TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

o0o

ĐỒ ÁN TỐT NGHIỆP

ĐIỀU KHIỂN TỐI ƯU THÍCH NGHI CHO QUADROTOR VỚI MÔ HÌNH KHÔNG XÁC ĐỊNH

NGUYỄN TẤT CHUNG
chung.nt170130@sis.hust.edu.vn
PHẠM ĐÌNH DƯƠNG
duong.pd3800@sis.hust.edu.vn

Ngành KT Điều khiển và Tự động hóa
Chuyên ngành Điều khiển tự động

Giảng viên hướng dẫn: **TS. Đào Phương Nam**

Chữ ký của GVHD

Bộ môn: **Điều khiển tự động**
Viện: **Điện**

HÀ NỘI, 06/2021

BỘ GIÁO DỤC và ĐÀO TẠO **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

TRƯỜNG ĐH BÁCH KHOA HÀ NỘI **Độc lập – Tự do - Hạnh phúc**

NHIỆM VỤ
ĐỒ ÁN TỐT NGHIỆP

Họ và tên sinh viên: PHẠM ĐÌNH DƯƠNG
Khóa 62 Viện: Điện Ngành: CN ĐK và TĐH

1. Tên đề tài:
   Nghiên cứu thuật toán Data-driven PI tối ưu thích nghi áp dụng cho bài toán điều khiển bám Quadrotor

2. Nội dung đề tài:

   - Nghiên cứu bài toán điều khiển bám tối ưu cho hệ Affine tổng quát sử dụng thuật toán Data-driven PI

   - Áp dụng thuật toán Data-driven PI đối với hệ tuyến tính và phi tuyến, ứng dụng điều khiển bám tối ưu cho Quadrotor

   - Thực hiện mô phỏng kiểm chứng tính hiệu quả của thuật toán trên phần mềm Matlab

3. Cán bộ hướng dẫn: **TS. Đào Phương Nam**

4. Thời gian giao đề tài: 01/02/2021

5. Thời gian hoàn thành: 22/06/2021

Ngày..... tháng ..... năm 2021

LÃNH ĐẠO BỘ MÔN **CÁN BỘ HƯỚNG DẪN**

SINH VIÊN THỰC HIỆN
(Ký và ghi rõ họ tên)

Lời cảm ơn

"Khi ta ở, chỉ là nơi đất ở -Khi ta đi, đất đã hoá tâm hồn! - Chế Lan Viên". Khi viết những dòng này, cũng là lúc chúng em đã hoàn thành đồ án, đã hoàn thành những điều cuối cùng còn có thể thực hiện dưới mái trường mến yêu này. 4 năm học không phải khoảng thời gian quá ngắn, cũng chẳng quá dài, nhưng đủ sâu đậm để chúng em luôn nhớ mãi. Khoảnh khắc xúc động này, chúng em không biết nói gì ngoài những lời biết ơn. Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành nhất tới thầy Đào Phương Nam vì đã định hướng và tận tình hướng dẫn không chỉ trong quá trình thực hiện đồ án tốt nghiệp này mà còn trong suốt 4 năm học tập và nghiên cứu tại Đại học Bách Khoa Hà Nội. Chúng em xin cảm ơn các thầy cô ở Bộ môn Điều khiển tự động nói riêng và các thầy cô trường Đại học Bách khoa Hà Nội nói chung vì những tri thức và cả những bài học cuộc sống, là hành trang để giúp chúng em vững chãi khi bước ra đường đời. Chúng em cũng xin gửi lời cám ơn sâu sắc đến gia đình và bạn bè vì đã luôn sát cánh và động viên tinh thần trong suốt quãng thời gian đã qua.

Do thời gian và khả năng còn hạn chế, đồ án không thể tránh khỏi những nhầm lẫn và thiếu sót, chúng em rất mong nhận được sự góp ý của các thầy cô và bạn đọc để giúp đồ án trở nên hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

