

Lời nói đầu

Robot tự hành (Mobile Robot) là một thành phần có vai trò quan trọng trong ngành Robot học. Cũng với sự phát triển mạnh mẽ của các hệ thống Cơ - Điện tử, robot tự hành ngày một được hoàn thiện và càng cho thấy lợi ích của nó trong công nghiệp và sinh hoạt.

Một vấn đề rất được quan tâm khi nghiên cứu về robot tự hành là làm thế nào để robot có thể di chuyển tới một vị trí xác định mà có thể tự động tránh được các chướng ngại vật trên đường đi. Tiểu luận này có mục đích đưa ra các phương pháp để giải quyết vấn đề trên. Trong phần một, chúng tôi xét 2 cách giải bài toán tìm đường cục bộ và toàn cục đối với một robot có mô hình động học cho trước, sau đó tiến hành mô phỏng thử nghiệm trên máy tính. Phần hai của tiểu luận tập trung vào giới thiệu các cảm biến, các mạch phần cứng điều khiển một robot tự hành cụ thể đã cho chạy thử.

Chúng em xin chân thành cảm ơn những gợi ý, chỉ dẫn rất quan trọng cho đề tài này của PGS.TS Tạ Duy Liêm. Thầy cũng đã cung cấp cho chúng em những kiến thức và tài liệu Robotics cần thiết trong học kỳ vừa qua. Chúng em hy vọng đây sẽ là một tiểu luận có ích cho những ai yêu thích đề tài này.

Hà Nội, ngày 30/10/2005

Sinh viên:

Nguyễn Vũ Quang
Đoàn Nam Thái

MỤC LỤC

PHẦN 1

GIẢI BÀI TOÁN TÌM ĐƯỜNG TRÁNH VẬT CẢN CHO ROBOT TỰ HÀNH ...	3
I. Giới thiệu về robot tự hành.....	4
II. Mô hình động học robot tự hành.....	6
III. Giải bài toán tìm đường cho robot tự hành.....	7
1. Đặt vấn đề.....	7
2. Bài toán tìm đường cục bộ.....	8
3. Bài toán tìm đường toàn cục.....	13

PHẦN 2

CẤU TẠO PHẦN CỨNG VÀ CÁC PHƯƠNG PHÁP ĐIỀU KHIỂN TRONG ROBOT TỰ HÀNH.....	19
I. Các loại cảm biến.....	20
1. Sensor quang (Optical Sensor).....	20
2. Đĩa encoder.....	20
3. Công tắc hành trình.....	21
II. Mạch điều khiển Robot.....	21
1. Khối nguồn.....	21
2. Khối điều khiển.....	22
3. Khối sensor.....	24
4. Khối hiển thị.....	25
5. Khối công suất điều khiển động cơ.....	27
III. Các phương pháp điều khiển trong Robot tự hành.....	28
1. Phương pháp tạo encoder từ chuột máy tính.....	28
2. Phương pháp quét đèn led 7 thanh.....	30
3. Phương pháp tạo xung PWM để điều khiển tốc độ cho robot.....	33
IV. Xây dựng một bài toán cho Robot tự hành.....	34
V. Hạn chế Robot và phương pháp khắc phục.....	36

PHẦN 1

GIẢI BÀI TOÁN TÌM ĐƯỜNG TRÁNH VẬT CẢN CHO ROBOT TỰ HÀNH

- Giới thiệu về robot tự hành
- Mô hình động học của robot tự hành
- Giải bài toán tìm đường cho robot tự hành
 - ✓ Đặt vấn đề
 - ✓ Bài toán tìm đường cục bộ (local navigation)
 - ✓ Bài toán tìm đường toàn cục (global navigation)

I. Giới thiệu về robot tự hành

Robot tự hành hay robot di động (mobile robots, thường được gọi tắt là mobots) được định nghĩa là một loại xe robot có khả năng tự dịch chuyển, tự vận động (có thể lập trình lại được) dưới sự điều khiển tự động để thực hiện thành công công việc được giao.

