

VỀ MỘT THUẬT TOÁN ĐIỀU KHIỂN TRƯỢT RÔBÔT DƯ DẪN ĐỘNG*

NGUYỄN VĂN KHANG¹, NGUYỄN QUANG HOÀNG¹, LÊ ĐỨC ĐẠT¹, TRẦN HOÀNG NAM²

¹Trường Đại học Bách khoa Hà Nội

²Trường Cao đẳng Sư phạm kỹ thuật Vĩnh Long

Abstract. Robotic manipulator control by traditional methods as PD, PID plus feed forward compensation requires exact dynamic parameters such as masses, centers of mass, and mass inertias of the system. This paper presents a algorithm of sliding mode controller for redundant manipulators. This control method does not require to know exactly dynamic parameters the system, while the control performance is guaranteed. A numerical simulation is presented to demonstrate the validity of the approach, in which a four degree of freedom planar manipulators is used for the simulation by mean of MATLAB.

Tóm tắt. Khi điều khiển rôbôt bằng các phương pháp thông thường như PD, PID cùng với bù truyền thẳng yêu cầu cần phải biết chính xác các tham số động lực học của rôbôt như khối lượng, vị trí khối tâm, mômen quán tính khối. Báo cáo này trình bày phương pháp điều khiển trượt áp dụng cho rôbôt dư dẫn động. Với phương pháp điều khiển này chất lượng điều khiển được đảm bảo mà không cần biết chính xác các tham số của hệ. Các kết quả mô phỏng số đối với rôbôt phẳng bốn bậc tự do được thực hiện nhờ phần mềm đa năng Matlab.

1. GIỚI THIỆU CHUNG

Ngày nay rôbôt được sử dụng trong nhiều lĩnh vực từ phục vụ sản xuất trong các nhà máy, trợ giúp các công việc nghiên cứu trên mặt đất, dưới đại dương, trong không gian và thậm chí trên cả bề mặt sao hỏa. Rôbôt công nghiệp là một thí dụ tiêu biểu về hệ cơ điện tử, một hệ bao hàm nhiều lĩnh vực trong nó như: cơ học, kỹ thuật điện-điện tử, kỹ thuật điều khiển, kỹ thuật đo xử lý số liệu, kỹ thuật lập trình,...

Bên cạnh việc chế tạo lắp ráp, bài toán nghiên cứu động lực học và thiết kế điều khiển có một vai trò quan trọng để rôbôt thực hiện được các thao tác mong muốn. Hiện nay có rất nhiều phương pháp khác nhau trong việc điều khiển rôbôt công nghiệp như điều khiển dựa trên mô hình tuyến tính hóa, điều khiển mờ, điều khiển thích nghi, điều khiển dựa trên mạng nơron,... Mỗi phương pháp đều có những ưu và nhược điểm (Hills and Yoerger, 1994). Phương pháp điều khiển tuyến tính đã được phát triển từ lâu và với độ tin cậy cao. Kỹ thuật này thường được áp dụng cho các hệ tuyến tính. Tuy nhiên, phương pháp này dựa trên các phương trình tuyến tính hóa quanh điểm làm việc và nó đòi hỏi phải biết được chính xác các thông số của rôbôt. Phương pháp điều khiển thích nghi cho phép điều chỉnh các thông số

* Bài báo được hoàn thành với sự tài trợ của Quỹ phát triển khoa học và công nghệ quốc gia

điều khiển, tuy nhiên quá trình điều chỉnh này lại nhạy đối với nhiễu động ngoài. Điều khiển mờ là một trong các phương pháp hiện đại, hiện nay đang được đề cập đến nhiều. Phương pháp này dựa trên một bộ qui tắc được xác lập bởi người thiết kế. Ưu điểm của điều khiển mờ là phương pháp này chỉ yêu cầu biết một số thông tin tối thiểu về hệ, nó có thể áp dụng được cho các hệ tuyến tính và phi tuyến. Tuy nhiên, phương pháp này không khai thác được sự hiểu biết về hệ động lực. Điều khiển trượt là phương pháp điều khiển dựa trên mô hình, nó đề cập đến hiệu ứng động lực cũng như đáp ứng lại các sai số phản hồi.

Trong báo cáo này phương pháp điều khiển trượt được khai thác sử dụng để thiết kế điều khiển cho rôbot dư dẫn động. Ưu điểm của phương pháp là ta không cần biết thật chính xác các thông số hệ mà vẫn đảm bảo độ chính xác của chuyển động mong muốn.

2. MÔ HÌNH ĐỘNG LỰC CỦA RÔBOT

Để nghiên cứu động lực học và trên cơ sở đó thiết kế bộ điều khiển cho một hệ động lực nói chung, một hệ cơ học như rôbot công nghiệp nói riêng. Một mô hình toán học mô tả liên hệ giữa đầu vào đầu ra của hệ là rất cần thiết. Mô hình toán học này có thể là phương trình vi phân thường hoặc là phương trình vi phân đại số. Phương trình Lagrange loại 2 và phương trình Newton-Euler thường được sử dụng để thiết lập mô hình toán học cho một rôbot công nghiệp có cấu trúc cây với n khâu động có n bậc tự do (Khang, 2007). Mô hình toán học mô tả động lực học của một rôbot công nghiệp n bậc tự do được viết dạng ma trận như sau (Spong, Hutchinson, and Vidyasagar, 2006, L. Sciavicco and B. Siciliano, 2000)

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + d = \tau, \quad (1)$$

trong đó:

$M(q)$: ma trận khối lượng cỡ $n \times n$,

$C(q, \dot{q})\dot{q}$: vectơ cỡ n chứa các lực Côriôlis và lực ly tâm,

$g(q)$: vectơ cỡ n chứa các lực do trọng trường,

d : vectơ chứa các lực/mômen do kích động nhiễu,

τ : vectơ chứa các lực/mômen của động cơ dẫn động.