Tóm tắt nội dung đồ án

Trong nhiều thập kỷ qua, điều khiển tối ưu đã và đang được nghiên cứu, ứng dụng sâu rộng trong lĩnh vực điều khiển, tự động hóa. Việc giải bài toán điều khiển tối ưu thường được đưa về việc giải nghiệm của phương trình Halmilton-Jacobi- Bellman (HJB). Tuy nhiên, đây là phương trình vi phân phi tuyến khá phức tạp, không có nghiệm giải tích. Ý tưởng được đưa ra và nhận được phần lớn sự quan tâm nghiên cứu là các phương pháp xấp xỉ nghiệm phương trình HJB. Song song với đó, với sự phát triển của các thuật toán học củng cố (Reinforcement Learning), sự ra đời của Quy hoạch động thích nghi (Adaptive Dynamic Programming) với việc xấp xỉ nghiệm của phương trình HJB bằng mạng Neural với cấu trúc Actor-Critic đã mở ra nhiều hướng phát triển: On-policy với Online Actor-Critic, Online IRL,... Off-policy với Off-policy IRL, Data-Driven PI, ... Mỗi phương pháp có ưu và nhược điểm riêng của nó. Với On-policy, tuy yêu cầu thông tin của toàn bộ hoặc một phần hệ thống nhưng thuật toán được chỉnh định trực tiếp trong quá trình điều khiển. Với Off-policy, điểm mạnh là không yêu cầu toàn bộ thông tin động học của hệ, nhưng cần phải thu thập và xử lý dữ liệu trước khi áp dụng điều khiển. Ở trong khuôn khổ đồ án, chúng em tập trung nghiên cứu thuật toán Off-policy (Data-Driven) dựa trên PI để tìm bộ điều khiển tối ưu cho các hệ tuyến tính và phi tuyến không biết trước mô hình. Hơn nữa, trong khi các nghiên cứu phần lớn giải quyết bài toán tối ưu ổn định tiệm cận cho hệ, đồ án chúng em quan tâm đến vấn đề điều khiển bám tối ưu, từ đó đưa ra thêm thành phần hệ số suy giảm cho hàm chi phí. Thuật toán được phát triển tổng quát cho hệ Affine, với ví dụ được trình bày ở mỗi phần cho cả hệ tuyến tính và phi tuyến, đi kèm với mô phỏng để kiểm nghiệm cho thấy tính hiệu quả của thuật toán.

Nhằm cho thấy khả năng ứng dụng của thuật toán, chúng em áp dụng điều khiển cho đối tượng là máy bay 4 cánh (Quadrotor). Trong những năm gần đây, máy bay không người lái nhận được sự quan tâm lớn trong cộng đồng nghiên cứu nhờ tiềm năng ứng dụng to lớn của nó. Là một loại máy bay không người lái điển hình, quadrotors đã và đang trở nên ngày càng phổ biến nhờ khả năng cất cánh và tiếp đất theo phương thẳng đứng, khả năng ổn định vị trí và quỹ đạo linh hoạt. Nhiều phương pháp thiết kế bộ điều khiển được đưa ra cho quadrotor: bộ điều khiển PID, backstepping, sliding mode

control,... Tuy nhiên với thực tế rằng: quadrotor là một hệ phi tuyến 6 bậc tự do có tính xen kênh lớn, mô hình chính xác rất khó đảm bảo (do cấu trúc phức tạp, do khi hoạt động có gắn thêm thiết bị hoặc mang, kéo các vật thể khác không xác định), việc tìm bộ điều khiển không biết mô hình mà vẫn đảm bảo chất lượng điều khiển là rất cần thiết. Từ đó, chúng em nghiên cứu áp dụng thuật toán Data-driven PI giải bài toán điều khiển bám tối ưu cho quadrotor. Mô phỏng được tiến hành trên phần mềm MATLAB. Cuối cùng là nhận xét, và định hướng phát triển thuật toán trong tương lai.

Hà Nội, 16 tháng 6 năm 2021
Sinh viên

Nguyễn Tất Chung

Phạm Đình Dương

Một số kí hiệu viết tắt

$\|.\|$	Chuẩn trong không gian Euclid.
RL	Reinforcement Learning.
IRL	Integral Reinforcement Learning.
HJB	Hamilton–Jacobi–Bellman.
PI	Policy Iteration.
ADP	Approximate/Adaptive Dynamic Programming.
NN	Neural Network.
CNN	Critic Neural Network.
ANN	Actor Neural Network.
AC	Actor-Critic.
PE	Persistent Excitation Condition.
$e_{N,j}$	Vecto $N \times 1$ có phần tử thứ j bằng 1 và các phần tử còn lại bằng 0
$I_N \in \mathbb{R}^{N \times N}$	Ma trận đơn vị cấp N
$0_{m \times n} \in \mathbb{R}^{m \times n}$	Ma trận không kích thước $m \times n$
DOF	Bậc tự do (Degree of Freedom)

