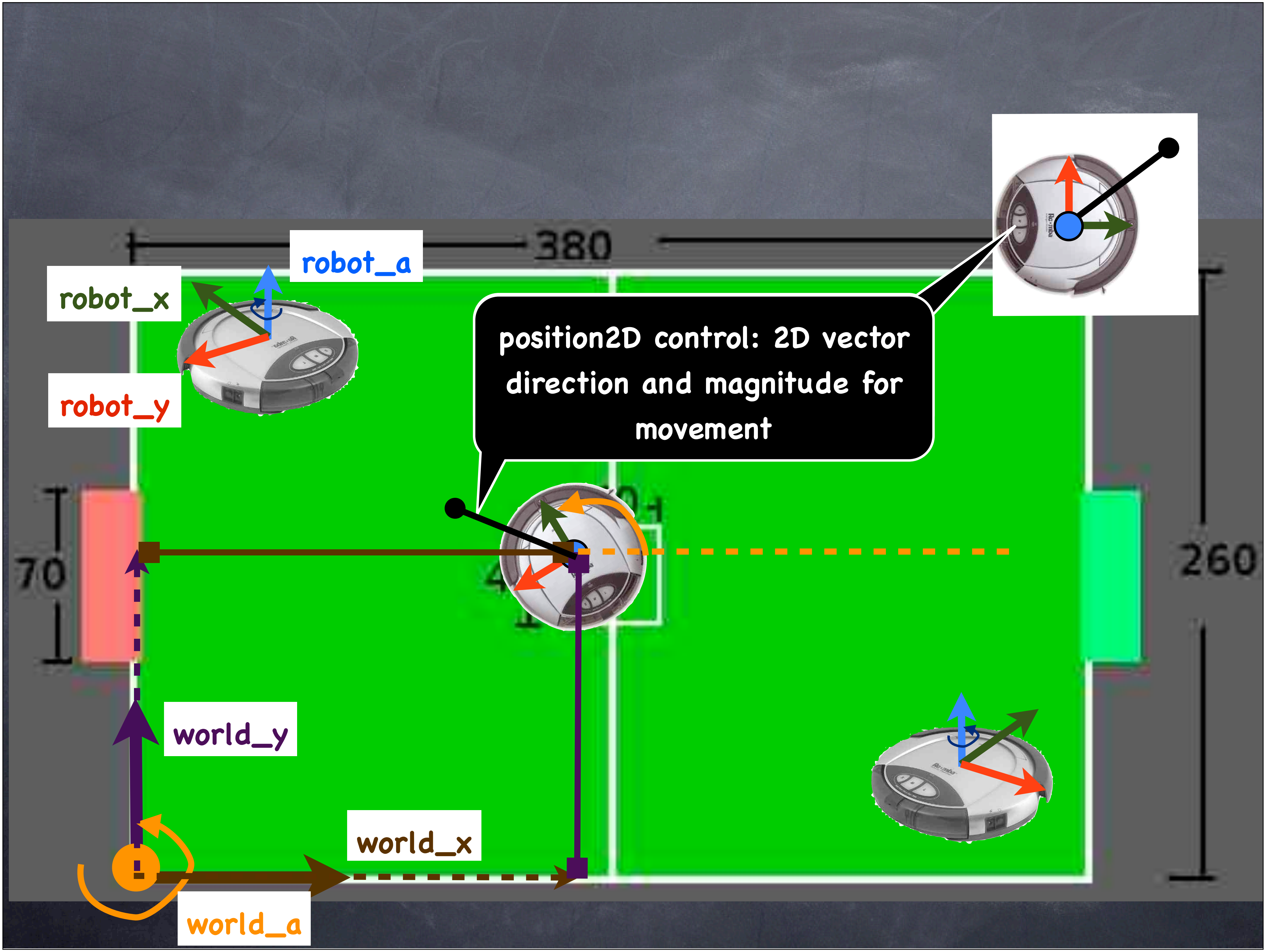
2) Define pose (3DOF position/orientation on field) and controls?



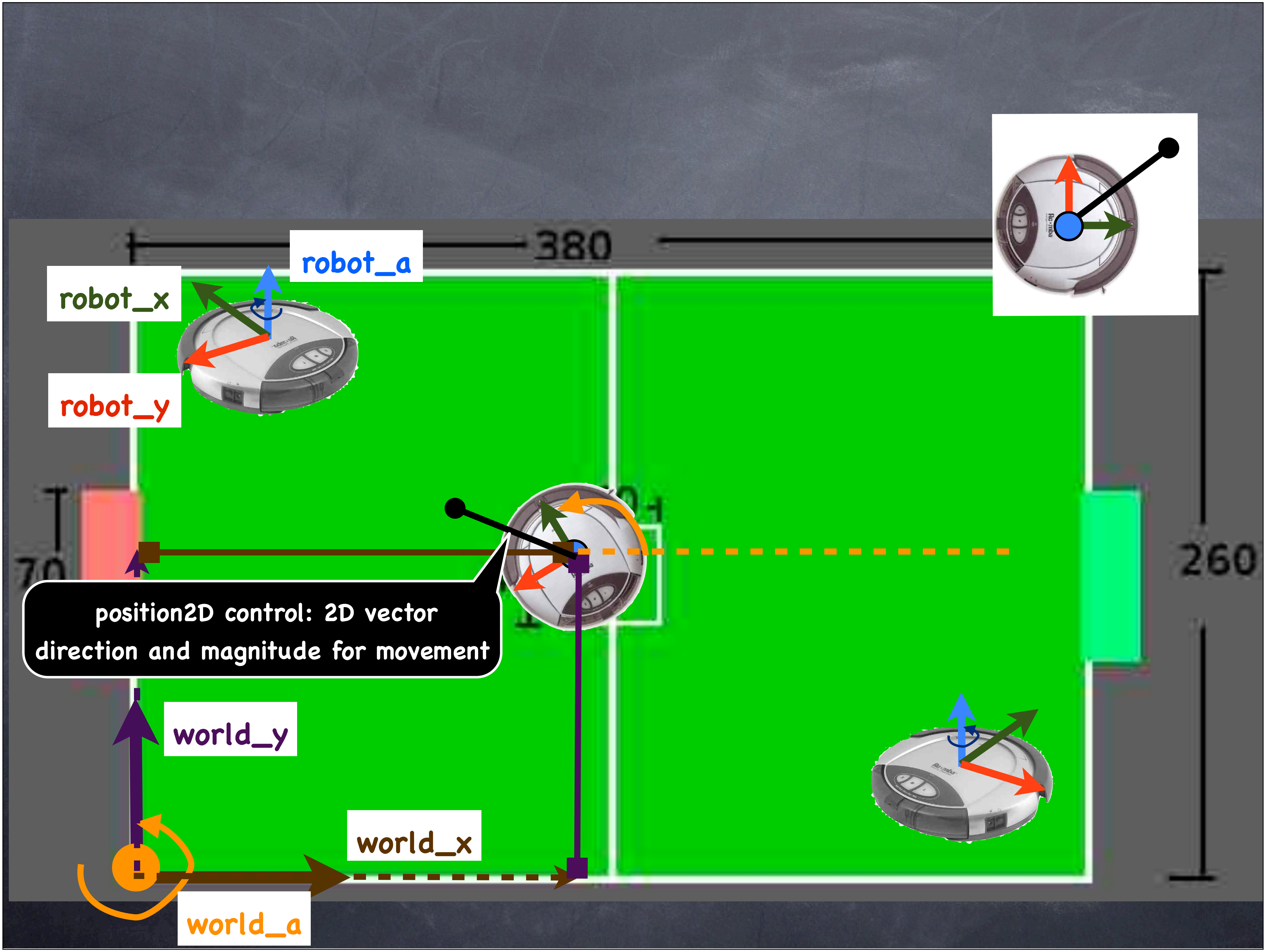
1) path planning: sequence of intermediate poses