Hệ động lực (1) có một số tính chất quan trọng sau (Murray, Li, and Sastry, 1994)

- Ma trận khối lượng $M(q)$ là ma trận đối xứng, xác định dương và tồn tại hai số dương $\mu_1(q)$ và $\mu_2(q)$ thỏa mãn

$$\mu_1(q)I \leq M(q) \leq \mu_2(q)I,$$

nếu tất cả các khớp đều là khớp quay thì μ_1, μ_2 là hằng số.

- Ma trận $N(q, \dot{q}) = \dot{M}(q) - 2C(q, \dot{q})$ là ma trận đối xứng lệch, tức là

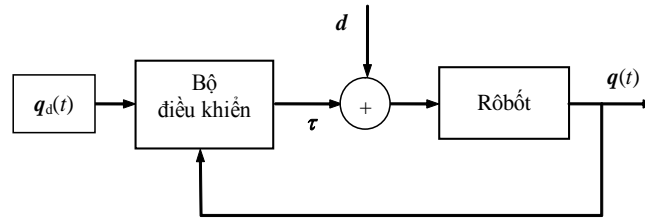
$$x^T [N(q, \dot{q})] x = 0.$$

- Tồn tại các hằng số ρ_i sao cho:

$$\|M(q)\| \leq \rho_1, \quad \|C(q, \dot{q})\| \leq \rho_2 \|\dot{q}\|, \quad \|g(q)\| \leq \rho_3.$$

3. NHIỆM VỤ THIẾT KẾ ĐIỀU KHIỂN

Nhiệm vụ của bài toán điều khiển là tìm qui luật của lực/mômen do các động cơ điện tạo ra tác dụng lên các khâu để đảm bảo rôbôt chuyển động theo qui luật mong muốn $q_d(t)$ cho trước, nhằm thực hiện một nhiệm vụ nào đó. Trên cơ sở chuyển động mong muốn $q_d(t)$ được xác định trước và chuyển động hiện tại $q(t)$ của các khớp rôbôt được đo bởi các cảm biến đặt tại các khớp, bộ điều khiển có nhiệm vụ tính toán và đưa ra các lực/mômen cần thiết. Các lực/mômen này tác động làm cho rôbôt thực hiện chuyển động mong muốn một cách ổn định và chính xác. Sơ đồ khối của một bộ điều khiển cho rôbôt có dạng như Hình 1.



Hình 1. Sơ đồ điều khiển của rôbôt

Để có được luật điều khiển đáp ứng các yêu cầu vừa nêu, luật điều khiển dựa trên động lực học ngược thường được sử dụng. Với luật điều khiển này lực/mômen của bộ phận dẫn động được tính như sau:

$$u = M(q)v + C(q, \dot{q})\dot{q} + g(q),$$

trong đó

$$v = \ddot{q}_d - K_D(\dot{q} - \dot{q}_d) - K_P(q - q_d). \quad (3)$$

Thay (3) và (2) vào phương trình (1) ta được

$$M(q)[(\ddot{q} - \ddot{q}_d) + K_d(\dot{q} - \dot{q}_d) + K_p(q - q_d)] = d. \quad (4)$$

Do ma trận khối lượng $M(q)$ là ma trận xác định dương, nên từ hệ thức (4) ta có

$$[(\ddot{q} - \ddot{q}_d) + K_d(\dot{q} - \dot{q}_d) + K_p(q - q_d)] = M^{-1}d. \quad (5)$$

Nếu đặt $e(t) = q(t) - q_d(t)$ là sai số giữa chuyển động thực của rôbôt và chuyển động mong muốn, ta có

$$\ddot{e} + K_d\dot{e} + K_p e = M^{-1}d. \quad (6)$$

Sự ổn định của hệ phương trình vi phân tuyến tính (6) phụ thuộc vào sự ổn định của hệ phương trình vi phân tuyến tính thuần nhất tương ứng

$$\ddot{e} + K_d\dot{e} + K_p e = 0. \quad (7)$$

Các ma trận K_D và K_P cần được chọn sao cho nghiệm $e(t)$ của phương trình vi phân (7) tiến tới không khi thời gian t tăng đủ lớn. Nếu chọn K_D và K_P là các ma trận đường chéo thì phương trình (7) tương đương với hệ các phương trình tách rời nhau:

$$\ddot{e}_i + k_{di}\dot{e}_i + k_{pi}e_i = 0, \quad (i = 1, \dots, n). \quad (8)$$

Để thuận tiện cho việc chọn thông số ta đặt

$$k_{pi} = \omega_i^2, \quad k_{di} = 2\delta_i, \quad (9)$$

khi đó hệ phương trình vi phân (8) có dạng

$$\ddot{e}_i + 2\delta_i \dot{e}_i + \omega_i^2 e_i = 0, \quad (i = 1, \dots, n). \quad (10)$$

Nghiệm của phương trình trên có dạng

$$e_i(t) = A_i e^{-\delta_i t} \sin(\omega_i t + \alpha_i), \quad (i = 1, \dots, n). \quad (11)$$

Khi áp dụng luật điều khiển (2) thì ta cần phải biết chính xác các tham số hệ có mặt trong các đại lượng $M(q)$, $C(q, \dot{q})$ và $g(q)$ như khối lượng, vị trí khối tâm, mômen quán tính khối, ... Việc xác định các tham số này không đơn giản đặc biệt khi rôbốt đã được lắp đặt sẵn. Để tránh những khó khăn này ở đây một phương pháp điều khiển bền vững được khai thác sử dụng.