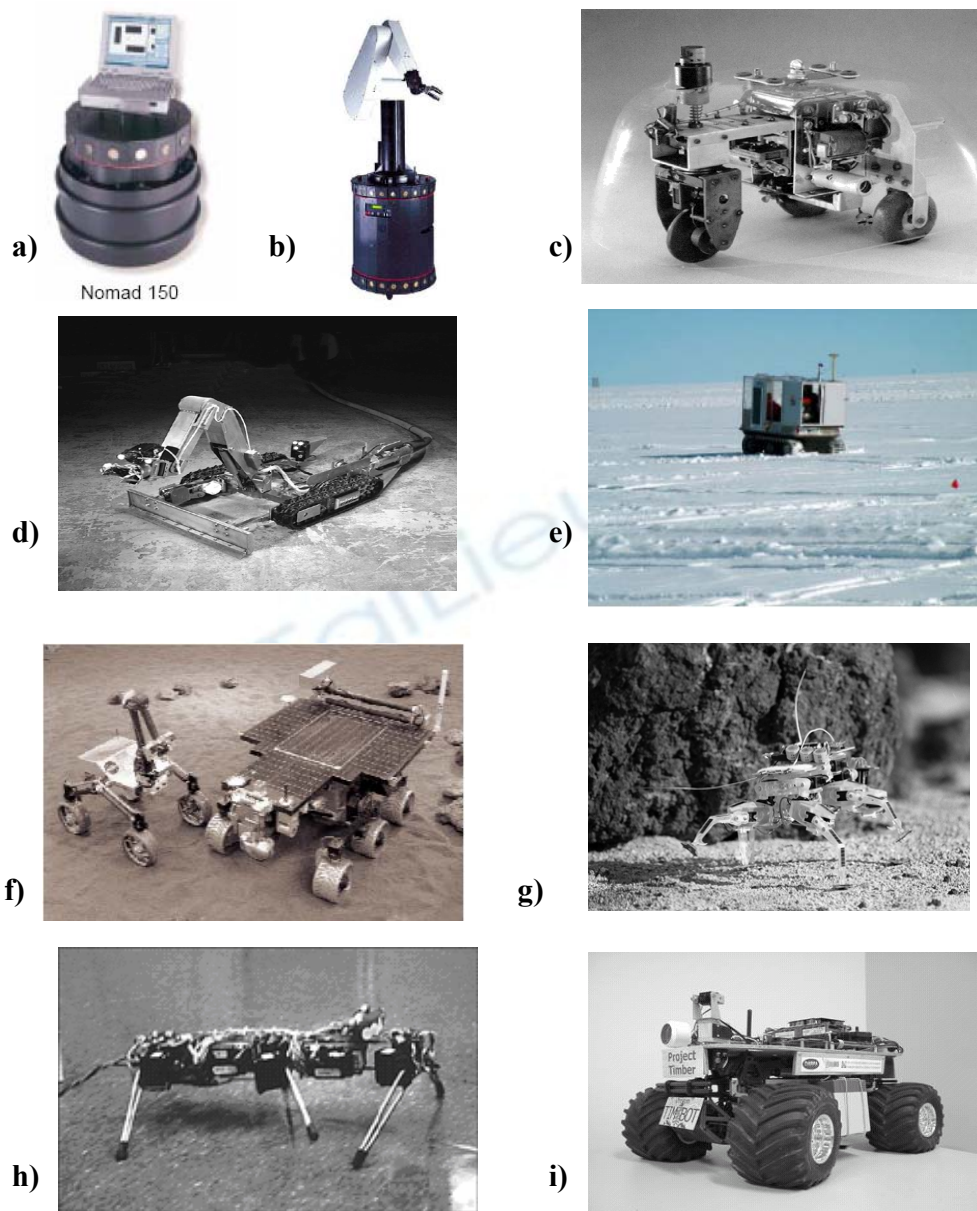
Theo lý thuyết, môi trường hoạt động của robot tự hành có thể là đất, nước, không khí, không gian vũ trụ hay sự tổ hợp giữa chúng. Địa hình bề mặt mà robot di chuyển trên đó có thể bằng phẳng hoặc thay đổi, lồi lõm.