2) Define pose (3DOF position/orientation on field) and controls (transform global control into robot coordinates)?

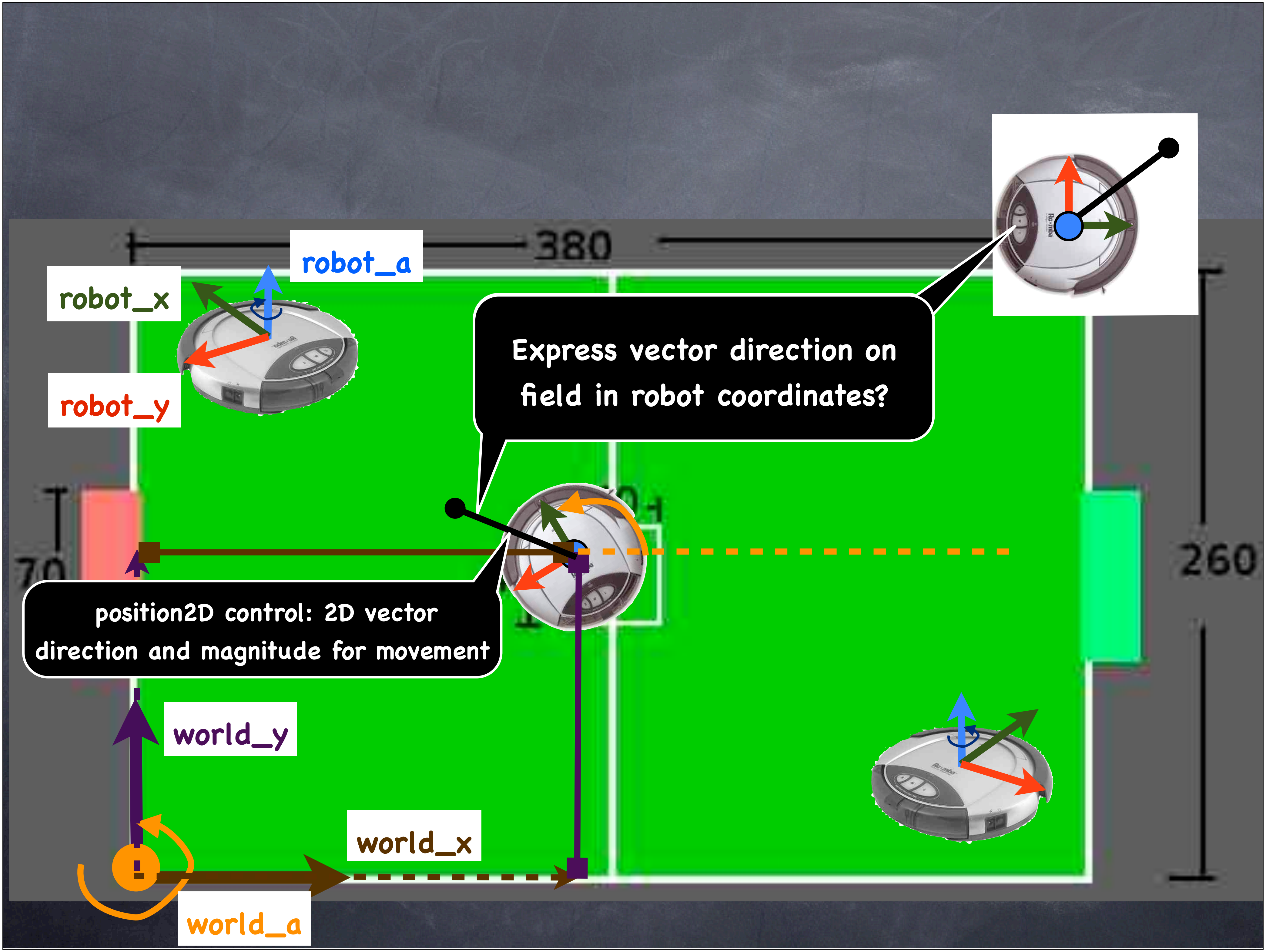




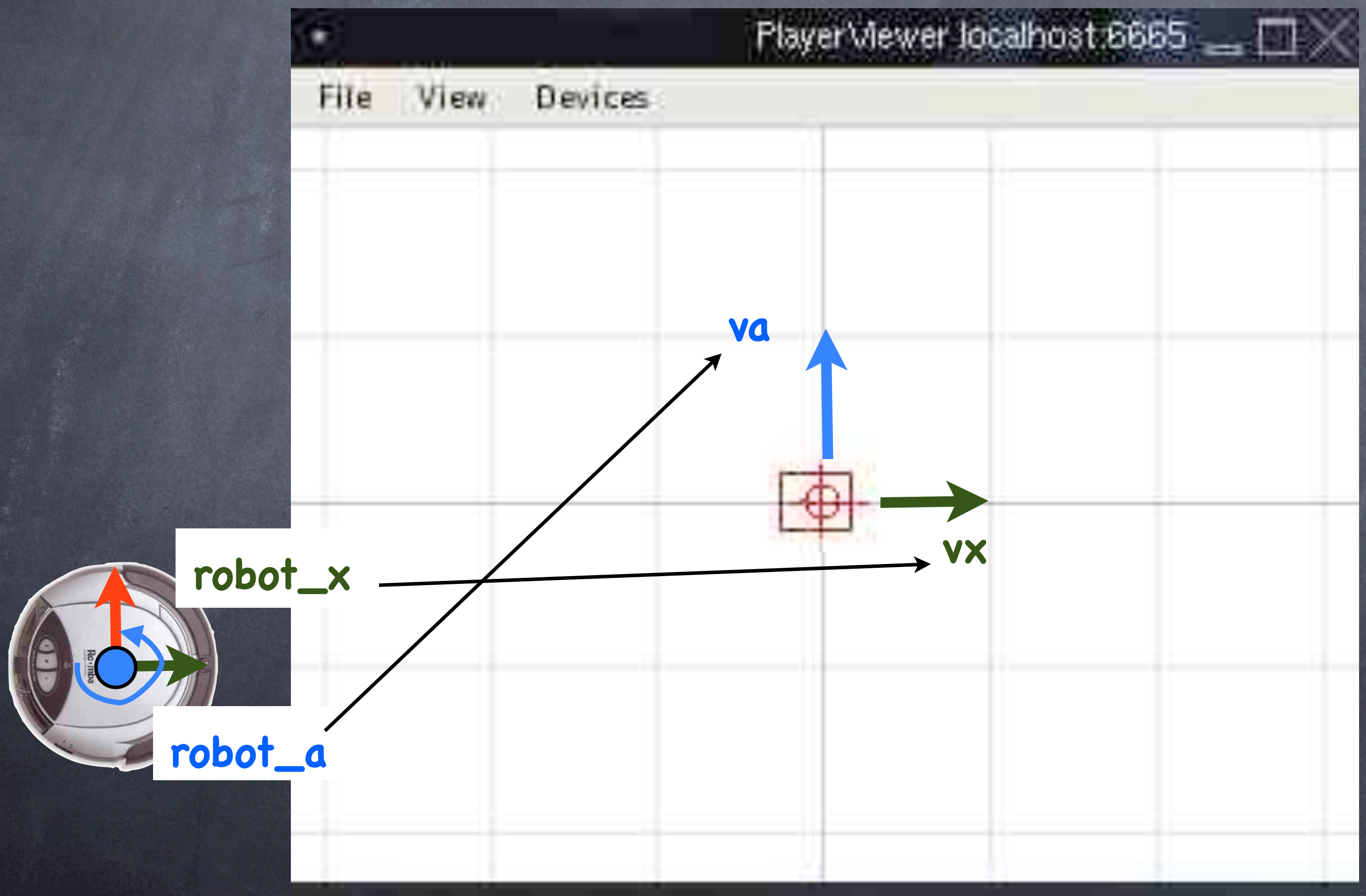
380
robot_a
robot_x
robot_y