Giả sử gọi s là sai số suy rộng được xác định bởi phương trình

$$s = \dot{e}(t) + \Gamma e(t), \quad (12)$$

trong đó

$$\Gamma = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\}, \quad \lambda_i > 0, \quad (13)$$

$$e(t) = q(t) - q_d(t). \quad (14)$$

Rõ ràng là khi $s_i = 0$, thì $e_i(t) = e_i(0)e^{-\lambda_i t}$ sẽ tiến về 0 khi thời gian t tăng đủ lớn. Bây giờ ta tìm luật điều khiển để đảm bảo $s_i = 0$ hoặc s_i tiến về 0.

Đặt $\dot{q}_r = \dot{q}_d - \Gamma e(t)$ ta có

$$s = \dot{q}(t) = \dot{q}_r(t) \Rightarrow \dot{s} = \ddot{q}(t) = \ddot{q}_r(t). \quad (15)$$

Để tìm luật điều khiển ta chọn hàm Lyapunov như sau

$$V = \frac{1}{2} s^T M(q) s. \quad (16)$$

Đạo hàm V theo thời gian t ta được

$$\dot{V} = s^T M(q) \dot{s} + \frac{1}{2} s^T \dot{M}(q) s. \quad (17)$$

Từ (1) và chú ý đến (15) ta có

$$M(q) \dot{s} = \tau - C(q, \dot{q}) \dot{q} - g(q) - M(q) \ddot{q}_r(t) - d. \quad (18)$$

Cũng từ (15) suy ra $\dot{q}(t) = s + \dot{q}_r(t)$, nên ta có

$$C(q, \dot{q}) \dot{q} = C(q, \dot{q}) s + C(q, \dot{q}) \dot{q}_r.$$

Cuối cùng ta nhận được

$$M(q) \dot{s} = \tau - C(q, \dot{q}) s - M(q) \ddot{q}_r - C(q, \dot{q}) \dot{q}_r - g(q) - d. \quad (19)$$

Thay hệ thức (19) vào (17) nhận được

$$\dot{V} = s^T [\tau - M(q) \ddot{q}_r - C(q, \dot{q}) \dot{q}_r - g(q) - d] + s^T [\frac{1}{2} \dot{M}(q) - C(q, \dot{q})] s. \quad (20)$$

Chú ý rằng $s^T [\frac{1}{2} \dot{M}(q) - C(q, \dot{q})] s = 0$. Từ (20) ta chọn luật điều khiển như sau

$$\tau = \hat{M}(q) \ddot{q}_r - \hat{C}(q, \dot{q}) \dot{q}_r + \hat{g}(q) - K_{pd} s - K_s \text{sgn}(s). \quad (21)$$

Ở đây $sgn(s) = [sgn(s_1), sgn(s_2), \dots, sgn(s_n)]^T$, K_{pd} và K_s là các ma trận thực đối xứng xác định dương, $K_{pd} = K_{pd}^T > 0$, $K_s = K_s^T > 0$.

Để đơn giản ta chọn hai ma trận này có dạng đường chéo

$$K_{pd} = \text{diag}\{k_{pd}^{11}, k_{pd}^{22}, \dots, k_{pd}^{nn}\}, \quad K_s = \text{diag}\{k_s^{11}, k_s^{22}, \dots, k_s^{nn}\}.$$

Với cách chọn (21), hệ thức (20) trở thành

$$\begin{aligned} \dot{V} &= -s^T K_{pd} s - s^T K_s sgn(s) - s^T d - s^T [\tilde{M}(q)\ddot{q}_r + \tilde{C}(q, \dot{q})\dot{q}_r + \tilde{g}(q)] \\ &= -\sum_{i=1}^n K_{pd}^{(ii)} s_i^2 - \sum_{i=1}^n K_s^{(ii)} |s_i| - \sum_{i=1}^n s_i (d_i + \rho_i) \leq -\sum_{i=1}^n K_{pd}^{(ii)} s_i^2 - \sum_{i=1}^n (K_s^{(ii)} - |d_i + \rho_i|) |s_i|. \end{aligned}$$

Ở đây các sai lệch giữa mô hình thực sử dụng trong (1) và các thông số mô hình sử dụng trong bộ điều khiển (21) như sau

$$\tilde{M}(q) = M(q) - \hat{M}(q),$$

$$\tilde{C}(q, \dot{q}) = C(q, \dot{q}) - \hat{C}(q, \dot{q}),$$

$$\tilde{g}(q) = g(q) - \hat{g}(q),$$

và

$$\rho_i = [\tilde{M}(q)\ddot{q}_r + \tilde{C}(q, \dot{q})\dot{q}_r + \tilde{g}(q)]_i, \quad (i = 1, \dots, n).$$

Như thế để đảm bảo $\dot{V} \leq 0$ thì ta phải chọn các phần tử của K_s sao cho $K_s^{(ii)} - |d_i + \rho_i| > 0$ hay $K_s^{(ii)} > |d_i + \rho_i| > 0$.