Theo bộ phận thực hiện chuyển động, ta có thể chia robot tự hành làm 2 lớp: chuyển động bằng chân (legged) và bằng bánh (wheeled). Trong lớp đầu tiên, chuyển động có được nhờ các chân cơ khí bắt chước chuyển động của con người và động vật (hình 1.1g, h). Robot loại này có thể di chuyển rất tốt trên các định hình lồi lõm, phức tạp. Tuy nhiên, cách phối hợp các chân cũng như vấn đề giữ vững tư thế là công việc cực kỳ khó khăn.

Lớp còn lại (di chuyển bằng bánh) tỏ ra thực tế hơn, chúng có thể làm việc tốt trên hầu hết các địa hình do con người tạo ra. Điều khiển robot di chuyển bằng bánh cũng đơn giản hơn nhiều, gần như luôn đảm bảo tính ổn định cho robot. Lớp này có thể chia làm 3 loại robot: Loại chuyển động bằng bánh xe (phổ biến) (hình 1.1a, b, c, f, i), loại chuyển động bằng vòng xích (khi cần mô men phát động lớn hay khi cần di chuyển trên vùng đầm lầy, cát và băng tuyết) (hình 1.1d, e), và loại hỗn hợp bánh xe và xích (ít gặp).

Tiềm năng ứng dụng của robot tự hành hết sức rộng lớn. Có thể kể đến robot vận chuyển vật liệu, hàng hóa trong các tòa nhà, nhà máy, cửa hàng, sân bay và thư viện; robot phục vụ quét dọn đường phố, khoang chân không; robot kiểm tra trong môi trường nguy hiểm; robot canh gác, do thám; robot khám phá không gian, di chuyển trên hành tinh; robot hàn, sơn trong nhà máy; robot xe lăn phục vụ người khuyết tật; robot phục vụ sinh hoạt gia đình v.v...

Mặc dù nhu cầu ứng dụng cao, nhưng những hạn chế chưa giải quyết được của robot tự hành, như chi phí chế tạo cao, đã không cho phép chúng được sử dụng rộng rãi. Một nhược điểm khác của robot tự hành phải kể đến là còn thiếu tính linh hoạt và thích ứng khi làm việc ở những vị trí khác nhau. Bài toán tìm đường (navigation problem) của robot tự hành cũng không phải là loại bài toán đơn giản như nhiều người nghĩ lúc ban đầu. Trong tiểu luận này, bài toán tìm đường sẽ được giải quyết ở mức độ không quá phức tạp.

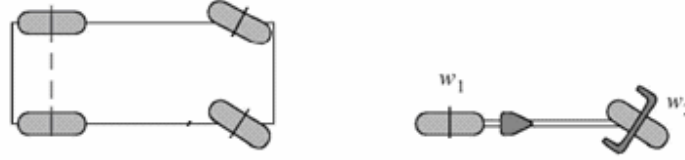


Hình 1.1. Một số hình ảnh về robot tự hành

a) Robot Nomad 150. b) Sự kết hợp giữa robot tự hành Nomadic XR4000 và cánh tay robot PUMA 560. c) Robot “con rùa”. d) Robot Houdini di chuyển bằng bánh xích e) Robot bán tự động nghiên cứu ở vùng cực bắc. f) Robot SRR và FIDO của NASA g) Robot tự hành bằng chân Spiderbot di chuyển trên sao Hỏa. h) Robot sáu chân Genghis i) Robot Meet Timbot dùng test các phần mềm hệ thống nhúng trong phòng thí nghiệm