Danh sách hình vẽ

Mục lục

Chapter 1

Tổng quan về học tăng cường và quy hoạch động thích nghi

1.1 Sơ lược về học tăng cường (Reinforcement Learning)

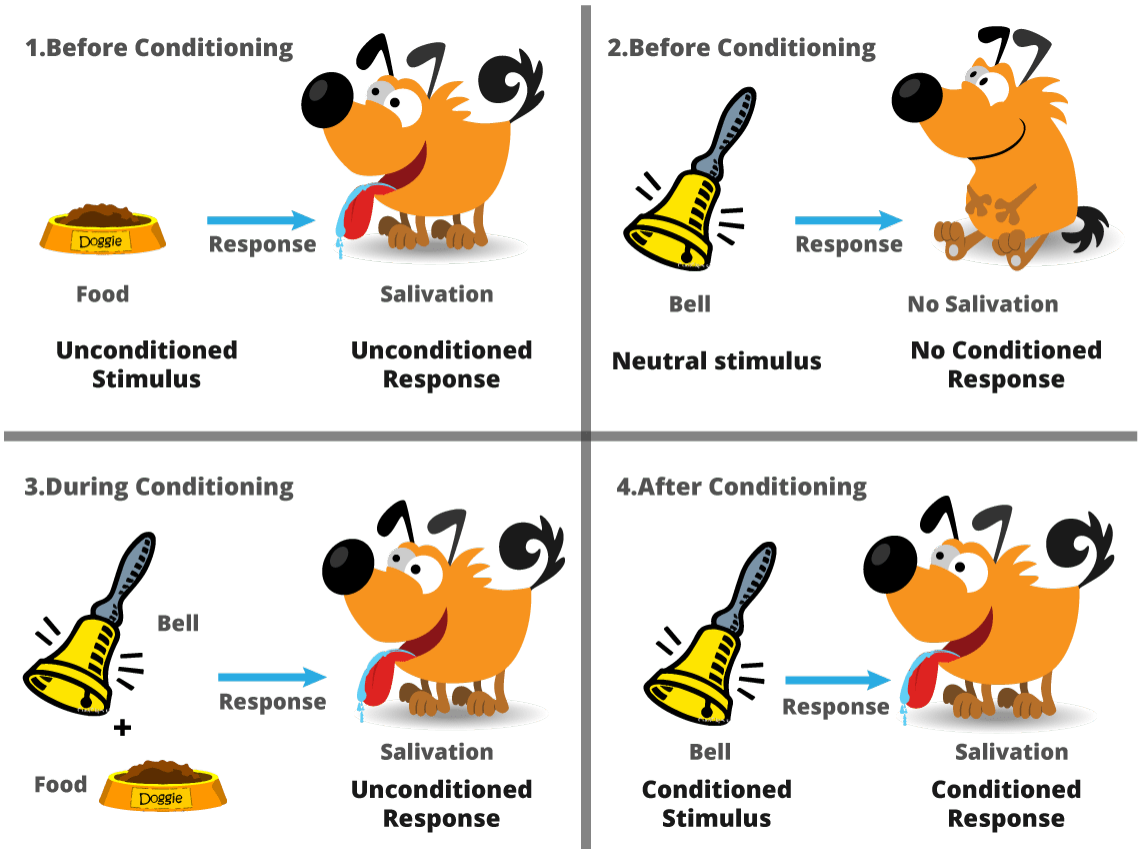


Figure 1.1: Ví dụ của học tăng cường ở động vật

Mọi cá thể sống đều tương tác với môi trường và sử dụng sự tương tác ấy để chỉnh định hành vi nhằm tồn tại và phát triển. Ta gọi sự chỉnh định hành vi dựa trên tương tác với môi trường ấy là học tăng cường (Reinforcement Learning) (hình 1.1). Học tăng cường liên quan đến việc một đối tượng (tác tử) tương tác với môi trường của nó, chỉnh sửa hành động (hay còn gọi là

"control policy") dựa trên những phản hồi của những hành động trước đó. Việc chỉnh định này được thực hiện nhờ vào những thông tin có thể đánh giá được từ môi trường. Nói cách khác, học tăng cường dựa trên mối quan hệ nguyên nhân- kết quả của hành vi và phần thưởng(hoặc hình phạt). Các thuật toán học tăng cường bắt đầu từ ý tưởng các hành vi thành công (đem lại phần thưởng cao) sẽ được ghi nhớ, theo nghĩa rằng chúng có khuynh hướng được sử dụng lại ở những lần sau [10]. Mặc dù ý tưởng về học tăng cường bắt nguồn từ những thí nghiệm về việc học của sinh vật, học tăng cường về mặt lý thuyết có sự kết nối chặt chẽ trực tiếp và gián tiếp với các phương pháp điều khiển tối ưu thích nghi.