4. ĐỘNG HỌC NGƯỢC RÔBÔT DƯ DẪN ĐỘNG

4.1 Đặt bài toán và các công thức cơ bản

Xét một rôbôt có n bậc tự do, gọi $q \in \mathbb{R}^n$ là vectơ chứa các tọa độ khớp. Bàn kẹp của rôbôt hoạt động trong không gian thao tác, gọi $x \in \mathbb{R}^m$ là vectơ chứa các tọa độ vị trí của nó trong không gian thao tác \mathbb{R}^m ($m \leq 6$). Kết quả của bài toán động học thuận cho biết

$$x = f(q), \quad x, f \in \mathbb{R}^m, q \in \mathbb{R}^n. \quad (22)$$

Khi $m < n$, rôbôt có số bậc tự do lớn hơn số tọa độ xác định vị trí của bàn kẹp, rôbôt được gọi là dư dẫn động. Đạo hàm 2 vế của (22) theo thời gian, ta suy ra

$$\dot{x} = \frac{\partial f}{\partial q} \dot{q} = J(q)\dot{q}, \quad (23)$$

trong đó

$$J(q) = \frac{\partial f}{\partial q} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \dots & \frac{\partial f_1}{\partial q_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_m}{\partial q_1} & \frac{\partial f_m}{\partial q_2} & \dots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \quad (24)$$

Ma trận Jacobi $J(q)$ cỡ $m \times n$. Giả sử hạng của $J(q)$ là m . Theo các tài liệu (Spong, Hutchinson, and Vidyasagar, 2006, Murray, Li, Sastry, 1994) ta chọn

$$J^+(q) = J^T(q)[J(q)J^T(q)]^{-1}. \quad (24)$$

Ma trận $J^+(q)$ được gọi là ma trận tựa nghịch đảo của ma trận $J(q)$. Từ (23) ta có

$$\dot{q} = J^+(q)\dot{x}. \quad (25)$$

Đạo hàm 2 vế của (23) theo thời gian thu được

$$\ddot{x} = \dot{J}(q)\dot{q} + J(q)\ddot{q}. \quad (26)$$

Từ (26) suy ra:

$$J(q)\ddot{q} = \ddot{x} - \dot{J}(q)\dot{q}. \quad (27)$$

Thế (25) vào (27) thu được

$$J(q)\ddot{q} = \ddot{x} - \dot{J}(q)J^+(q)\dot{x}.$$

Và từ đó ta suy ra

$$\ddot{q} = J^+(q)[\ddot{x} - \dot{J}(q)J^+(q)\dot{x}]. \quad (28)$$

Các công thức (25) và (28) cho phép ta xác định được vector vận tốc suy rộng và vector gia tốc suy rộng, nếu như biết $q(t)$ tại thời điểm khảo sát và các quy luật $x(t), \dot{x}(t), \ddot{x}(t)$.

Nếu ta chia khoảng thời gian rôbốt làm việc $[0, T]$ thành n^* khoảng thời gian bằng nhau

$$h = \Delta t = \frac{T}{n^*} \Rightarrow n^* = \frac{T}{h} \text{ thì } (k = 0, 1, \dots, n^* - 1).$$



Áp dụng khai triển Taylor ta có

$$q(t_{k+1}) = q(t_k + \Delta t) = q(t_k) + \dot{q}(t_k)\Delta t + \frac{1}{2}\ddot{q}(t_k)\Delta t^2 + \dots \quad (29)$$

Thế biểu thức (25) vào (29) và bỏ qua các vô cùng bé bậc lớn hơn 2 ta được

$$q(t_{k+1}) = q(t_k) + J^+(q(t_k))\dot{x}(t)\Delta t, \quad (k = 0, 1, \dots, n^* - 1). \quad (30)$$

Từ đó, ta có sơ đồ các bước tính toán như sau:

- 1) Tìm q_0 (Dùng phương pháp lặp xem phần 2), sau đó tính được $J(q_0), J^+(q_0), \dot{J}(q_0)$. Theo (25) tìm được $\dot{q}(0) = \dot{q}_0$, theo (28) tìm được $\ddot{q}(0) = \ddot{q}_0$.
- 2) Cho $k = 0, 1, 2, \dots, n^* - 1$, tính $q(t_{k+1})$ theo (30), theo (25) và (28) tìm được $\dot{q}(t_{k+1}) = \dot{q}_{k+1}$, $\ddot{q}(t_{k+1}) = \ddot{q}_{k+1}$.

Nhận xét: Tính $q(t_{k+1})$ theo (30) là khá thô. Vì vậy để nâng cao độ chính xác cần phải nêu ra một thuật toán xác định $q(t_{k+1})$ chính xác hơn.

4.2. Thuật toán xác định chính xác hơn các vector tọa độ trạng thái $q(t_k)$

a) *Thuật toán xác định $q_0 = q(t_0)$*

Giả sử $t_0 = 0$ là thời điểm đầu. Ta có thể xác định giá trị gần đúng \tilde{q}_0 của q_0 bằng cách vẽ (hoặc bằng thực nghiệm). Sau đó áp dụng khai triển Taylor tìm gần đúng tốt hơn của q_0 như sau.

Giả sử lấy $q(t_0) = \tilde{q}_0 + \Delta q_0$. Theo phương trình (22) ta có

$$x(t_0) = f(q_0) = f(\tilde{q}_0 + \Delta q_0) = f(\tilde{q}_0) + \frac{\partial f}{\partial q}(\tilde{q}_0)\Delta q_0 + \dots$$

Từ đó suy ra

$$J(\tilde{q}_0)\Delta q_0 \approx x(t_0) - f(\tilde{q}_0). \quad (31)$$

Giải phương trình (31), ta được $\Delta q_0 = J^+(\tilde{q}_0)[x(t_0) - f(\tilde{q}_0)]$. Sau đó lấy

$$\tilde{q}_0 := \tilde{q}_0 + \Delta q_0. \quad (32)$$

Nếu $\|\Delta q_0\| \geq \varepsilon_1$ thì ta lại tiếp tục thế (22) vào phương trình (21) và lại giải phương trình này. Quá trình lặp này được dừng tại $\|\Delta q_0\| < \varepsilon_1$.