position2D control: 2D vector direction and magnitude for movement
70
260
world_y
world_x
world_a



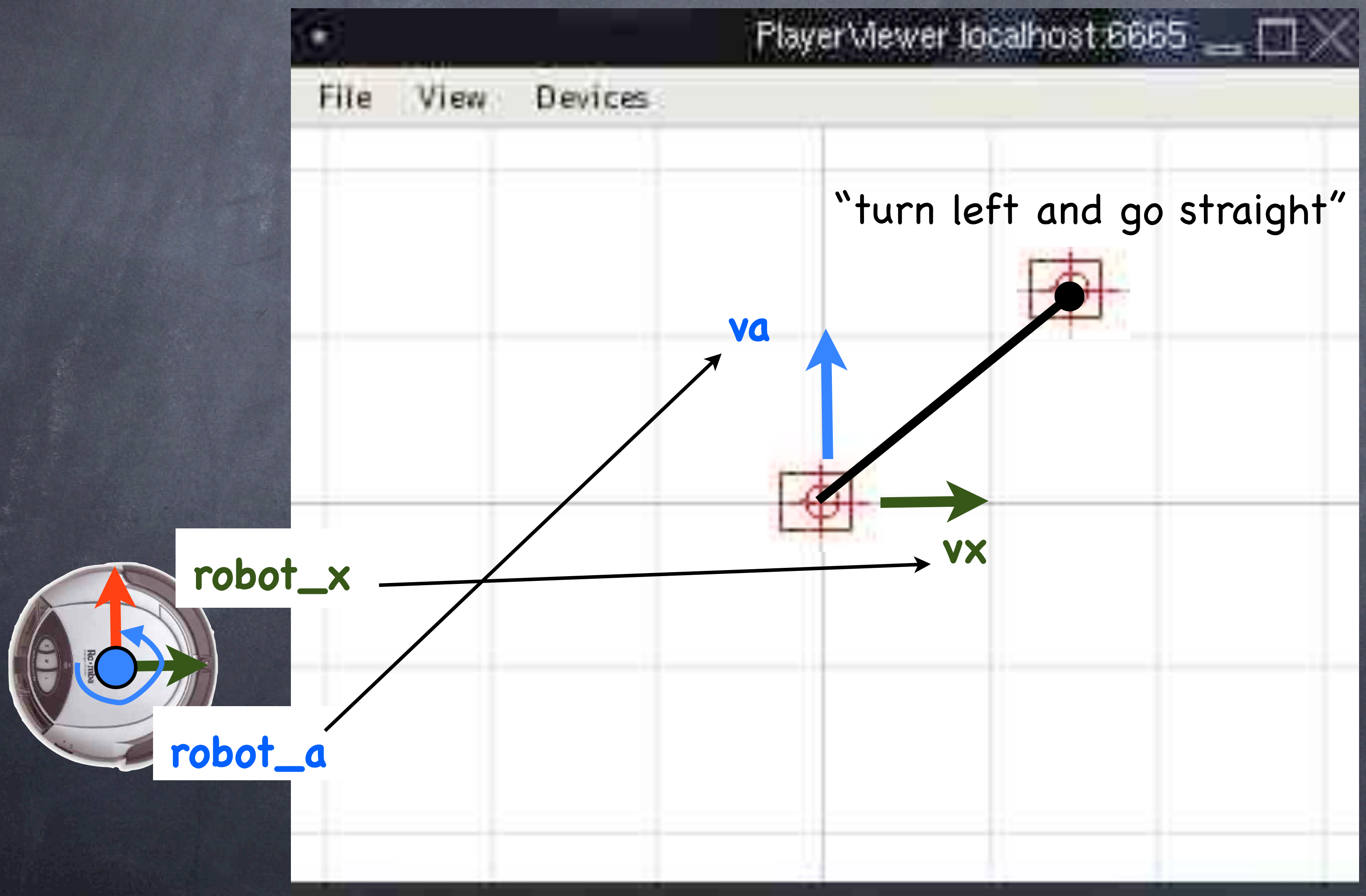
380
robot_a
robot_x
robot_y
70
260
position2D control: 2D vector direction and magnitude for movement
world_y
world_x
world_a


