



WABO SMAC

Các tính năng hỗ trợ cuộc thi SMAC Challenge

Nội dung

- ❑ Marker & kinect
- ❑ Hệ tọa độ
- ❑ Chức năng Get Current Pose
- ❑ Chức năng Relocation
- ❑ Chức năng Moving closer to object
- ❑ Chức năng Moving closer to table

Marker & kinect

Khoảng cách nhận diện Marker

- Min: 40 cm
- Max: 300 cm

Khoảng cách nhận diện Object đặt trên bàn

- Min: 40 cm
- Max: 100 cm

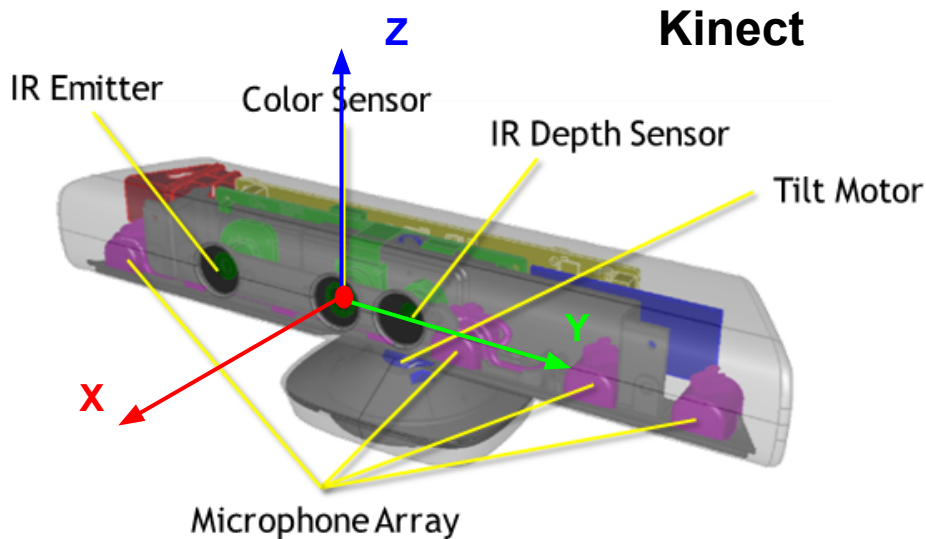
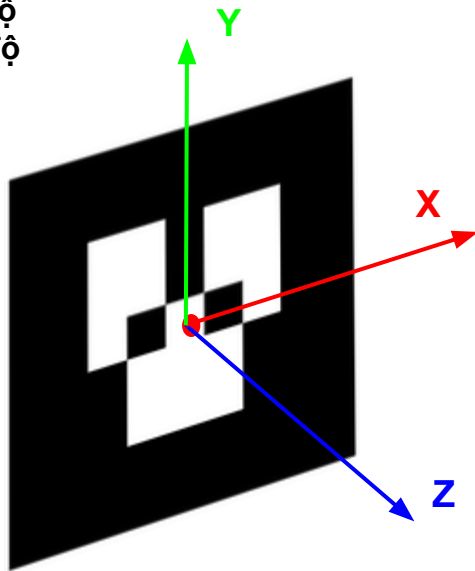
Trường nhìn của kinect

- Chiều dọc: 43 độ
- Chiều ngang: 57 độ

Marker

Các marker (cỡ nhỏ)
gắn với object gọi là
object

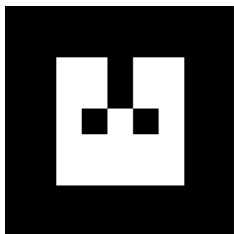
Các marker (cỡ lớn)
gọi là marker



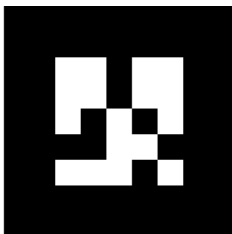
Marker là các hình ảnh dấu hiệu giúp ứng dụng xác định được vị trí cũng như hướng của nó thông qua việc xử lý hình ảnh với camera

Marker & kinect (hệ thống marker)