Sau đó lấy

$$q_0 := \tilde{q}_0. \quad (33)$$

Biết được nghiệm q_0 của phương trình (22) tại thời điểm $t_0 = 0$. Thay vào (25), (28) ta tìm được $\dot{q}(0) = \dot{q}_0$ và $\ddot{q}(0) = \ddot{q}_0$.

Giá trị sai số cho phép ε_1 xác định theo yêu cầu về độ chính xác của bài toán.

b) *Thuật toán xác định* $q_{k+1} = q(t_{k+1})$, ($k = 0, 1, \dots, n^* - 1$)

Áp dụng khai triển Taylor, ta có

$$q(t_{k+1}) = \tilde{q}(t_{k+1}) + \Delta q(t_{k+1}), \quad (34)$$

trong đó $\tilde{q}(t_{k+1})$ được xác định gần đúng bởi công thức (30)

$$\tilde{q}(t_{k+1}) = q(t_k) + J^+(q(t_k))\dot{x}(t_k)\Delta t. \quad (35)$$

Thế (34) vào phương trình (22) ta được

$$x(t_{k+1}) = f(q(t_{k+1})) = f[\tilde{q}(t_{k+1}) + \Delta q_{k+1}] = f[\tilde{q}(t_{k+1})] + \frac{\partial f}{\partial q}[\tilde{q}(t_{k+1})]\Delta q_{k+1} + \dots$$

Từ đó suy ra

$$J[\tilde{q}(t_{k+1})]\Delta q_{k+1} \approx x(t_{k+1}) - f[\tilde{q}(t_{k+1})]. \quad (36)$$

Chú ý rằng $J[\tilde{q}(t_{k+1})]$ là ma trận cỡ $m \times n$, Δq_{k+1} , là vector có n phần tử. Do đó số ẩn lớn hơn số phương trình, nghiệm có chuẩn nhỏ nhất của phương trình (36) có dạng ($m < n$)

$$\Delta q_{k+1} = J^+(\tilde{q}_{k+1})[x(t_{k+1}) - f(\tilde{q}_{k+1})]. \quad (37)$$

