



U2.2 The rotation matrix

Name _____

Class _____

Date _____

What this lesson is about

Turning a vector from one frame into the other, in two dimensions, by hand and in code.

- From body to world
- In code
- Checking it without trusting it
- The other way

Questions

6 marks in all

1. A robot at heading 60 measures (0, 10) in its own frame. What does this print?

[1 mark]

```
import math

def body_to_world(vx, vy, h_deg):
    a = math.radians(h_deg)
    return (vx * math.cos(a) + vy * math.sin(a),
            -vx * math.sin(a) + vy * math.cos(a))

wx, wy = body_to_world(0, 10, 60)
print("wx:", round(wx, 2))
print("wy:", round(wy, 2))
```

2. A robot at heading 90 slides 10 cm to its own right, the body vector (10, 0). What does this print?

[1 mark]

```
import math
a = math.radians(90)
vx, vy = 10.0, 0.0
wx = vx * math.cos(a) + vy * math.sin(a)
wy = -vx * math.sin(a) + vy * math.cos(a)
print(round(wx, 2), round(wy, 2))
```

3. A student computes $10 * \mathbf{math.sin(90)}$ for a robot facing +x and gets 8.94 instead of 10. What is wrong? [1 mark]

- ☐ A $\mathbf{math.sin}$ takes radians, so 90 was treated as 90 radians; it needs $\mathbf{math.radians(90)}$
- ☐ B The x component should use $\mathbf{cos}$, not $\mathbf{sin}$
- ☐ C The minus sign belongs on this term in a clockwise convention
- ☐ D The vector is in the world frame and needs rotating back first

4. Which matrix undoes $R(h)$, turning a world vector back into the body frame? [1 mark]
- ☐ A $R(-h)$, which is also the transpose of $R(h)$
 - ☐ B $R(h)$ itself, applied a second time
 - ☐ C $R(h + 180)$
 - ☐ D The matrix with each entry of $R(h)$ replaced by its reciprocal
5. In $R(h)$ as the lesson writes it, the minus sign sits on $-\sin h$ below the diagonal. Why there? [1 mark]
- ☐ A Because heading is clockwise positive; in an anticlockwise convention it sits above the diagonal
 - ☐ B Because Python's trigonometric functions work in radians
 - ☐ C Because the body's y axis points forward rather than right
 - ☐ D Because a rotation matrix always has its minus sign below the diagonal
6. This rotates the body vector (3, 4) into the world at heading 37 and back again. What does it print? [1 mark]

```
import math

def body_to_world(vx, vy, h):
    a = math.radians(h)
    return vx * math.cos(a) + vy * math.sin(a), -vx * math.sin(a) + vy * math.cos(a)

def world_to_body(wx, wy, h):
    a = math.radians(h)
    return wx * math.cos(a) - wy * math.sin(a), wx * math.sin(a) + wy * math.cos(a)

wx, wy = body_to_world(3, 4, 37)
bx, by = world_to_body(wx, wy, 37)
print(round(math.hypot(wx, wy), 6), round(bx, 6), round(by, 6))
```

The task: rotate a vector

A robot at heading 60 measures `(0, 10)` in its own frame. Print the same vector in the world frame, as `wx:` and `wy:`, worked out with the rotation matrix.

```
from bugbot import *
import math
connect()

h = 60
vx, vy = 0.0, 10.0
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/u2-2-the-rotation-matrix/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Rotate a vector into the world and back again. Do you get the original to the last decimal place?
2. Print the length of the vector in both frames and confirm it does not change.
3. Write `world_to_body()` and check it against `body_to_world()` for ten random headings.