Một trong những cấu trúc phổ biến của học tăng cường là cấu trúc Actor-Critic[Barto, Sutton, Anderson 1983], trong đó, một thành phần Actor thực hiện hành động (control policy) tác động đến môi trường, và một thành phần Critic đánh giá hành động đó (hình 1.2) . Dựa trên sự đánh giá đó, nhiều phương pháp được sử dụng để chỉnh định hoặc cải thiện hành động sao cho hành động mới sẽ tạo ra giá trị (phần thưởng) tốt hơn so với giá trị trước. Tóm lại, cấu trúc Actor-Critic gồm 2 bước: đánh giá hành vi và cải thiện hành vi. Việc đánh giá hành vi được thực hiện bởi việc quan sát kết quả trả về từ môi trường sau khi thực hiện hành vi.

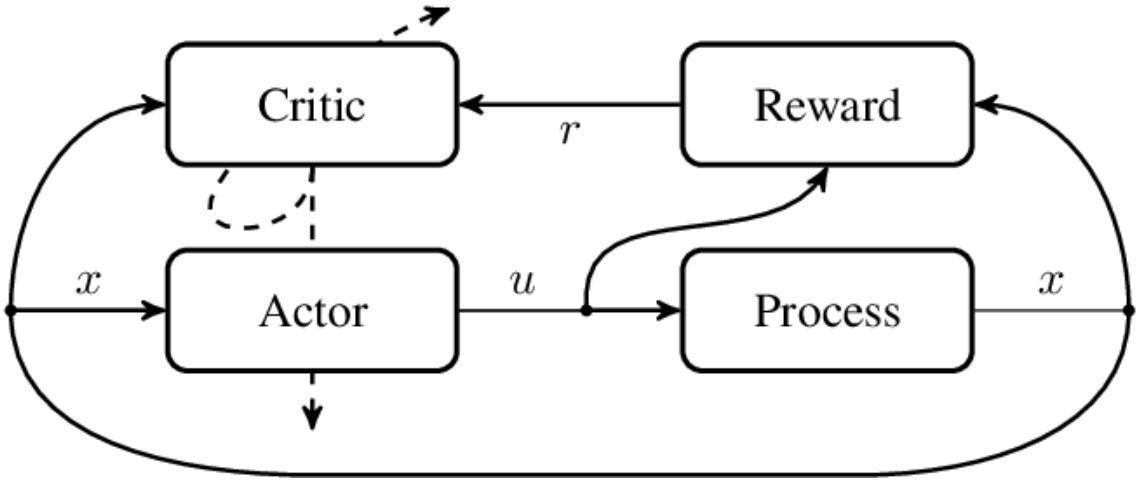


Figure 1.2: Cấu trúc Actor-Critic trong Reinforcement Learning

Trở lại với việc học từ tự nhiên, việc sinh tồn trong thiên nhiên là tối quan trọng và cũng cực kỳ khó khăn. Do đó, các cá thể thường phải cố gắng thực hiện những hành vi sao cho vừa giữ được năng lượng tối đa, vừa đạt được mục đích hành vi tối đa. Những hành vi như vậy được gọi là hành vi tối ưu. Hành vi tối ưu thường dựa trên những tiêu chuẩn sau: tối thiểu hóa chi phí năng lượng, rủi ro,... tối đa hóa phần thưởng,... Do đó, việc nghiên cứu học tăng cường với cấu trúc Actor-Critic trong đó Critic đánh giá hành vi dựa trên những tiêu chuẩn tối ưu là tự nhiên và cần thiết.

Điều khiển phản hồi là sự nghiên cứu các phương pháp thiết kế và phát triển

bộ điều khiển dựa trên phản hồi của hệ thống, nhằm đạt được chất lượng điều khiển và an toàn mong muốn. Các hệ thống này bao gồm: thiết bị bay, tàu, ô tô, hệ thống robot, các quá trình công nghiệp, các hệ thống điều chỉnh nhiệt độ, thời tiết, và rất nhiều các đối tượng khác nữa. Việc bắt chước, học theo tự nhiên để thiết kế điều khiển sao cho tối ưu (tối ưu thường được hiểu là đạt được mục đích điều khiển sao cho tiêu tốn ít tài nguyên nhất) là có cơ sở và cần thiết. Tiếp theo, luận văn xin được trình bày sơ lược về sự kết hợp giữa học tăng cường và điều khiển phản hồi: thuật toán quy hoạch động thích nghi.

1.2 Sơ lược về học quy hoạch động thích nghi - Adaptive Dynamic Programming

1.2.1 Nguyên lý quy hoạch động của Bellman

Nhà toán học Richard Bellman đã phát minh phương pháp "Quy hoạch động" vào năm 1957. Nguyên lý tối ưu của R.Bellman được phát biểu như sau [2]: "Tối ưu n bước bằng cách tối ưu tất cả con đường tiến đến bước $n-1$ và chọn con đường có tổng chi phí từ bước 1 đến bước $n-1$ và từ $n-1$ đến n là thấp nhất(phần thưởng nhiều nhất)".