II. Mô hình động học robot tự hành

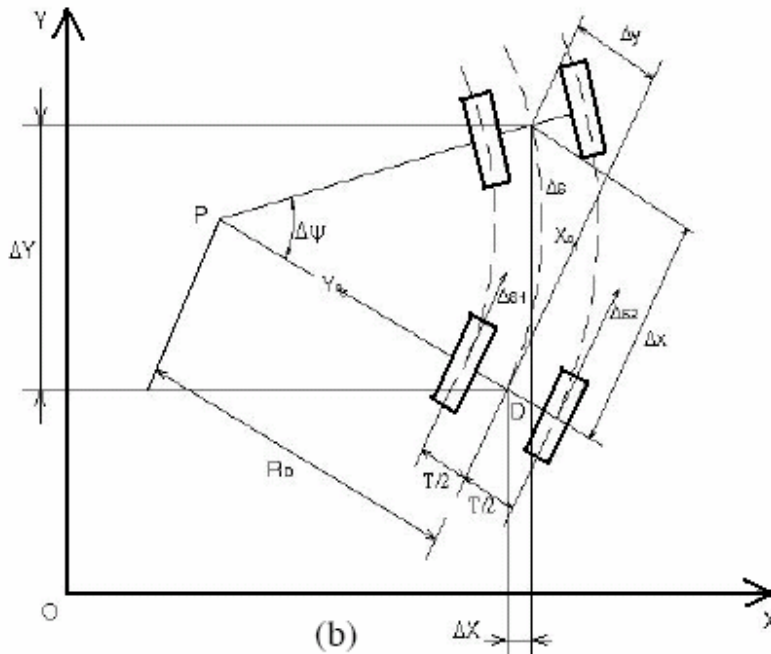
Mô hình kết cấu



Hình 1.2. Mô hình kết cấu của robot

Kết cấu robot lựa chọn gồm 4 bánh trong đó có 2 bánh chủ động (phía sau) và 2 bánh tùy động có khả năng quay tùy ý. Với kết cấu cơ khí này robot có khả năng di chuyển rất linh hoạt: tiến, lùi và quay một góc bất kỳ.

Động học của robot



Gọi $\Delta s_1, \Delta s_2$ là đoạn dịch chuyển của 2 bánh chủ động. Ta có
 $\Delta s_1 = \Delta \varphi_1 \cdot r$
 $\Delta s_2 = \Delta \varphi_2 \cdot r$
 với $\Delta \varphi_1, \Delta \varphi_2, r$ lần lượt là lượng dịch chuyển quay và bán kính của 2 bánh chủ động.

Hình 1.3. Động học của robot

T là khoảng cách giữa 2 bánh. $R_D, \Delta \Psi$ là bán kính quay và góc dịch chuyển của robot trong mặt phẳng nằm ngang. Khi đó:

$$R_D = \frac{T}{2} \left(\frac{\Delta s_2 + \Delta s_1}{\Delta s_2 - \Delta s_1} \right), \quad \Delta \Psi = \frac{\Delta s_2 - \Delta s_1}{T}$$

Trong tiêu luận này, ta chỉ xét 2 trường hợp đặc biệt của chuyển động:

Khi $\Delta s_2 = \Delta s_1$: $R_D = \infty, \Delta \Psi = 0$, robot đi theo quỹ đạo là đường thẳng (tiến hoặc lùi)

Khi $\Delta s_2 = -\Delta s_1$: $R_D = 0, \Delta \Psi = 2\Delta s_2 / T$, robot quay quanh điểm giữa 2 bánh xe D.

Δx , Δy : lượng dịch chuyển theo 2 phương của hệ tọa độ gắn với robot:

$$\Delta x = R_D \sin(\Delta\Psi), \quad \Delta y = R_D(1 - \cos(\Delta\Psi))$$

ΔX , ΔY : lượng dịch chuyển của robot theo 2 phương gắn với hệ tọa độ gốc:

$$\begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} = \begin{bmatrix} \Delta x & -\Delta y \\ \Delta y & \Delta x \end{bmatrix} \begin{bmatrix} \cos(\Delta\Psi) \\ \sin(\Delta\Psi) \end{bmatrix}$$

Vị trí của robot được xác định bởi tọa độ gốc của D (X_D và Y_D) + góc định hướng Ψ_D , tọa độ tại thời điểm thứ (i) được xác định như sau:

$$\begin{bmatrix} \Psi_{D(i)} \\ X_{D(i)} \\ Y_{D(i)} \end{bmatrix} = \begin{bmatrix} \Psi_{D(i-1)} \\ X_{D(i-1)} \\ Y_{D(i-1)} \end{bmatrix} + \begin{bmatrix} \Delta\Psi_{D(i)} \\ \Delta X_{D(i)} \\ \Delta Y_{D(i)} \end{bmatrix}$$

III. Giải bài toán tìm đường cho robot tự hành

1. Đặt vấn đề

Dò đường (navigation) là một khoa học (hay nghệ thuật) dẫn hướng robot tự hành di chuyển trong không gian làm việc của nó (đất, nước, không khí...). Trong vấn đề dò đường, bài toán được quan tâm nhiều nhất có lẽ là tìm đường về đích mà không chạm vật cản trên đường đi. Có hai loại bài toán tìm đường cho robot: bài toán cục bộ (local) và bài toán toàn cục (global). Trong bài toán toàn cục, bản đồ môi trường làm việc của robot hoàn toàn được biết trước, vấn đề chính phải giải quyết là tìm được đường đi cho robot trước khi nó xuất phát. Ở bài toán cục bộ, môi trường làm việc của robot hoàn toàn không được biết trước hoặc chỉ biết được một phần, robot hoàn toàn phải nhờ vào sự cảm nhận môi trường thông qua cảm biến gắn trên nó để dò đường.