0



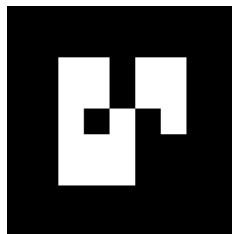
1



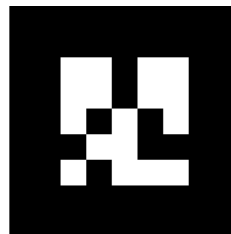
2



9



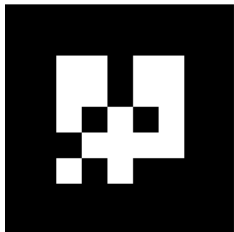
10



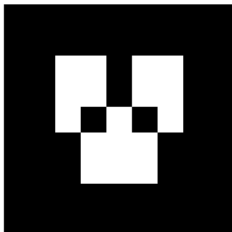
11



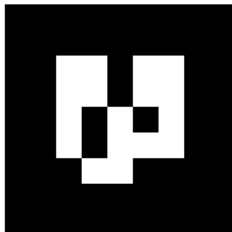
3



4



5



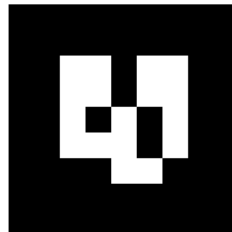
12



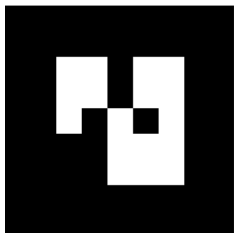
13



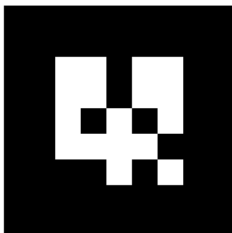
14



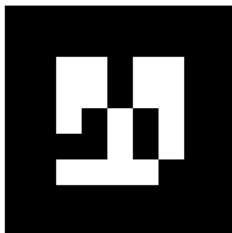
6



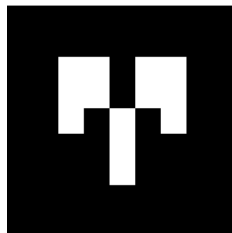
7



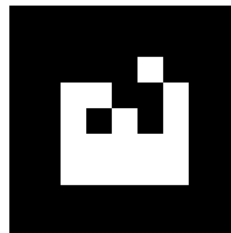
8



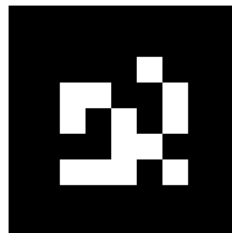
15



16



17



Hệ tọa độ

(Định nghĩa dùng trong SMAC Challenge)

Hệ tọa độ tương đối là hệ tọa độ gắn với các vật thể không cố định

Marker gắn trên object
Robot

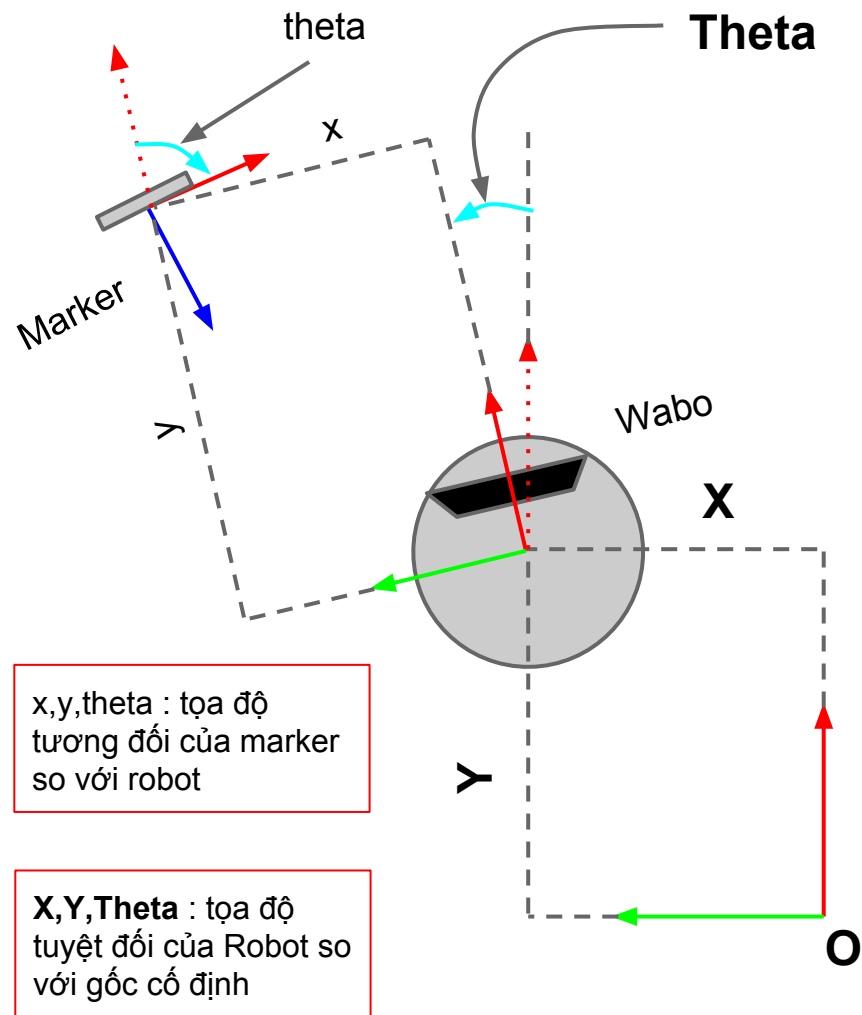
Hệ tọa độ tuyệt đối là hệ tọa độ gắn với gốc cố định trên sân

Được đặt trên mặt sàn tại vị trí bàn giữa sân

VD:

Tọa độ object so với robot được gọi là tọa độ **tương đối** của object so với robot

Tọa độ **tuyệt đối** của object là tọa độ của nó so với gốc cố định trên sân



Hệ tọa độ

(Định nghĩa dùng trong SMAC Challenge)