380
robot_a
robot_x
robot_y
Express vector direction on field in robot coordinates?
position2D control: 2D vector direction and magnitude for movement
70
260
world_y
world_x
world_a

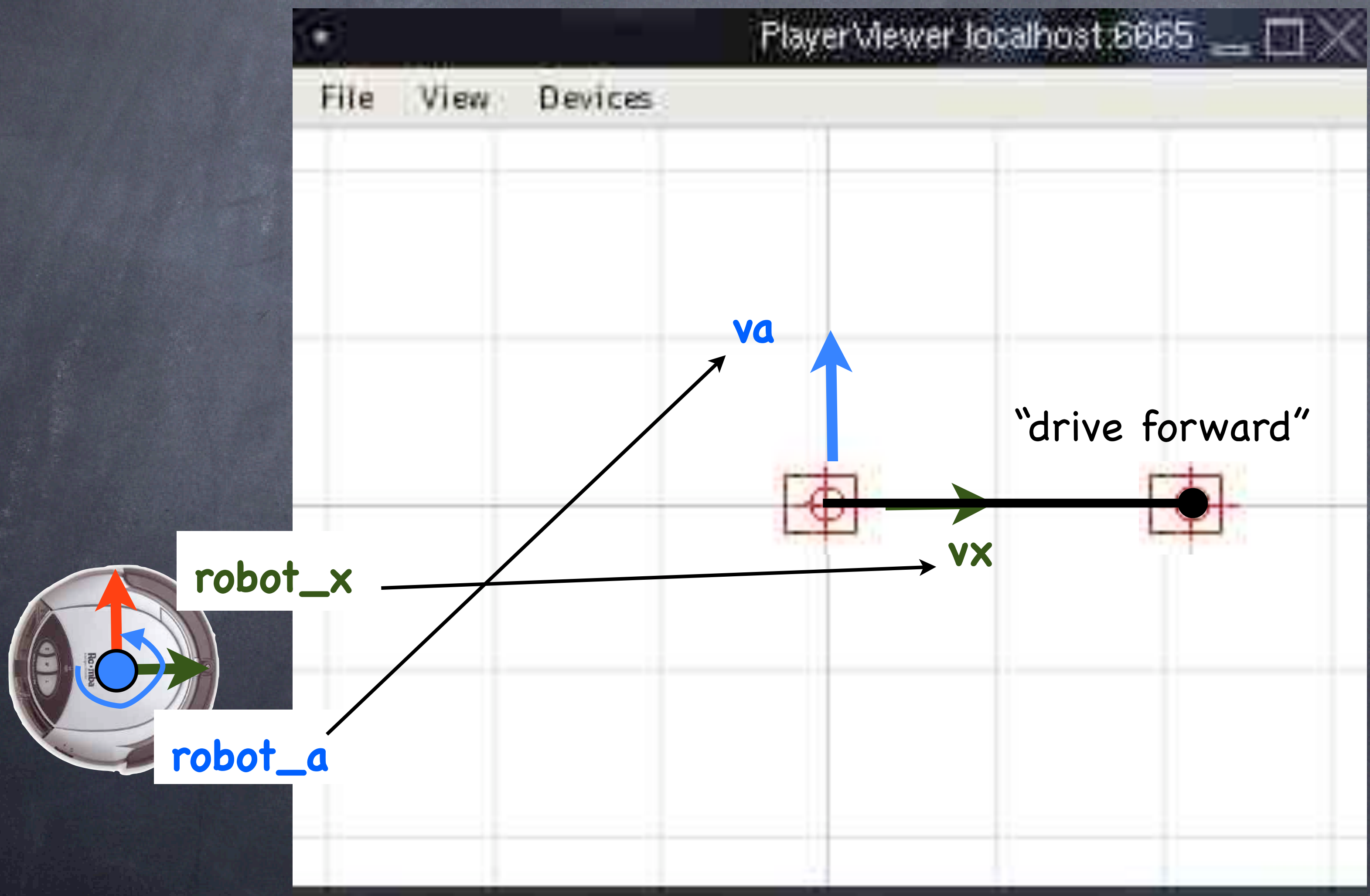