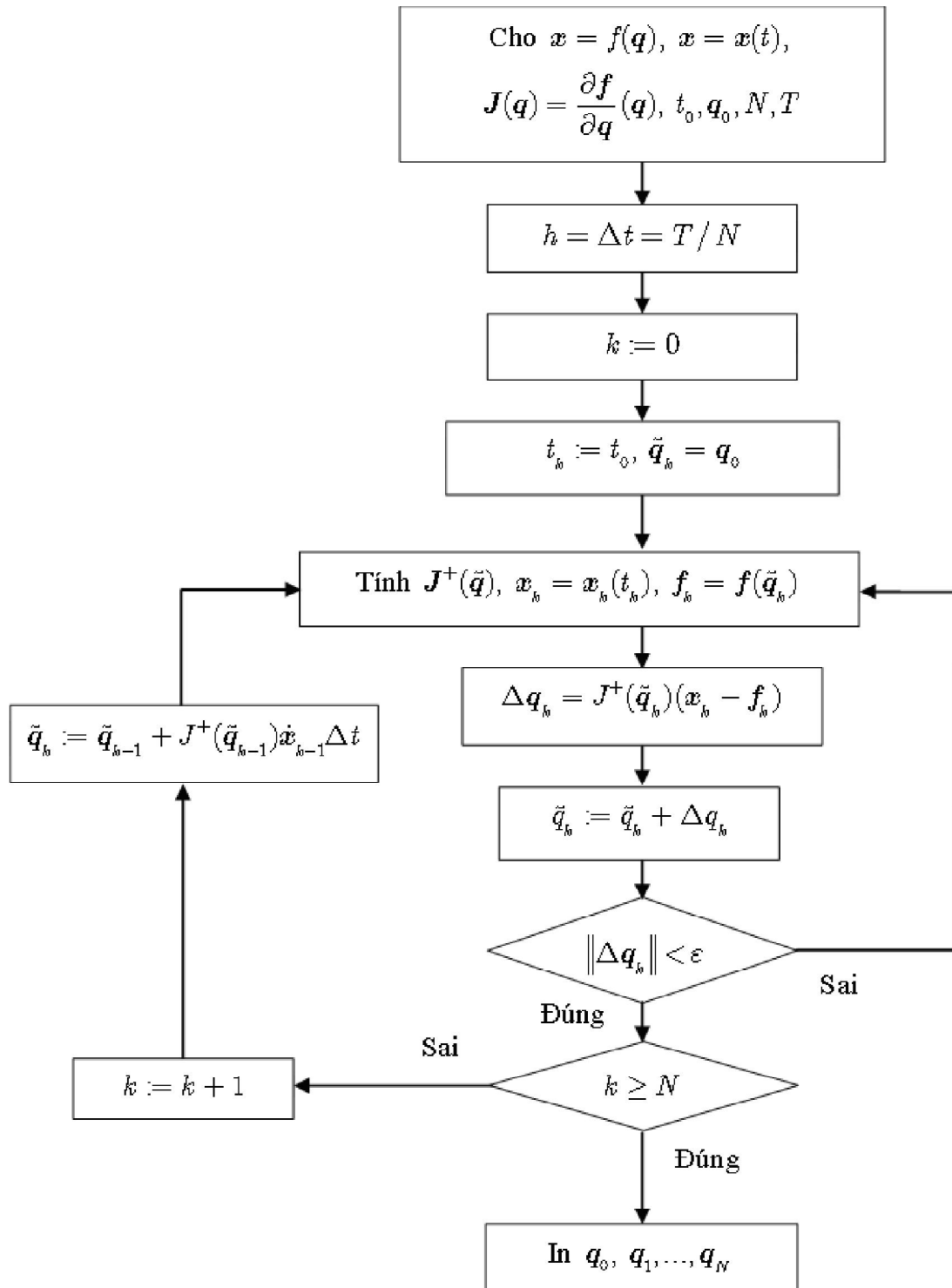
Sau đó lấy

$$\tilde{q}(t_{k+1}) := \tilde{q}(t_{k+1}) + \Delta q_{k+1}, \quad (38)$$

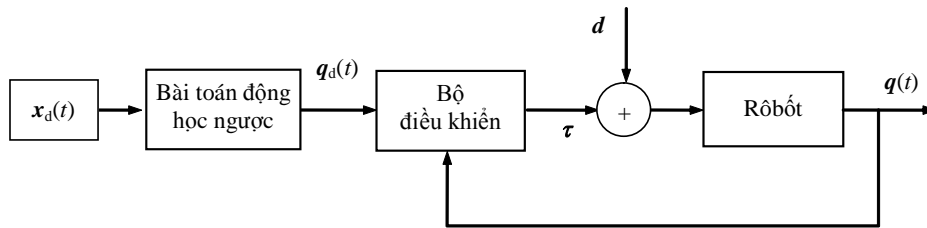
Nếu $\|\Delta q_{k+1}\| \geq \varepsilon_2$ thì thay (38) vào (36) và tiếp tục giải phương trình (36). Nếu $\|\Delta q_{k+1}\| < \varepsilon_2$ thì ta lấy

$$q(t_{k+1}) := \tilde{q}(t_{k+1}). \quad (39)$$

Biết được $q(t_{k+1})$, ta tính $\dot{q}(t_{k+1})$ theo công thức (25) và $\ddot{q}(t_{k+1})$ theo công thức (28). Giá trị sai số cho phép ε_2 xác định theo yêu cầu về độ chính xác của bài toán. Sơ đồ khối giải bài toán động học ngược rôbôt dư dẫn động được trình bày trên Hình 2.



Hình 2. Sơ đồ khối giải bài toán động học ngược

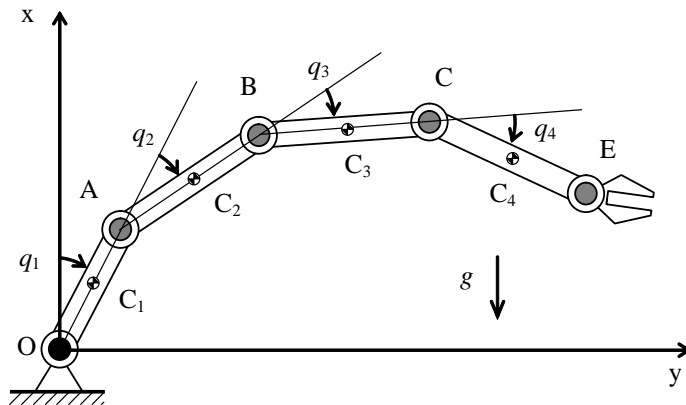


Hình 3. Sơ đồ điều khiển của rôbốt trong không gian thao tác

Như vậy, nếu cho biết chuyển động của bàn kẹp trong không gian thao tác thì ta cần bổ sung bài toán động học ngược vào sơ đồ điều khiển hình 1. Trong trường hợp này, sơ đồ điều khiển một rôbốt trong không gian thao tác có thể được thực hiện theo sơ đồ như trên Hình 3.

5. MÔ PHỎNG SỐ

Để minh họa luật điều khiển vừa trình bày, một mô phỏng số trong phần mềm Matlab được thực hiện. Ta khảo sát một tay máy phẳng dư dẫn động bốn bậc tự do chuyển động trong mặt phẳng thẳng đứng. Các khâu lần lượt có chiều dài L_i , khối tâm C_i , khoảng cách từ khối tâm đến khớp quay nối với vật trước đó là a_i , khối lượng m_i , và có mômen quán tính đối với trục qua khối tâm J_i . Mômen của các động cơ đặt trên các khớp tương ứng là τ_i ($i = 1, \dots, 4$). Sơ đồ rôbốt và các thông số của rôbốt được đưa ra trong Hình 4.



Khâu	1	2	3	4
m [kg]	10	5	2	1
I_z [kgm ²]	0.11	0.10	0.03	0.02
l [m]	0.25	0.30	0.45	0.40
a [m]	0.15	0.16	0.19	0.20