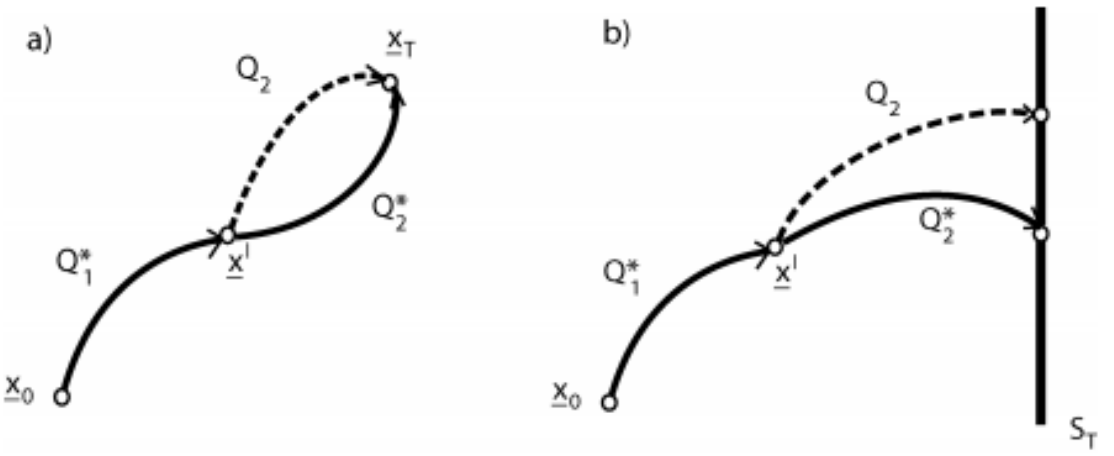


Figure 1.3: Nguyên lý tối ưu Bellman [17]

1.2.2 Phương trình HJB và quy hoạch động thích nghi

Xét hệ động học affine:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \tag{1.1}$$

Với $x \in \mathbb{R}^n$ là vector trạng thái hệ thống, $u =\in \mathbb{R}^m$ là tín hiệu điều khiển, $f(x(t)) \in \mathbb{R}^n$ và $g(x(t)) \in \mathbb{R}^{n\times m}$ Giả sử rằng $f(0) = 0$ và $f(x(t)) + g(x(t))u(x(t))$ thỏa mãn điều kiện Lipschitz liên tục trong một miền $\Omega \subseteq \mathbb{R}^n$. Xét hàm chi phí:

$$V(x) = \int_t^\infty r(x,u)d\tau \tag{1.2}$$

Với $r(x,u) = Q(x) + u^T Ru$. $Q(x)$ là hàm xác định dương, R là ma trận đối xứng xác định dương. Theo lý thuyết điều khiển tối ưu [17], tín hiệu điều khiển tối ưu nhằm ổn định hệ thống và tối thiểu hàm chi phí trên là:

$$u^*(x) = -\frac{1}{2}R^{-1}g^T(x)\frac{\partial V^*(x)}{\partial x} \tag{1.3}$$

Trong đó $V^*(\underline{x},t) = \min_{\underline{u}\in\mathbb{U}} \int_t^T r(\underline{x},\underline{u},\tau)d\tau$ Từ đó, ta có phương trình HJB cho hệ affine:

$$\begin{cases} H^*(\underline{x},\underline{u}^*,V_{\underline{x}}^*) = \frac{\partial V^*(\underline{x})}{\partial \underline{x}}\left(\underline{f}(\underline{x}) + \underline{g}(\underline{x})\underline{u}^*\right) + Q(\underline{x}) + \underline{u}^{*T}R\underline{u}^* = 0 \\ V^*(\underline{0}) = 0 \end{cases} \tag{1.4}$$

Nếu hệ thống là tuyến tính và hàm phạt có dạng quadratic với trạng thái và tín hiệu điều khiển, bộ điều khiển tối ưu có thể viết dưới dạng bộ điều khiển phản hồi trạng thái, với ma trận Gain là nghiệm của phương trình Riccati. Tuy nhiên, nếu hệ thống là phi tuyến, chúng ta phải giải phương trình HJB để tìm nghiệm tối ưu. Tuy nhiên, việc giải phương trình HJB thực sự khó khăn bởi tính phi tuyến của phương trình phi phân riêng phần này, hệ thống càng phi tuyến, phức tạp, phương trình càng khó giải.