Quy định về dấu

Các tọa độ về phía dương các trục x, y, z có giá trị dương, ngược lại có giá trị âm - đơn vị: **met**

Các góc bên phải trục làm gốc (trục X ▶) có giá trị âm, bên trái có giá trị dương - đơn vị: **radian**

VD: Như bên hình ta có:

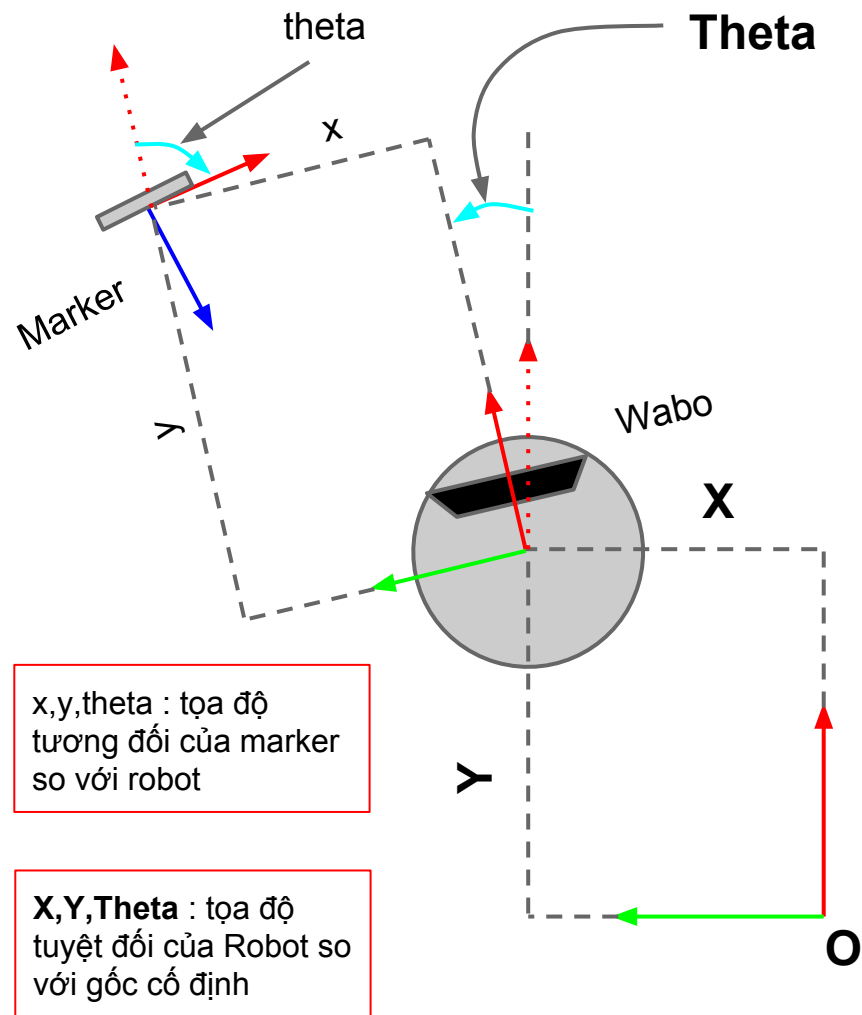
$$x, y, X, Y > 0$$

$$\text{theta} < 0, \text{Theta} > 0$$

Ví dụ:

Tọa độ object so với robot được gọi là tọa độ **tương đối** của object so với robot

Tọa độ **tuyệt đối** của object là tọa độ của nó so với gốc cố định trên sân



Hệ tọa độ (tiếp theo)

(Định nghĩa dùng trong SMAC Challenge)

Các marker gắn cố định trên sân (dùng để định vị): Marker(x,y,theta) - kích thước: 18x18 cm

Marker 17 (-2.5, 0.0, 0.0)

Marker 16 (0.0,-2.5, 0.0)

Marker 15 (2.5, 0.0, 0.0)

Marker 14 (0.0, 2.5, 0.0)

Dán trên sàn

Marker 13 (3.0, 2.0, 0.0)