position2d in playerv



position2d in playerv

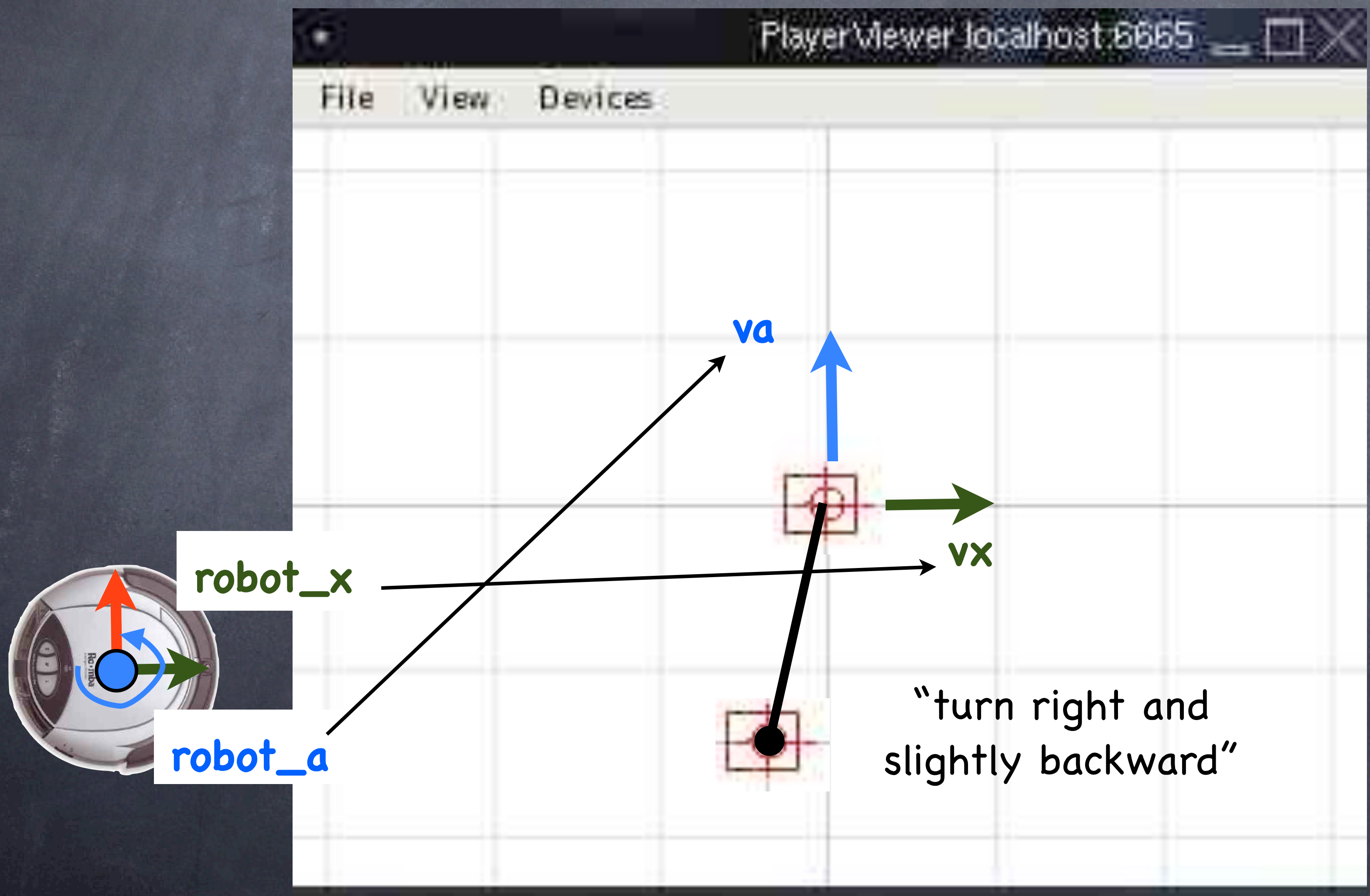


position2d in playerv



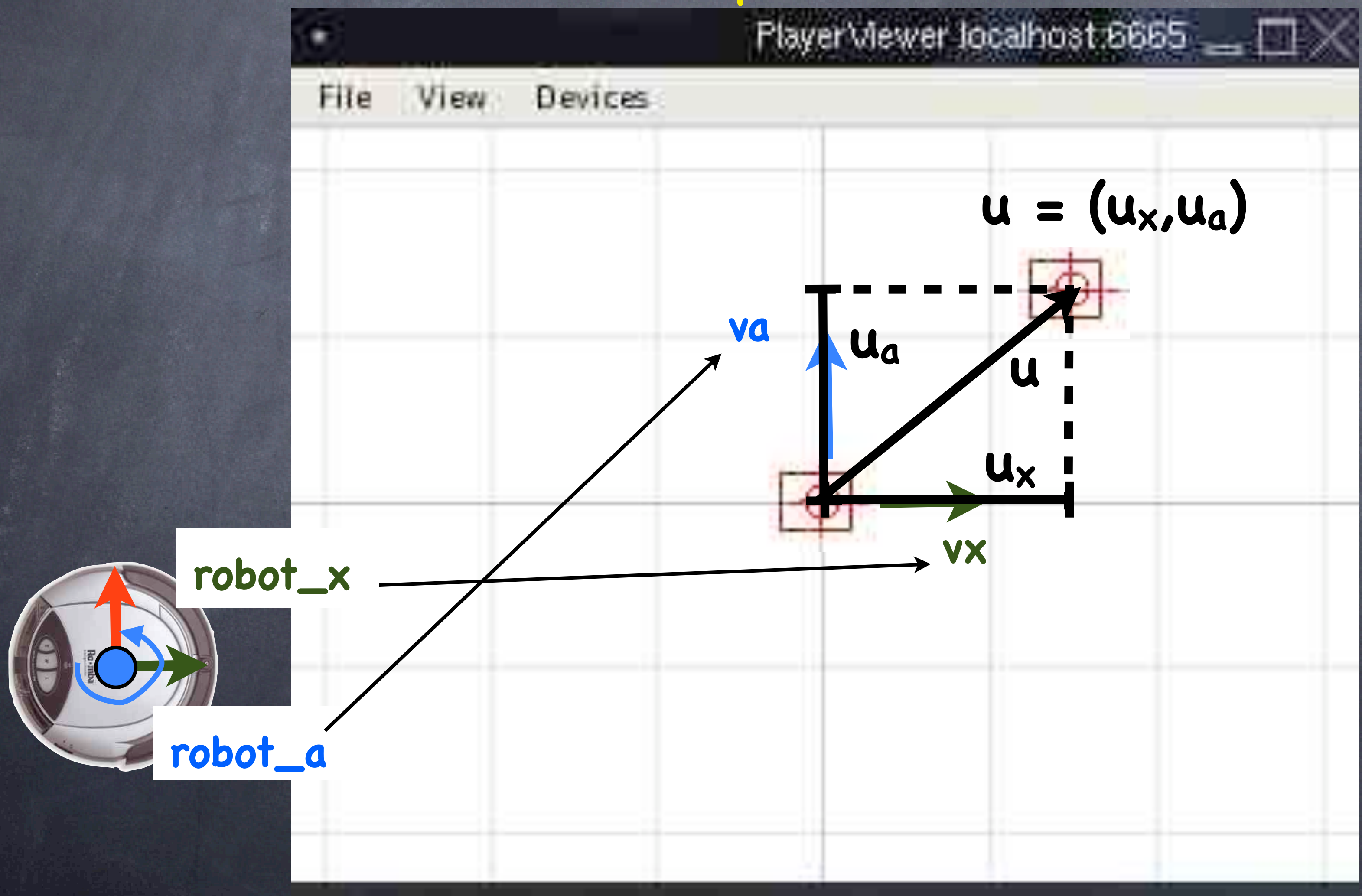
position2d in playerv

Can we be more precise?