Trong thời gian đầu, người ta sử dụng Quy hoạch động để giải bài toán tối ưu, điển hình là năm 1957 với sự ra đời của nguyên lý Bellman: Một luật điều khiển tối ưu có tính chất rằng với trạng thái đầu và quyết định đầu tiên bất kỳ thì những quyết định sau đó phải có tổng là tối ưu. Nguyên lý này có thể được biểu diễn đơn giản dưới dạng một công thức hồi quy. Do đó, nguyên lý này được ứng dụng khá rộng rãi, từ hệ rời rạc đến liên tục, từ hệ tuyến tính cho đến phi tuyến.

Tuy nhiên, việc sử dụng nguyên lý Quy hoạch động để giải phương trình HJB cho hệ phi tuyến có nhiều điểm yếu. Khi số chiều của trạng thái x và tín hiệu điều khiển u tăng thì việc triển khai thuật toán càng trở nên khó khăn hơn, thậm chí không thể giải do yêu cầu khối lượng tính toán backward cực kỳ lớn (vấn đề "curse of dimensionality"). Để giải quyết những điểm yếu này, Werbos đề xuất cấu trúc Adaptive Dynamic Programing (Quy hoạch động thích nghi), trong đó, ý tưởng chính là sử dụng xấp xỉ hàm(ví dụ mạng Neural, Fuzzy model, Polynomial,...) để xấp xỉ hàm chi phí và giải bài toán quy hoạch động một cách forward theo thời gian.

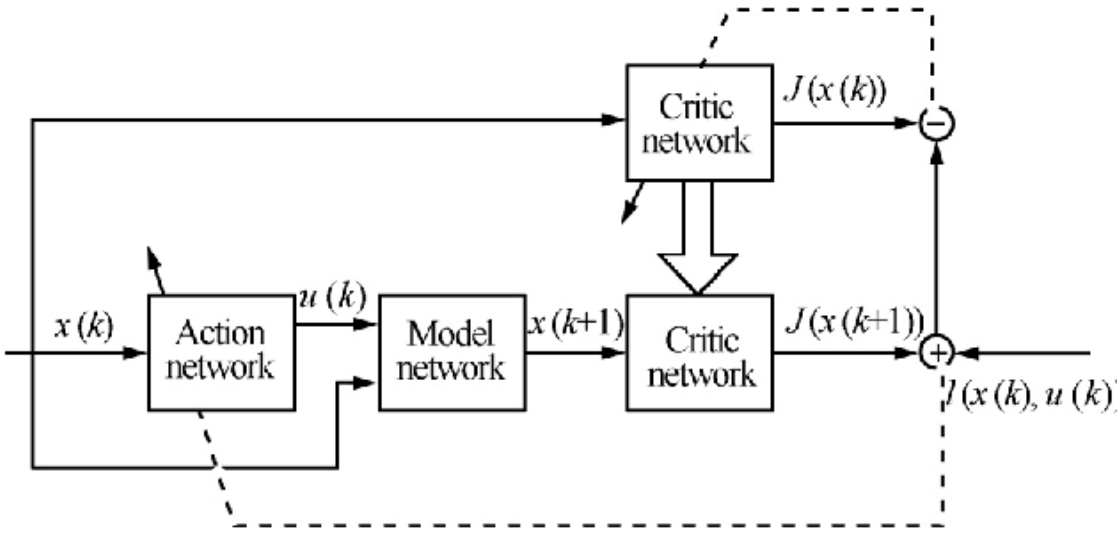


Figure 1.4: Cấu trúc Actor-Critic trong ADP

Thuật toán ADP được phát triển, sử dụng xấp xỉ hàm để xấp xỉ hàm chi phí (Critic) và luật điều khiển (Actor), cấu trúc nổi tiếng này được gọi ngắn gọi với tên Actor-Critic (hình 1.4). Việc cập nhật trọng số cho 2 xấp xỉ hàm này được thực hiện theo 2 cách, đồng thời cũng mở ra 2 kiểu thuật toán: On-policy và Off-policy. Với On-policy, luật điều khiển được cập nhật trực tiếp trong quá trình điều khiển. Một số phương pháp đã được phát triển trong những năm gần đây, có thể kế đến như: Online Actor-Critic cập nhật trọng số song song với việc cần biết rõ thông tin động học hệ thống hoặc sử dụng một bộ nhận dạng hệ thống (Identifier); Online Integral Reinforment Learning (IRL) loại bỏ được yêu cầu phải biết động học nội của hệ thống, nhưng vẫn cần biết thông tin của $g(x)$. Với Off-policy, hai luật điều khiển được tách riêng: một luật điều khiển chấp nhận được, với yêu cầu về nhiễu đảm bảo điều kiện PE, được dùng để tạo dữ liệu về động học của hệ thống, sau đó dữ liệu (chủ yếu là trạng thái x và tín hiệu điều khiển u) được train để tìm ra bộ điều khiển tối ưu. Điểm mạnh của thuật toán Off-policy là chúng ta hoàn toàn không cần thông tin động học của hệ thống, điều này được giải thích đơn giản: nhờ lượng dữ liệu lớn được thu thập trước đó, thực chất thông tin động học của hệ thống đã được gián tiếp thể hiện trong lượng dữ liệu này.