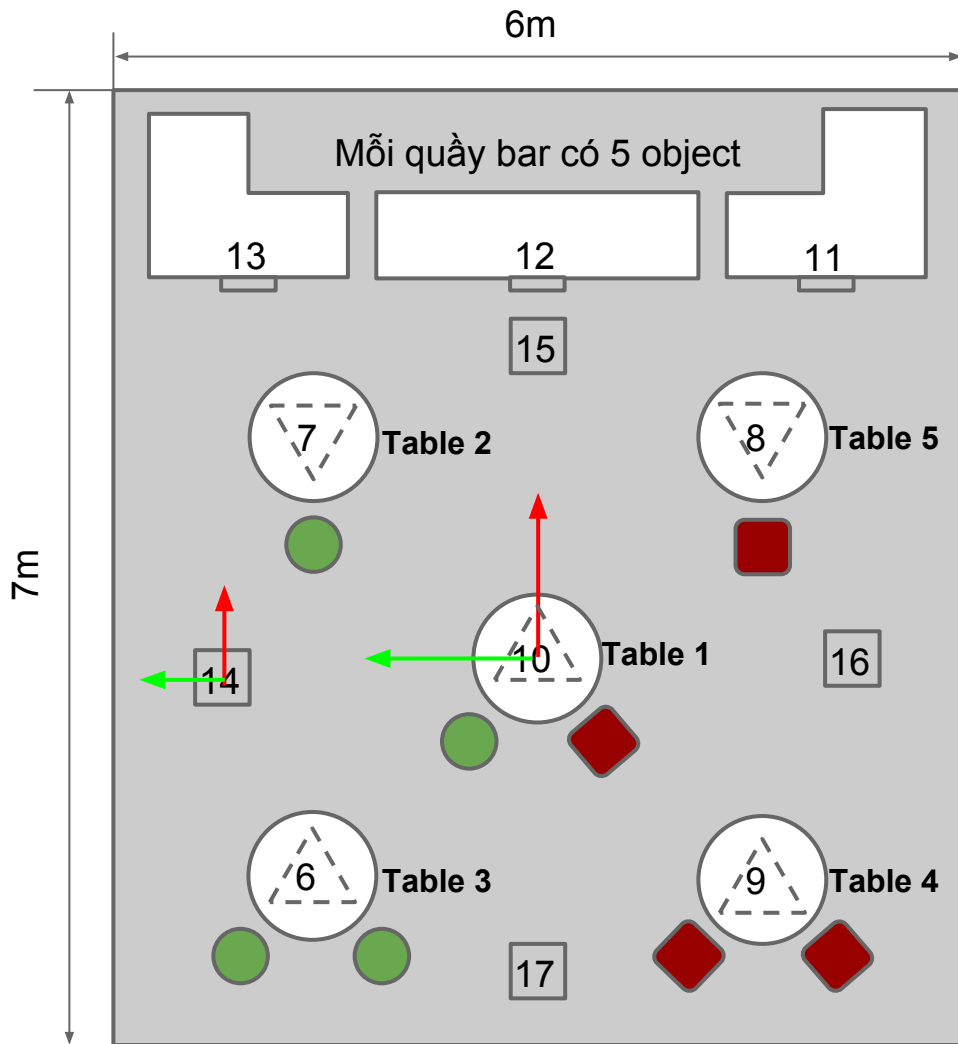
Marker 12 (3.0, 0.0, 0.0)

Marker 11 (3.0,-2.0, 0.0)

Dán trên bàn
quầy bar

Các marker gắn tại bàn tròn dùng để trợ giúp quá trình đặt object lên bàn (**Marker 10,9,8,7,6**).

Các marker gắn tại bàn tròn theo 3 hướng (cạnh của hình tam giác có nét đứt)



Get current pose (xác định vị trí hiện thời của wabo)

Là action cho phép lấy thông tin tọa độ hiện thời của robot

Sử dụng khi cần biết vị trí của robot trên sân theo hệ tọa độ gắn với robot khi khởi động

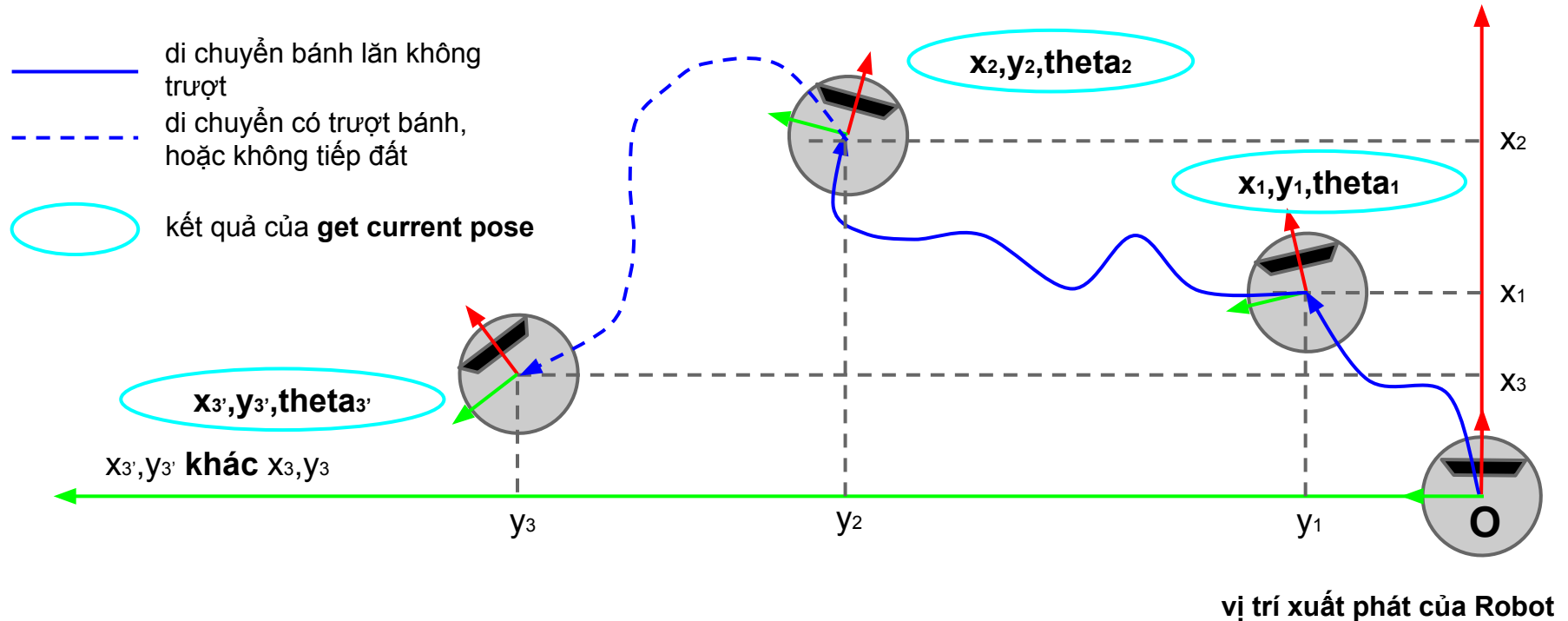
Điều kiện trả về kết quả chính xác là trong quá trình chuyển động của robot, bánh của nó phải luôn lăn không trượt trên mặt sân