Hình 4. Rôbốt 4 khâu phẳng và các thông số của nó

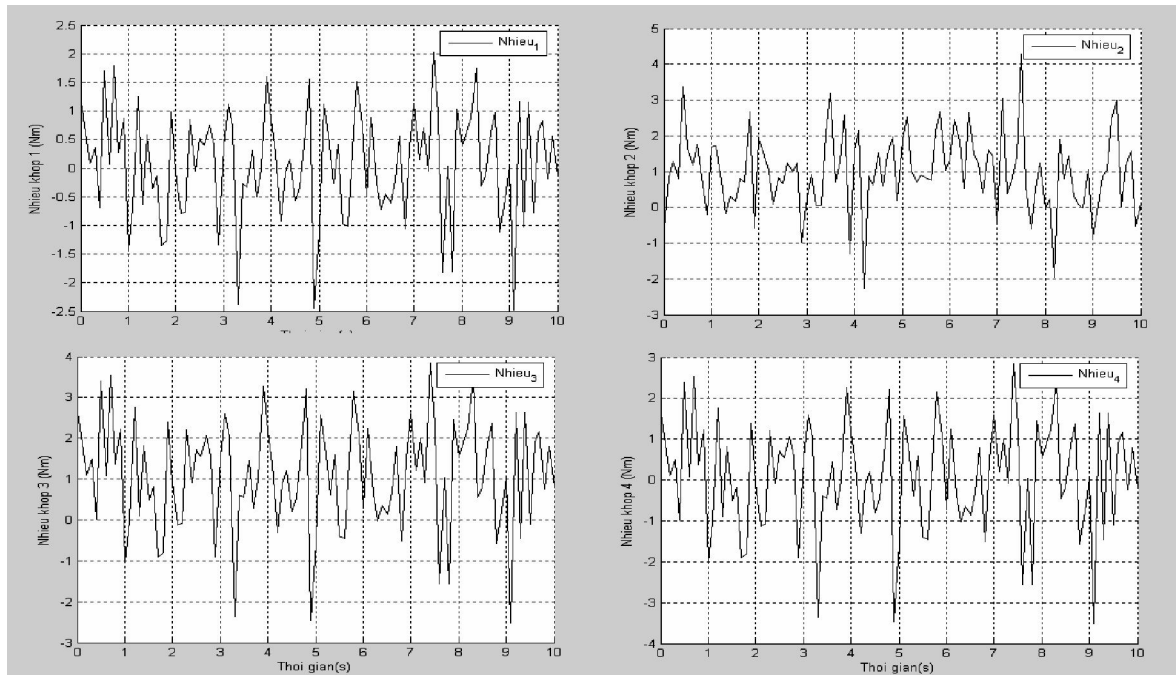
Phương trình vi phân chuyển động của rôbốt được thiết lập nhờ phần mềm xử lý ký tự Maple. Để kiểm chứng tính bền vững của luật điều khiển thiết kế, ta lấy các thông số rôbốt sử dụng trong bộ điều khiển khác với thông số trong mô hình là 10%. Ngoài ra rôbốt còn bị nhiễu động ngoài có dạng kích động như trình bày trên Hình 5.

Chọn bước thời gian cho bài toán động học ngược và chu kỳ lấy mẫu của bộ điều khiển $T_c = 0.001$ s, sai số cho phép trong bài toán động học ngược $\varepsilon = 10^{-6}$ rad. Các thông số của bộ điều khiển được chọn như sau

$$\Gamma = \text{diag}(100, 100, 50, 50), K_s = \text{diag}(30, 10, 10, 5),$$

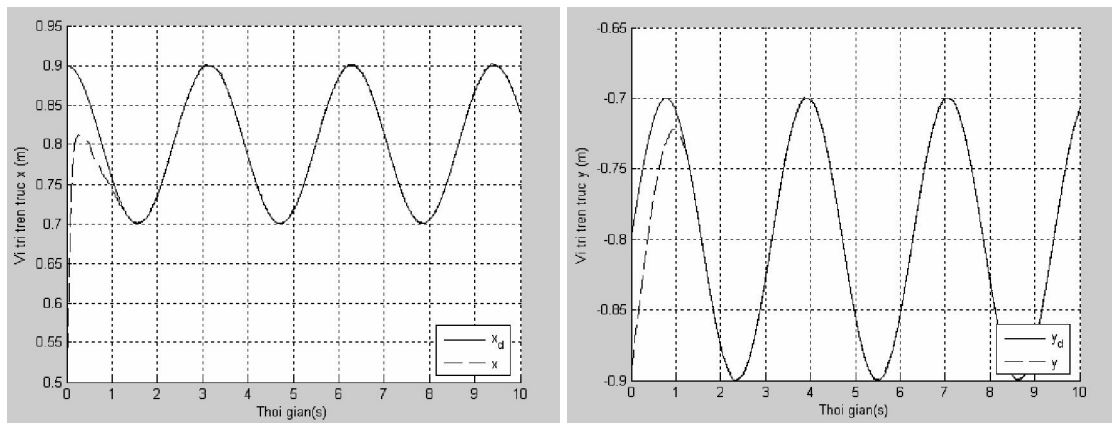
$$K_{pd} = \text{diag}(2, 2, 2, 2), \xi = 0, 2.$$

Ở đây, ta có thêm khâu bão hoà vào bộ điều khiển với mục đích giới hạn mômen điều khiển của động cơ luôn nằm trong giới hạn $-100 \dots +100$ Nm.