Trong lĩnh vực điều khiển tối ưu sử dụng thuật toán ADP, nghiên cứu tập trung chủ yếu phát triển các giải thuật cho bài toán điều khiển ổn định hệ thống (bài toán nhằm đưa trạng thái hệ thống tiệm cận về gốc 0). Không giống với bài toán điều khiển ổn định mà trong đó, tín hiệu điều khiển u tiến về 0 khi hệ ổn định, bài toán điều khiển bám yêu cầu giữ một tín hiệu điều khiển u khác 0 khi trạng thái hệ thống nằm trên quỹ đạo mong muốn. Một số nghiên cứu sử dụng cách tách bộ điều khiển thành 2 phần, một thành phần là tín hiệu điều khiển Feed-forward - tín hiệu cần để giữ hệ trên quỹ đạo, một

thành phần là tín hiệu đưa hệ về trạng thái ổn định dựa trên nguyên lý tối ưu [8] Tuy nhiên, ta dễ thấy phương pháp này chỉ là tối ưu một phần, chưa triệt để. Một cách xử lý tổng quát hơn là sử dụng hệ số suy giảm trong hàm chi phí, phương pháp phổ biến được sử dụng trong Reinforcement Learning ở các thuật toán Học Máy. Từ hàm phạt này, thuật toán ADP được triển khai để tìm ra bộ điều khiển tối ưu cho hệ thống. Trong đồ án này, chúng em tập trung nghiên cứu hướng xử lý này, xử dụng Off-policy ADP để giải bài toán tối ưu với hàm chi phí được sửa đổi bằng cách thêm hệ số suy giảm.

Bố cục còn lại của đồ án được thể hiện như sau: Chương 2 sẽ trình bày thuật toán Data-driven PI tổng quát cho hệ affine. Cụ thể, phần 2.1 giới thiệu vấn đề điều khiển bám tối ưu, phần 2.2 trình bày thuật toán Data-driven PI giải quyết bài toán ở 2.1, phần 2.3 đưa ra ví dụ áp dụng thuật toán cho hệ tuyến tính và phi tuyến, kèm theo mô phỏng kiểm chứng. Chương 3 trình bày ứng dụng của thuật toán Data-driven PI để tìm bộ điều khiển tối ưu cho đối tượng là Quadrotor. Cụ thể, phần 3.1 giới thiệu sơ lược về Quadrotor, phần 3.2 đưa ra mô hình động học của đối tượng, phần 3.3 nêu phương pháp điều khiển Quadrotor nói chung gồm 2 vòng điều khiển vị trí và góc hướng, phần 3.4 trình bày cụ thể thuật toán Data-driven PI cho 2 vòng điều khiển của Quadrotor, phần 3.5 đưa ra kết quả mô phỏng áp dụng thuật toán này. Cuối cùng là kết luận và định hướng nghiên cứu, phát triển đồ án trong tương lai.

Chapter 2

Thuật toán Data-driven PI cho bài toán Điều khiển bám tối ưu

2.1 Vấn đề điều khiển bám tối ưu (Optimal Tracking Control Problem)

Xét hệ affine:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \tag{2.1}$$

Ở đây $x = [x_1...x_n]^T \in \mathbb{R}^n$ là vector trạng thái hệ thống có thể đo được, $u = [u_1...u_n] \in \mathbb{R}^m$ là tín hiệu điều khiển, $f(x(t)) \in \mathbb{R}^n$ và $g(x(t)) \in \mathbb{R}^{n\times m}$ Giả sử rằng $f(0) = 0$ và $f(x(t)) + g(x(t))u(x(t))$ thỏa mãn điều kiện Lipschitz liên tục trong một miền $\Omega \subseteq \mathbb{R}^n$. Hệ phi tuyến affine trên có khả năng ổn định trong miền Ω, có nghĩa rằng tồn tại tín hiệu điều khiển $u(t)$ để hệ (2.1) ổn định tiệm cận trên miền Ω. Định nghĩa sai lệch bám:

$$e_d(t) = x(t) - x_d(t) \tag{2.2}$$

Với $x_d(t) \in \mathbb{R}^n$ là quỹ đạo trạng thái mong muốn. Hàm chi phí được định nghĩa như sau:

$$V(e_d(t), x_d(t)) = \int_t^\infty e^{-\lambda(\tau-t)}(e_d(\tau)^T Q_e e_d(\tau) + u(\tau)^T R u(\tau))d\tau \tag{2.3}$$