Bài toán toàn cục tỏ rõ lợi thế là ta đã biết trước có đường đi (tối ưu) tới đích hay không trước khi cho robot khởi hành. Tuy vậy nó có hạn chế là đòi hỏi nhiều lệnh tính toán và bộ nhớ, và đặc biệt tình huống xấu có thể xảy ra nếu bản đồ môi trường làm việc không được khai báo chính xác, yêu cầu biết trước hoàn toàn môi trường hoạt động cũng là một nhược điểm của nó. Trong khi đó, robot tìm đường cục bộ chỉ biết được thông tin xung quanh qua sensor cảm nhận môi trường gần cùng. Vì thế, robot tìm đường cục bộ có thể không hoàn thành việc tới đích (mặc dù thực tế có đường đi tới đích), khái niệm tối ưu không có ý nghĩa trong bài toán này. Tuy nhiên, yêu cầu tính toán, dung lượng nhớ thấp cùng tính linh hoạt cao (như tránh được vật cản ngay cả khi vật đó di động) khiến tìm đường cục bộ trở thành một công cụ mạnh của robot tự hành. Hiện nay, cách tiếp cận kết hợp giữa cục bộ và toàn cục đang ở ngày càng phổ biến, giúp robot tự hành tăng tính linh hoạt và hiệu quả.

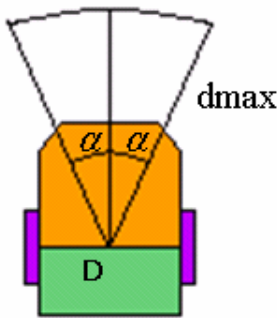
Trong tiểu luận, chúng tôi tập trung vào giải quyết cả hai bài toán nhưng ở mức độ đơn giản. Môi trường hoạt động của robot là một mặt phẳng, được giới hạn bởi các bức tường. Các vật cản được xem là vật cản hai chiều, *tĩnh hoàn toàn*. Trong bài toán cục bộ, robot được trang bị 3 sensor xác định khoảng cách vật cản, đã biết trước vị trí đích (đích có thể phát sóng để robot hướng theo). Trong bài toán toàn cục, từ đầu vào là vị trí xuất phát, vị trí đích, kích thước robot, bản đồ hoạt động, ta tìm ra đường đi để robot tự hành. Mục

đích cuối cùng là robot tự về được tới đích không chạm vật cản mà không cần bất cứ sự trợ giúp nào từ bên ngoài.

2. Bài toán tìm đường cục bộ

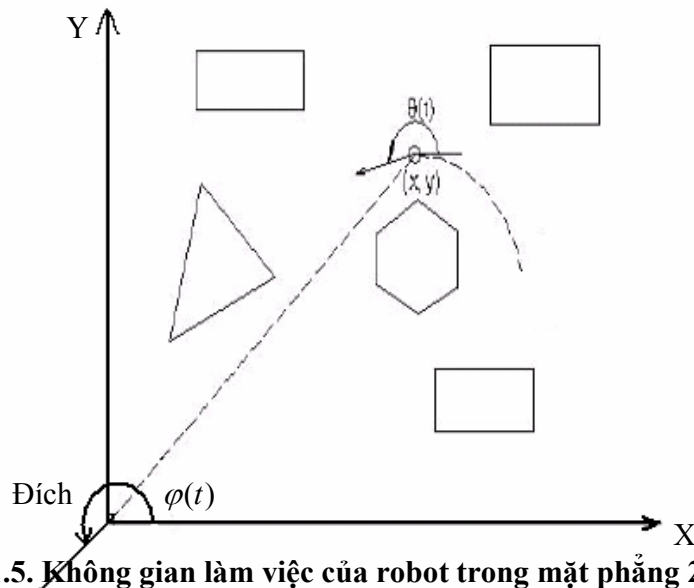
Giải quyết *hoàn toàn* bài toán này khó khăn hơn nhiều so với giải quyết bài toán toàn cục. Một cách tìm đường cục bộ khá hoàn hảo là phương pháp bản đồ nơron cực (Polar Neurol Map) của Michail G. Lagoudakis (1998) dùng 16-32 cảm biến gắn trên con robot Normad (hình 1.1a, b) tạo trường cảm biến bao phủ vùng xung quanh robot, thuật toán sử dụng là thuật toán ánh xạ nơron tương đối phức tạp. Do giới hạn của tiểu luận, phương pháp này không được đề cập đến.