Khi khởi động tại vị trí xuất phát, gọi hàm get current pose sẽ trả về tọa độ (0.0,0.0,0.0). Các tọa độ khi get current pose sau đó sẽ lấy điểm xuất phát này làm gốc

Chú ý:

- Khi robot không thỏa mãn điều kiện của hàm này (có trượt bánh khi chuyển động, hoặc bị nhấc khỏi mặt đất) kết quả trả về sẽ không chính xác (xem hình trang sau)
- Lúc này nên sử dụng hàm **Relocation** để định vị lại vị trí robot. Khi sử dụng hàm Relocation thì hệ tọa độ gắn với Robot lúc khởi động sẽ được chuyển về hệ tọa độ gốc của sân. Do đó lúc này hàm get current pose sẽ trả về tọa độ tuyệt đối của Robot trên sân

Get current pose (xác định vị trí hiện thời của wabo)



Relocation (định vị lại vị trí hiện thời của wabo)

Là action cho phép robot tự định vị lại vị trí của mình dựa vào các marker (có vị trí cố định) dán sẵn trên sân

Sử dụng khi robot được cho là bị lạc khỏi quỹ đạo mong muốn, hoặc đơn giản chỉ muốn xác định lại vị trí cho các mục đích khác

Thông tin trả về

status: trạng thái của action (thành công, time out, thất bại ...)

0: success

1: time out - không tìm thấy marker nào trong tầm nhìn (tầm nhìn định vị 2,5m)

2: reset odometry false - call service reset odometry thất bại

3: lookuptransform false - chuyển hệ tọa độ thất bại

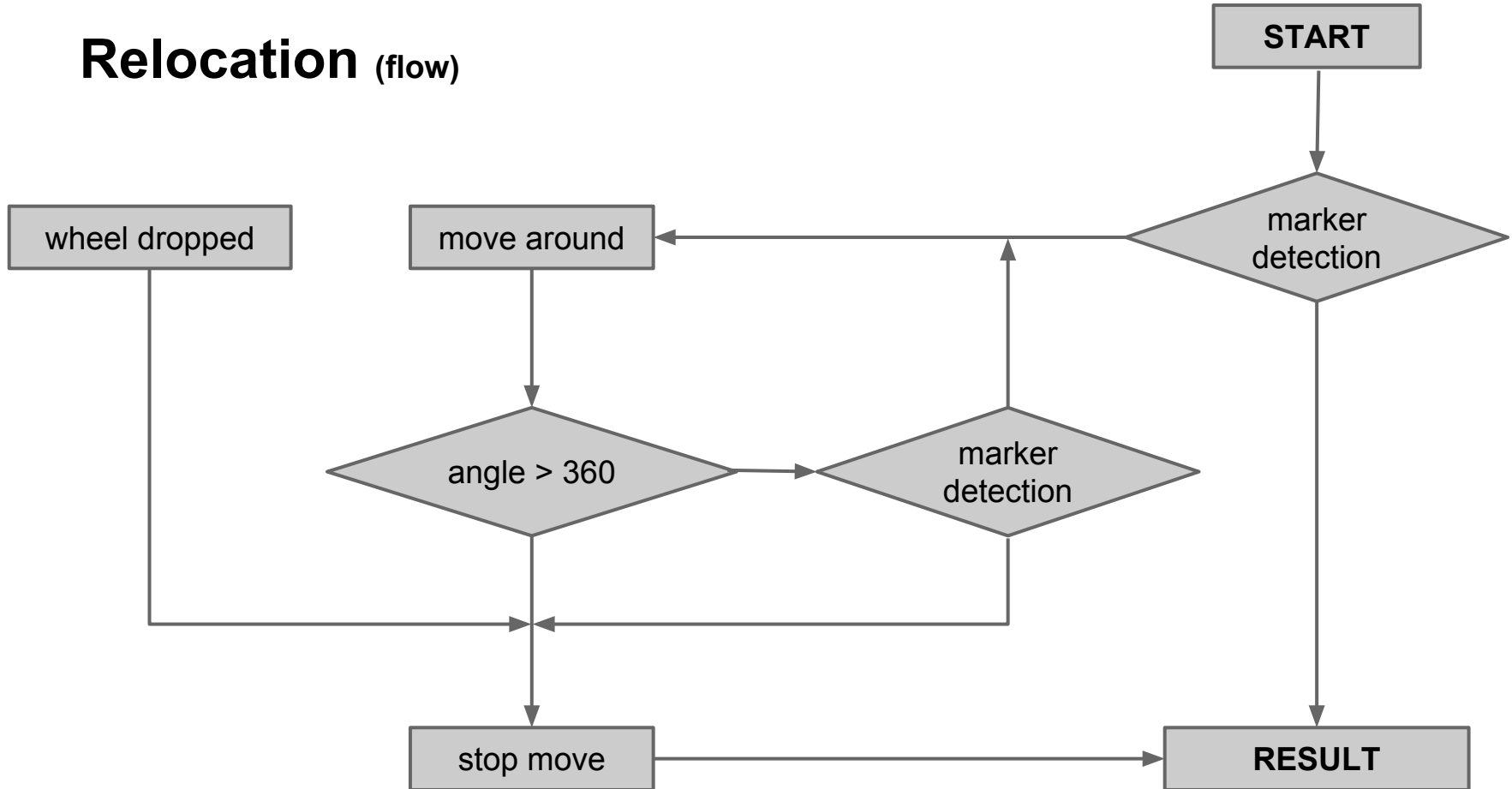
4: stop move false - call service stop move thất bại