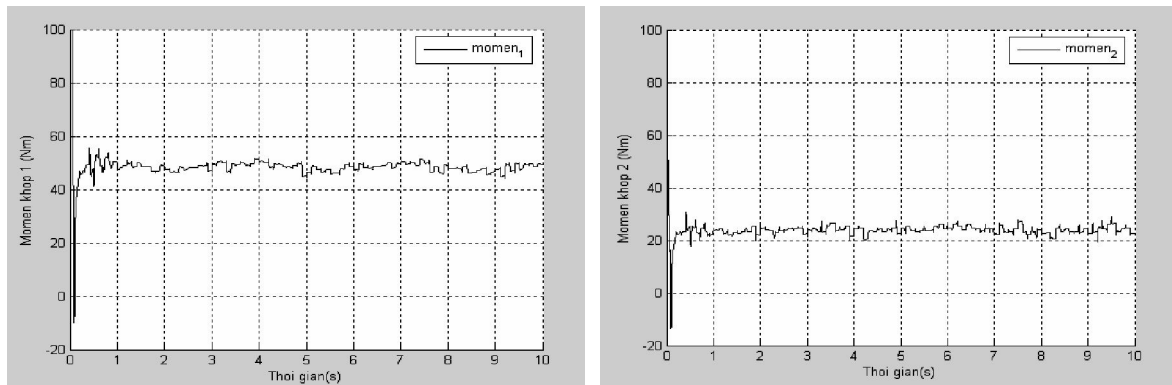


Hình 5. Đồ thị nhiễu $d(t)$ theo thời gian

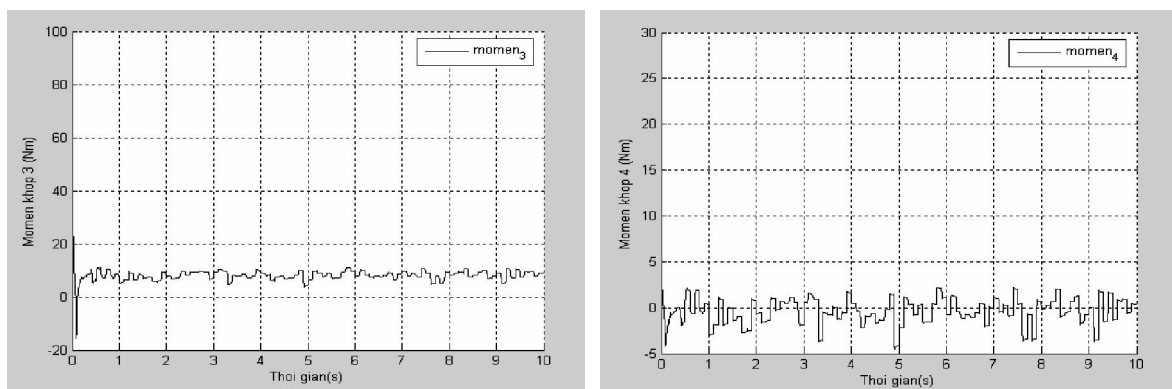
Các kết quả mô phỏng được thể hiện dưới dạng đồ thị theo thời gian. Trên hình 6 là tọa độ bàn kẹp thực tế và mong muốn. Bàn kẹp đã đi vào quỹ đạo mong muốn chỉ sau khoảng 1.5 s. Các mômen điều khiển tại các khớp được trình bày trong hình 7.



Hình 6. Đồ thị tọa độ trong không gian thao tác x, y theo thời gian



Hình 7a. Mômen điều khiển tại các khớp



Hình 7b. Mômen điều khiển tại các khớp

6. KẾT LUẬN

Trong bài báo này đã trình bày việc thiết kế điều khiển bền vững cho rôbôt có cấu trúc dạng chuỗi. Với luật điều khiển này chất lượng điều khiển được đảm bảo mặc dù ta không biết được chính xác các tham số động lực của hệ. Bài toán động học ngược rôbôt dư dẫn động được giải quyết nhờ phương pháp ma trận tựa nghịch đảo. Với việc giải bài toán ngược này việc vận hành rôbôt trở nên thân thiện và dễ hiểu hơn với người sử dụng. Các kết quả mô phỏng với rôbôt dư dẫn động bốn bậc tự do đã minh chứng tính chính xác của phương pháp thiết kế.

TÀI LIỆU THAM KHẢO

- [1] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, John Wiley & Sons, New York, 2006.
- [2] L. Sciavicco, B. Siciliano, *Modelling and Control of Robot Manipulators*, 2nd Edition, Springer-Verlag, London, UK, 2000.
- [3] R. V. Patel, F. Shadpey, *Control of Redundant Robot Manipulators*, Springer-Verlag, Berlin, 2005.

- [4] R. M. Murray, Z. Li, S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Raton, Fla. [u.a.], 1994.
- [5] J. E. Slotine, *Applied Nonlinear Control*, Prentice Hall, Englewood Cliffs, NJ, 1991.
- [6] F. L. Lewis, D. M. Dawson, C. T. Abdallah, *Robot Manipulator Control/ Theory and Practice*, 2nd Edition, Marcel Dekker, New York 2004.
- [7] Y. Nakamura, *Advanced Robotics/Redundancy and Optimization*, Addison-Wesley Publishing Company, Reading 1991.
- [8] V. Parra-Vega, S. Arimoto, Nonlinear PID control with sliding modes for tracking of robot manipulators, *Proceedings of the 2001 IEEE International Conference on Control Applications*, 2001 (351–356).
- [9] S. J. Hills and D. R. Yoerger, A nonlinear sliding mode autopilot for unmanned undersea vehicles, *Proceedings of the IEEE Oceans Conference*, 1994 (93–98).
- [10] P. Hsu, J. Hanser, and S. Sastry, Dynamic control of redundant manipulators, *Journal of Robotic Systems* (6) (1989) 133–148.
- [11] Nguyễn Văn Khang, *Động lực học hệ nhiều vật*, NXB Khoa học và Kỹ thuật 2007.
- [12] Nguyễn Văn Khang, Lê Đức Đạt, Trần Hoàng Nam, Về một thuật toán giải bài toán động học ngược rôbot dạng chuỗi, *Tuyển tập Hội nghị Cơ học toàn quốc lần thứ VIII*, Tập 1, Hà Nội 2008.
- [13] Nguyễn Quang Hoàng, Nguyễn Văn Khang, Về một thuật toán điều khiển trượt rôbot công nghiệp, *Tuyển tập Hội nghị Cơ học toàn quốc lần thứ VIII*, Tập 1, Hà Nội 2008.

Nhận bài ngày 19 - 2 - 2008