Phương pháp được đề cập đến ở đây sử dụng 3 cảm biến dò đường xác định khoảng cách từ trung điểm hai bánh xe tới vật cản. Tùy vào khoảng cách tới vật cản ứng với từng cảm biến, ta xác định được phương hướng di chuyển robot phù hợp. Ý tưởng của phương pháp xuất phát từ một bài báo Cơ điện tử [1]. Tuy nhiên luật dẫn hướng robot trong bài báo còn khá đơn giản, điều kiện yêu cầu hơi thô (như khoảng cách các vật cản phải lớn hơn kích thước robot), kết quả mô phỏng cho thấy thuật toán của bài báo tỏ ra nhiều bất cập: robot nhiều lúc chạm vật cản, lặp đi lặp lại một đoạn đường và khó thoát ra được. Dựa theo nguyên lý của thuật toán đó, chúng tôi đã có những thay đổi và bổ sung để đem lại kết quả tìm đường hiệu quả hơn.



Hình 1.4. Ba cảm biến của robot

Các cảm biến được đặt lệch một góc α (trong phần mô phỏng, chọn $\alpha = 25^\circ$). Khoảng cách lớn nhất (tính từ D) mà các cảm biến có thể nhận diện được vật cản là d_{max} ; d_{max} và α phải đảm bảo sao cho cảm biến có vùng kiểm tra đủ rộng để khi tiến thẳng, robot có thể nhận diện được vật cản.



Hình 1.5. Không gian làm việc của robot trong mặt phẳng 2 chiều

Gọi:

$\theta(t)$: góc chỉ hướng của robot ở thời điểm t.

$\varphi(t)$: góc hướng về đích tính từ tâm robot (điểm D).

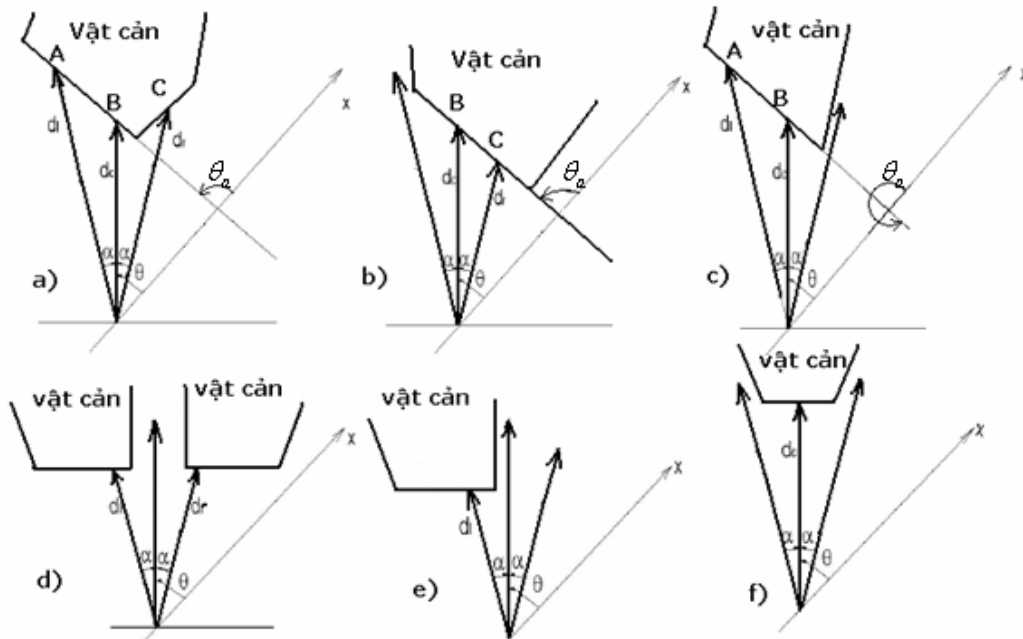
$\theta_a(t)$: góc dẫn hướng cho robot tránh vật cản.