5: wheel dropped false - robot bị nhấc lên trong khi đang relocation





x, y, theta: vị trí robot trên sân trong hệ tọa độ tuyệt đối

Sử dụng khi robot được cho là bị lạc khỏi quỹ đạo mong muốn, hoặc đơn giản chỉ muốn xác định lại vị trí cho các mục đích khác

Relocation (flow)



Relocation (example)

-  di chuyển bánh lăn không trượt
-  di chuyển có trượt bánh, hoặc không tiếp đất
-  kết quả của **get current pose**
-  kết quả của **relocation**

Bị lạc rồi, làm sao đây ?
⇒ Relocation

x_2, y_2, θ_2

x_2', y_2', θ_2'

x_1, y_1, θ_1

x_2

x_1

x_3

O

y_3

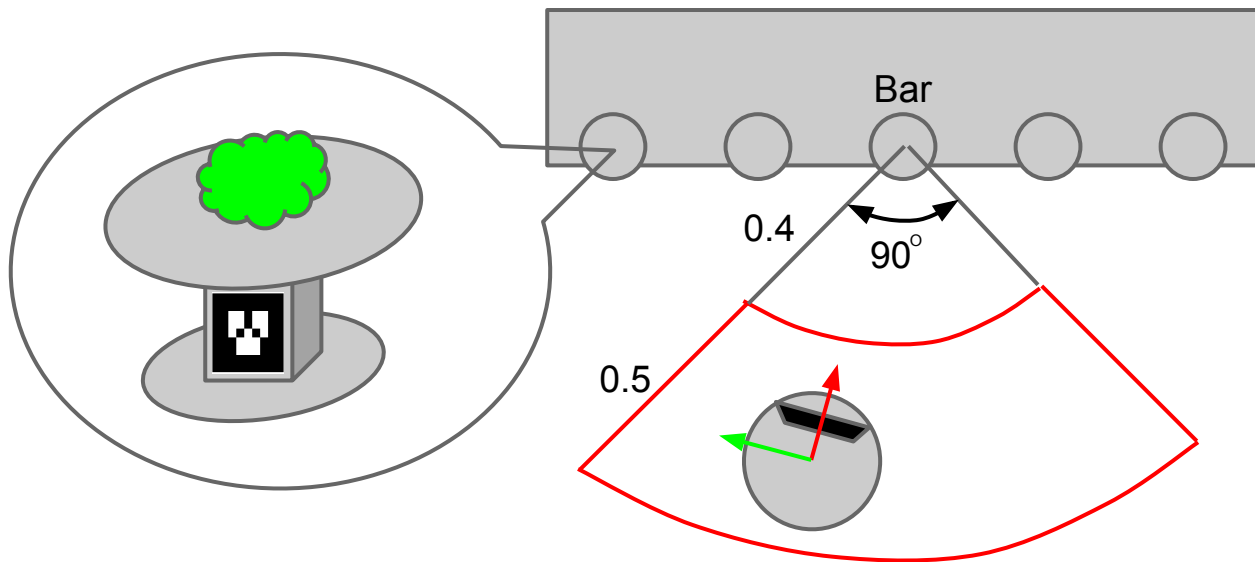
y_2

y_1

Moving closer to object (di chuyển tới vị trí có thể gấp được object)

Là action cho phép robot di chuyển từ **vị trí thuận lợi** tới gần object - vị trí có thể gấp được object