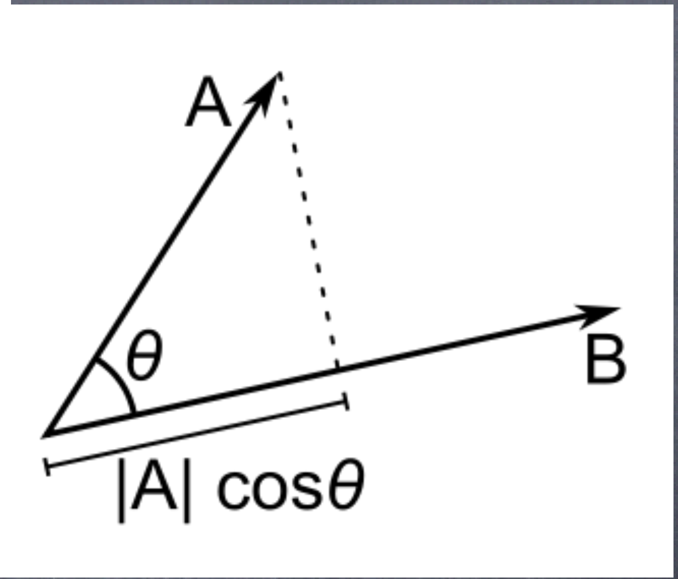
position2d in playerv

Note: u_a is dot product of u and va



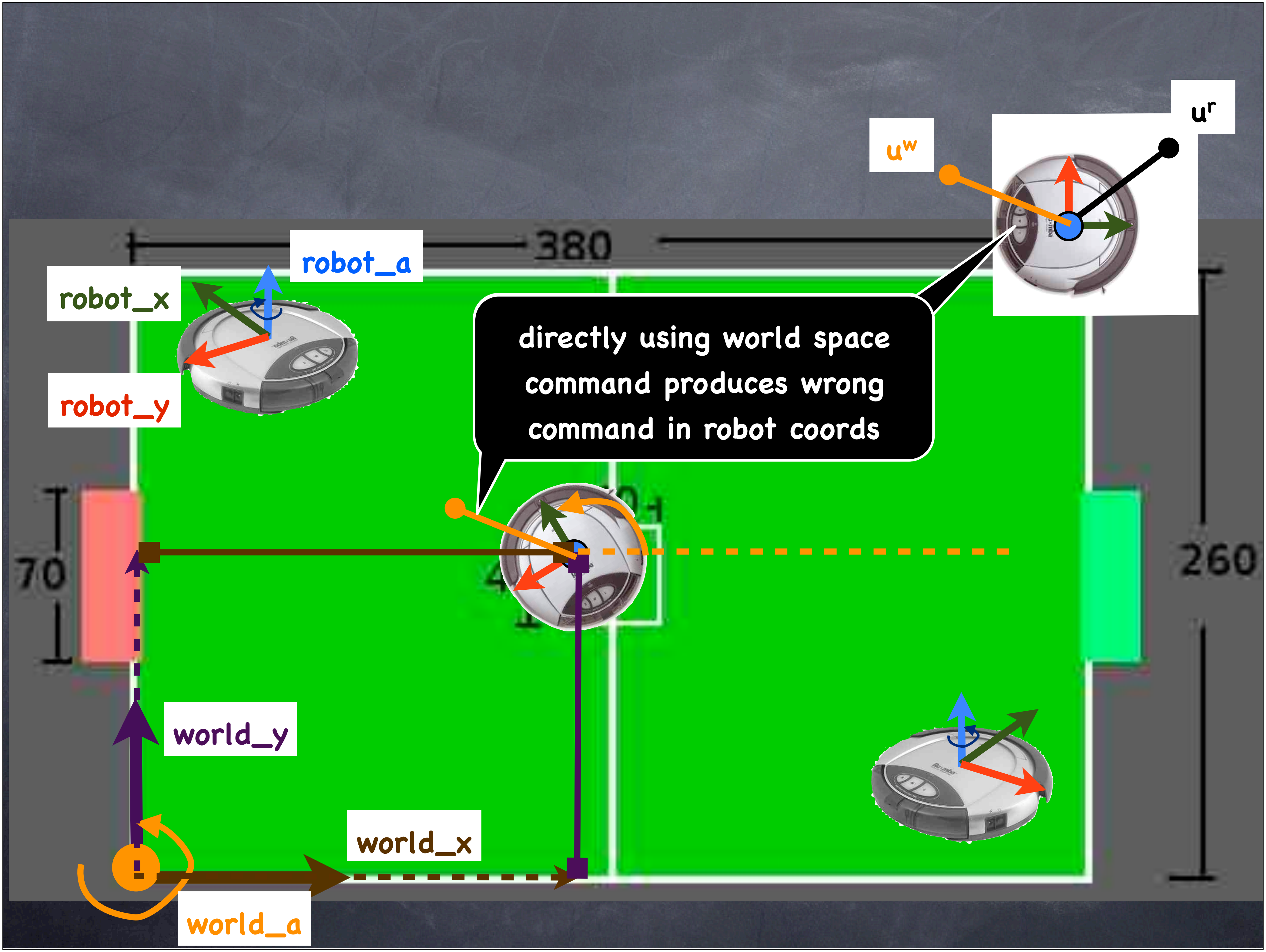
Dot Product

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$



- If one is a unit vector (length 1), the dot product is the projection onto this vector
- Related to angle between vectors