$\theta^*(t)$: góc dẫn hướng điều khiển chuyển động cho robot.

Khi sensor không phát hiện vật cản thì $\theta^*(t) = \varphi(t)$, ngược lại thì $\theta^*(t) = \theta_a(t)$.

Cách ứng xử của robot khi sensor phát hiện vật cản:

Có 7 tình huống khác nhau khi sensor phát hiện vật cản, mỗi tình huống cần có cách ứng xử riêng.



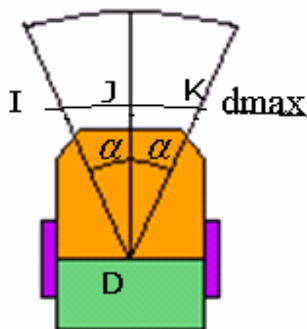
Hình 1.6. Các tình huống phát hiện vật cản

- ✓ Tình huống 1 (TH1): cả 3 cảm biến đều phát hiện vật cản (Hình 1.6a)
 - + Nếu $d_l > d_r$ thì robot quay sang trái: $\theta_a = (\overline{BA}, Ox)$
 - + Nếu $d_l \leq d_r$ thì robot quay sang phải: $\theta_a = (\overline{BC}, Ox)$
- ✓ TH2: Chỉ có cảm biến bên trái không phát hiện vật cản (Hình 1.6b)
 - Robot quay sang trái: $\theta_a = (\overline{CB}, Ox)$
- ✓ TH3: Chỉ có cảm biến bên phải không phát hiện vật cản (Hình 1.6c)
 - Robot quay sang phải: $\theta_a = (\overline{AB}, Ox)$

- ✓ TH4: Chỉ có cảm biến ở giữa không phát hiện vật cản (Hình 1.6d)
Robot quay sang phải một góc 90° (gán $\theta_a = \theta_a - 90^\circ$, sau đó hiệu chỉnh $\theta_a \in [0, 360^\circ)$)
- ✓ TH5: Chỉ có cảm biến bên trái phát hiện vật cản (Hình 1.6e)
Robot quay sang phải một góc 25° (gán $\theta_a = \theta_a - 25^\circ$, sau đó hiệu chỉnh $\theta_a \in [0, 360^\circ)$)
- ✓ TH6: Chỉ có cảm biến bên phải phát hiện vật cản
Robot quay sang trái một góc 25° (gán $\theta_a = \theta_a + 25^\circ$, sau đó hiệu chỉnh $\theta_a \in [0, 360^\circ)$)
- ✓ TH7: Chỉ có cảm biến ở giữa phát hiện vật cản (Hình 1.6f)
Robot quay sang trái một góc 90° (gán $\theta_a = \theta_a - 90^\circ$, sau đó hiệu chỉnh $\theta_a \in [0, 360^\circ)$)

Khi robot đang quay hướng về đích, nếu phát hiện vật cản nó sẽ tiến hành chuyển hướng quay luôn. Để đảm bảo điều này, trong quá trình quay hướng về đích, mỗi lần robot quay một góc đơn vị (đv) nhỏ, ta phải cập nhật cảm biến.

Sau khi robot chuyển hướng rồi tiến thẳng một đoạn, vẫn có khả năng robot chạm vật cản (nhất là ở những ngõ cụt), vì vậy khi robot quá gần vật ta tiến hành lùi lại 1 đoạn, quay một góc 45° sang phải (gán $\theta_a = \theta_a - 45^\circ$, hiệu chỉnh $\theta_a \in [0, 360^\circ)$), sau đó cập nhật cảm biến.



Hình 1.7. Khi khoảng cách nhỏ hơn hoặc bằng DI, DJ, DK tức là robot quá gần vật