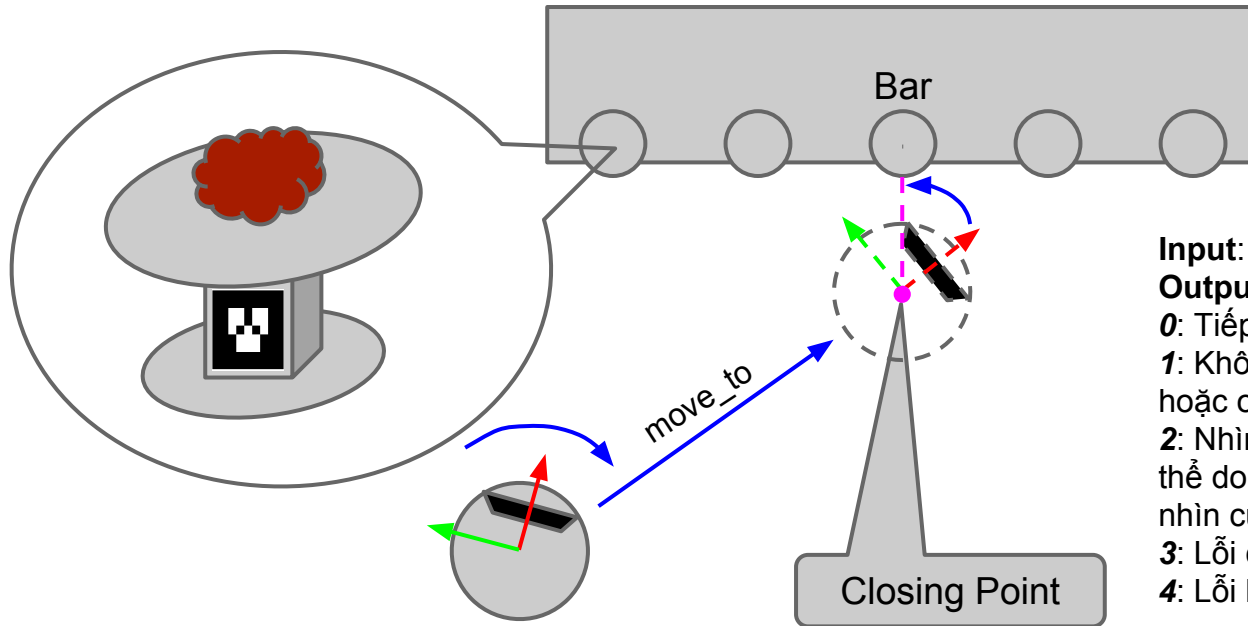
Vị trí thuận lợi: Là vị trí nằm trong một phần hình vành khăn có tâm là object bán kính lớn là 90 cm, bán kính nhỏ là 40 cm, góc mở là 90 độ (vùng màu đỏ)



Độ chính xác tiếp cận sẽ cao hơn khi Robot ở gần object và hướng vào object

Moving closer to object (flow)

Ý tưởng: Tìm tọa độ điểm đích mà robot sẽ dừng lại trước Object (closing point), và di chuyển đến đó - sử dụng hàm *move to*



Độ chính xác tiếp cận không phải 100%, nên có xác suất gặp trượt object

Input: ID của object cần tiếp cận

Output: result

0: Tiếp cận thành công

1: Không nhìn thấy object trong vùng nhìn hoặc object xuất hiện không ổn định

2: Nhìn thấy object nhưng nhiều quá nhiều, có thể do ánh sáng hoặc object nằm ở biên vùng nhìn của robot

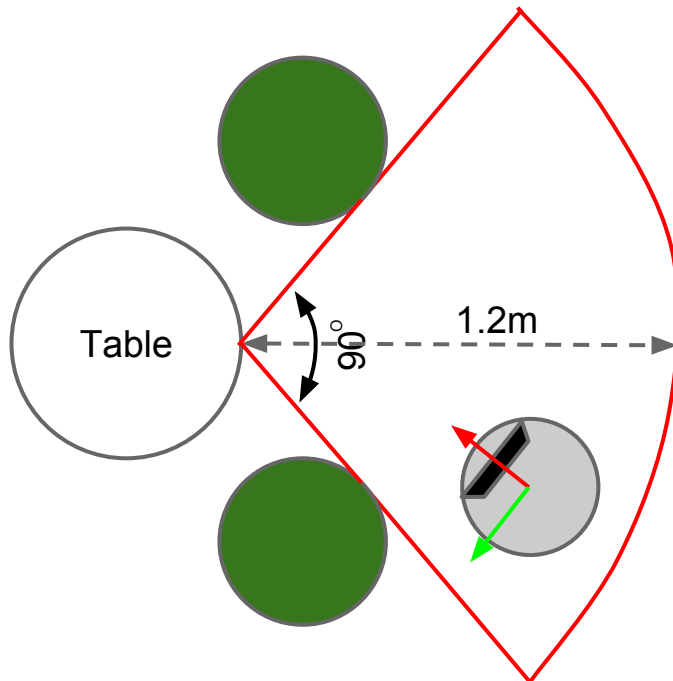
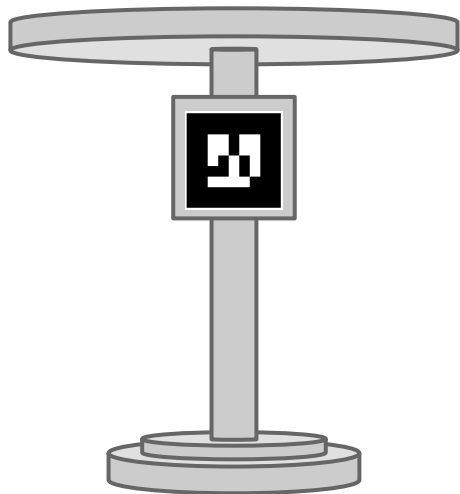
3: Lỗi chức năng biến đổi tọa độ