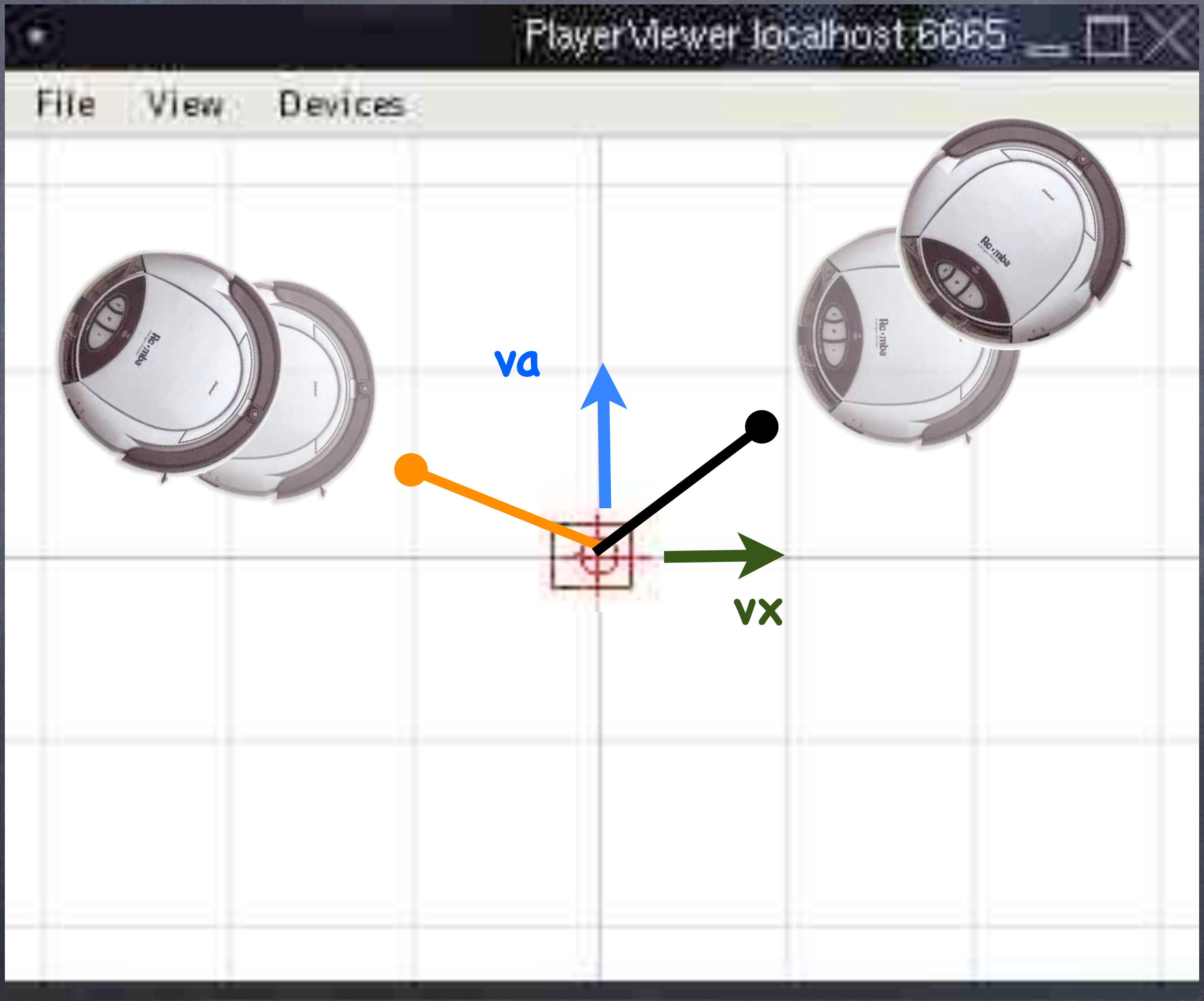
$$\theta = \arccos\left(\frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|}\right).$$



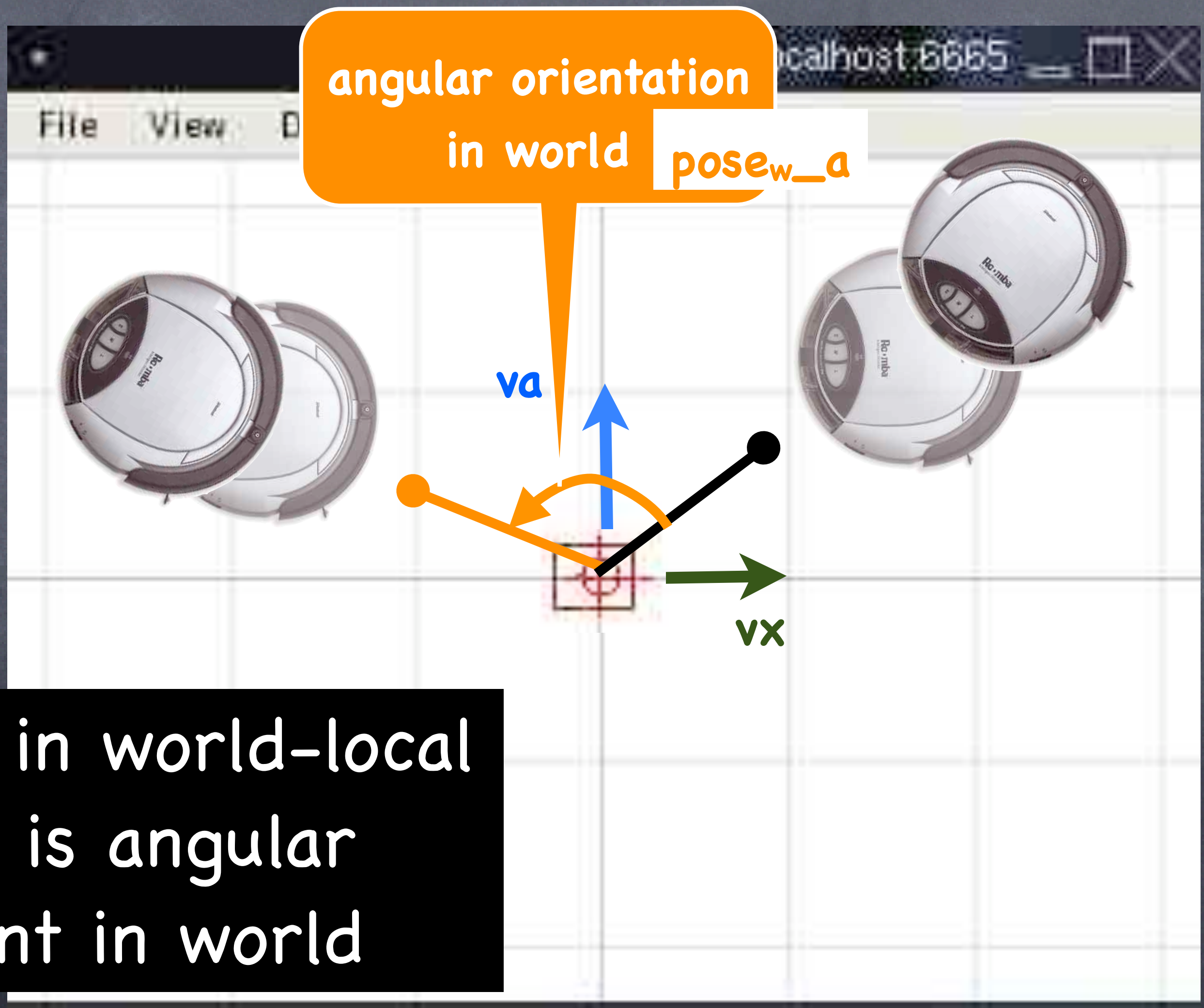
u^w
u^r
robot_a
robot_x
robot_y
380
directly using world space command produces wrong command in robot coords
70
260
world_y
world_x
world_a

position2d in playerv

How can we transform control vector into robot coords?