4: Lỗi khi gọi hàm *move_to*

Moving closer to table (di chuyển tới vị trí có thể đặt được object)

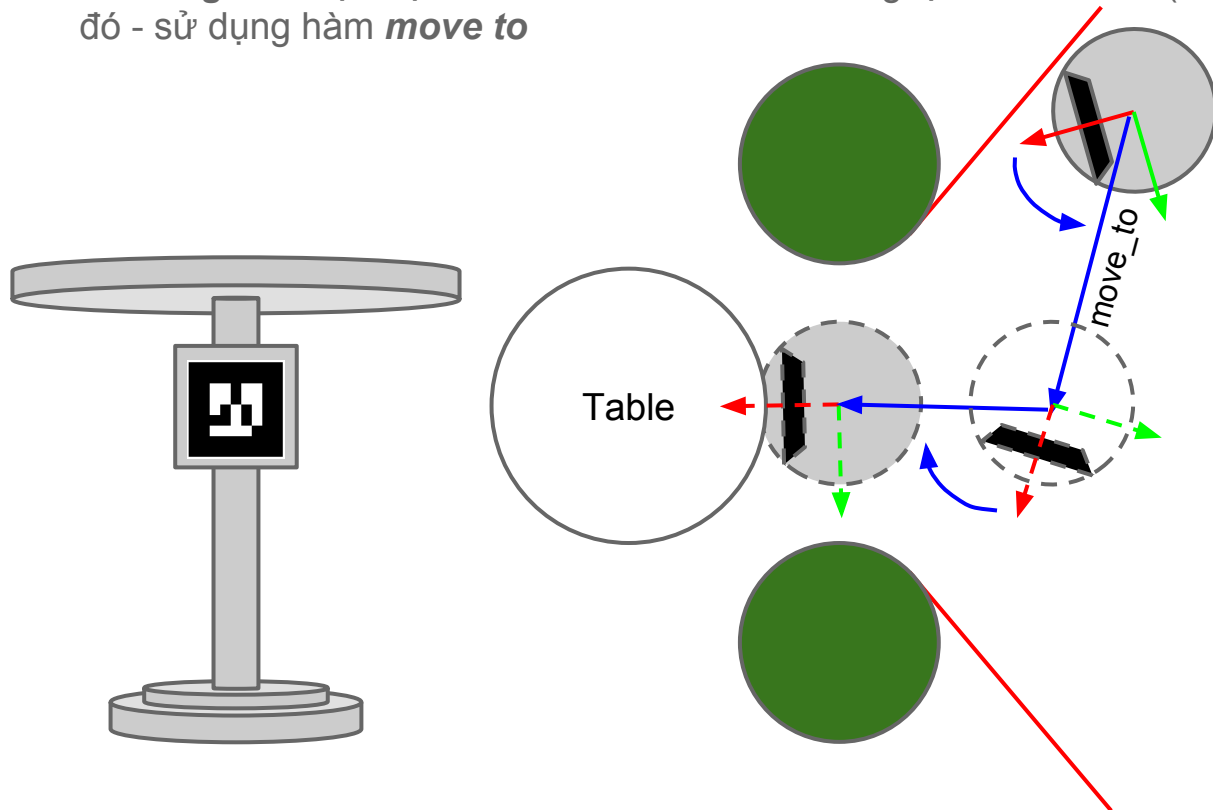
Là action cho phép robot di chuyển từ **vị trí thuận lợi** tới gần bàn (table) - vị trí có thể đặt được object

Vị trí thuận lợi: Là vị trí nằm trong một phần hình tròn có dạng như vùng màu đỏ ở hình dưới



Moving closer to table (flow)

Ý tưởng: Tìm tọa độ điểm đích mà robot sẽ dừng lại trước Table (closing point), và di chuyển đến đó - sử dụng hàm *move to*



Input: ID của object cần tiếp cận

Output: result

- 0:** Tiếp cận thành công
- 1:** Không nhìn thấy table trong vùng nhìn hoặc table xuất hiện không ổn định
- 2:** Nhìn thấy table nhưng nhiễu quá nhiều, có thể do ánh sáng hoặc table nằm ở biên vùng nhìn của robot
- 3:** Lỗi chức năng biến đổi tọa độ
- 4:** Lỗi khi gọi hàm *move_to*
- 5:** Thấy marker nhưng nằm trong vùng vướng vật cản