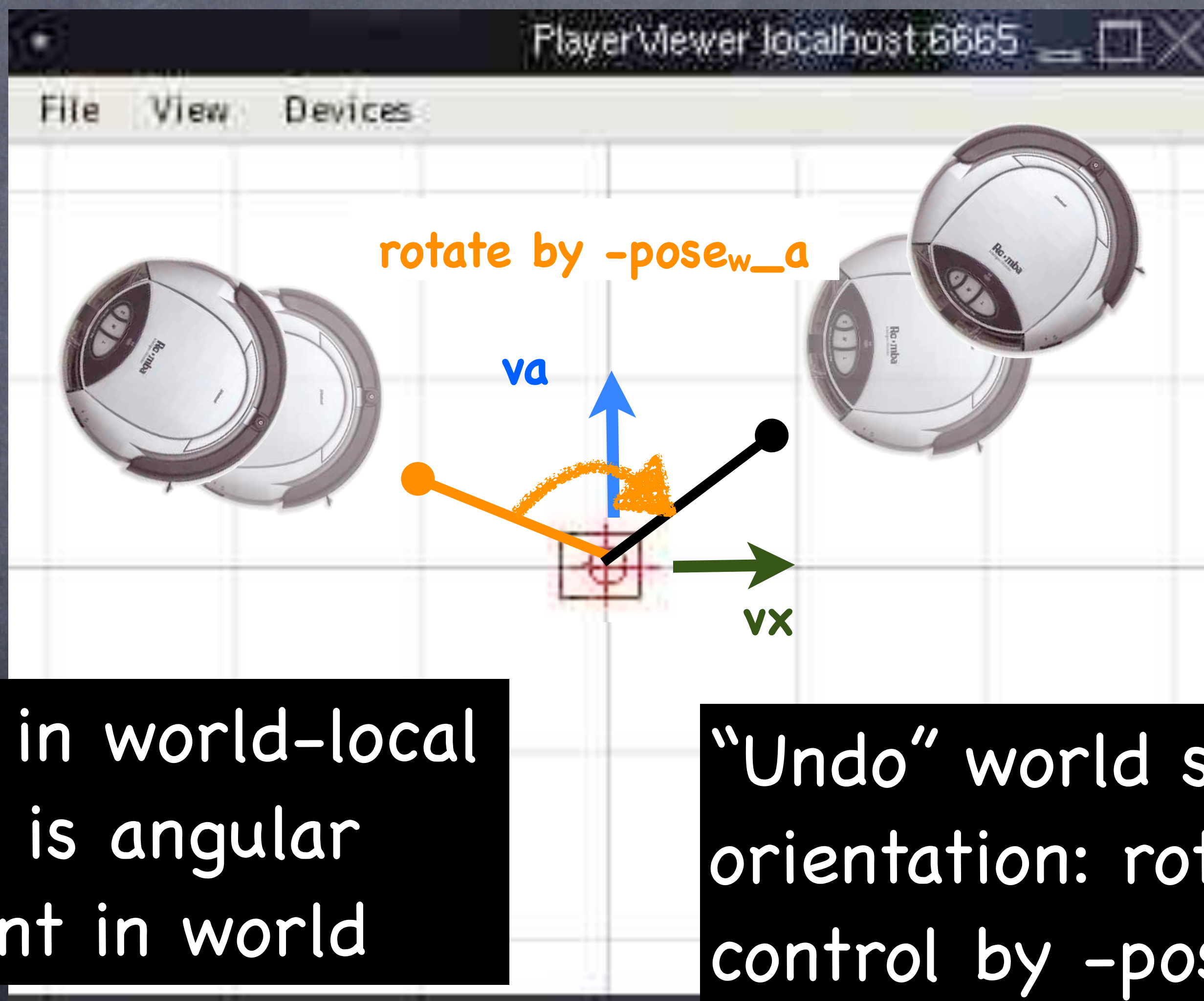


position2d in playerv



Difference in world-local orientation is angular displacement in world

position2d in playerv

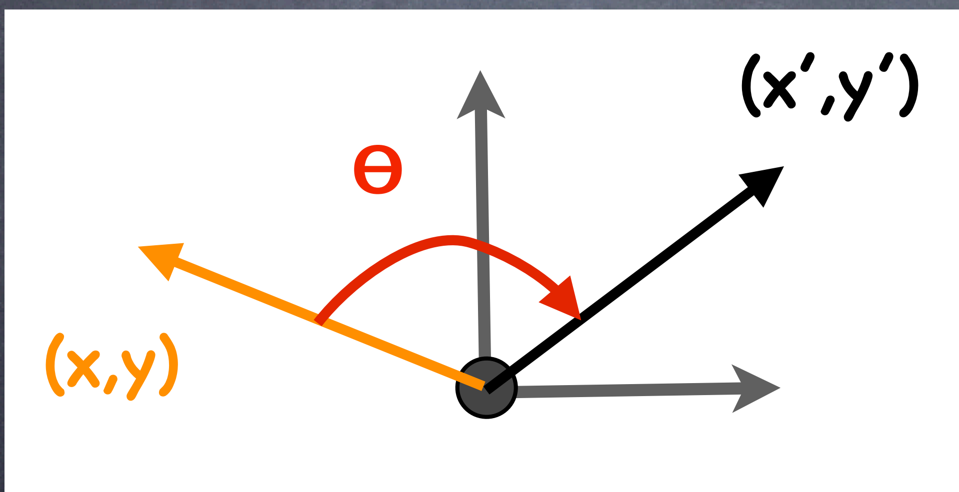


Difference in world-local orientation is angular displacement in world

"Undo" world space orientation: rotate control by -pose$_w$_a

Rotating a 2D vector

(counterclockwise)



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Matrix multiply vector by 2D rotation matrix
- Matrix parameterized by rotation angle
- Check matrix correctness yourself

Matrix multiplication

$$\underbrace{\begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 2 & 3 & 4 \end{bmatrix}}_{3\times 4 \text{ matrix}} \underbrace{\begin{bmatrix} \cdot & \cdot & \cdot & a & \cdot \\ \cdot & \cdot & \cdot & b & \cdot \\ \cdot & \cdot & \cdot & c & \cdot \\ \cdot & \cdot & \cdot & d & \cdot \end{bmatrix}}_{4\times 5 \text{ matrix}} = \underbrace{\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & x_{3,4} & \cdot \end{bmatrix}}_{3\times 5 \text{ matrix}}$$

$$\begin{aligned} x_{3,4} &= (1, 2, 3, 4) \cdot (a, b, c, d) \\ &= 1 \times a + 2 \times b + 3 \times c + 4 \times d. \end{aligned}$$