Ở đây λ là hệ số suy giảm, $Q_e \in \mathbb{R}^{n\times n}$ và $R \in \mathbb{R}^{m\times m}$ là các ma trận đối xứng xác định dương. Để ý rằng hàm chi phí ở trên chứa chí phí cho sai lệch bám và toàn bộ chi phí của tín hiệu điều khiển tác động, để đảm bảo rằng hệ vừa có thể bám được quỹ đạo và chi phí cho điều khiển là tối ưu. Để điều khiển hệ (2.1) bám được quỹ đạo $x_d(t)$ một cách tối ưu, ta cần chọn được tín hiệu điều khiển $u(t)$ để làm cực tiểu hàm chi phí (2.3) đối với mọi tín hiệu $x(t)$. Ở đây, chúng ta giả sử rằng, quỹ đạo $x_d(t)$ thỏa mãn

$$\dot{x}_d(t) = r_d(x_d(t)), \quad r_d(0) = 0 \tag{2.4}$$

Ở đây $x_d(t)$ bị chặn và hàm $r_d(x_d(t)) \in \mathbb{R}^n$ là hàm Lipschitz liên tục. Trong các giải pháp đã tồn tại cho *Điều khiển bám tối ưu (OTCP)* trong [8], tín

hiệu điều khiển tối ưu bao gồm thành phần feedforward $u_d(t)$ và tín hiệu điều khiển phản hồi tối ưu $u_e(t)$. Giả sử quỹ đạo tối ưu $x_d(t)$ thỏa mãn

$$\dot{x}_d(t) = f(x_d(t)) + g(x_d(t))u_d(t) \tag{2.5}$$

Tín hiệu điều khiển feedforward $u_d(t)$ có thể tính được nếu như biết động học của hệ thống $f(x(t))$, $g(x(t))$ và tồn tại $g^{-1}(x)$:

$$u_d(t) = g^{-1}(x_d(t))(\dot{x}_d(t) - f(x_d(t))) \tag{2.6}$$

Tín hiệu phản hồi tối ưu $u_e(t)$ được thiết kế để tối ưu hóa hàm chi phí sau:

$$V_e(e_d(t)) = \int_t^\infty (e_d(\tau)^T Q_e e_d(\tau) + u_e(\tau)^T R u_e(\tau))d\tau \tag{2.7}$$

Và có thể đạt được bằng việc giải phương trình HJB đối với biểu thức (2.7):

$$u_e^* = -\frac{1}{2}R^{-1}g(x(t))^T \frac{\partial V_e(e_d(t))}{\partial e_d} \tag{2.8}$$

Chú ý 2.1.1. *Trong những phương pháp tiêu chuẩn để giải quyết bài toán OTCP, tín hiệu $u_d(t)$ yêu cầu phải biết mô hình động học của hệ thống, cả $f(x)$ và $g(x)$. Trong khi tín hiệu điều khiển $u_e(t)$ ít nhất cũng cần phải biết $g(x)$.*

2.2 Thuật toán Data-driven Policy Iteration (PI) cho OTCP hoàn toàn không biết về mô hình

Trong phần này, chúng em xin trình bày thuật toán Data-driven PI cho vấn đề OTCP đã được trình bày ở trên. Để làm được vậy ta cần phải đo được trạng thái x và tín hiệu điều khiển u, trong khi quỹ đạo mong muốn $x_d(t)$ ta đã biết chính xác.

2.2.1 Viết lại phương trình động học hệ thống

Trong bước này, phương trình động học hệ thống được viết lại ở dạng mở rộng như sau, hệ thống mở rộng này kết hợp sai số động học của hệ thống $e_d(t)$ và quỹ đạo đặt $x_d(t)$.

$$\dot{X}(t) = F(X(t)) + G(X(t))u(t) \tag{2.9}$$

Ở đây $X(t) = [e_d(t)^T \ x_d(t)^T]^T \in \mathcal{X} \subset \mathbb{R}^{2n}$ và

$$F(X(t)) = \begin{bmatrix} f(e_d(t) + x_d(t)) - r_d(x(t)) \\ r_d(x_d(t)) \end{bmatrix} \tag{2.10}$$

$$G(X(t)) = \begin{bmatrix} g(e_d(t) + x_d(t))) \\ 0 \end{bmatrix} \tag{2.11}$$

Hàm chi phí lúc này được viết lại thành

$$V(X(t)) = \int_t^\infty e^{-\lambda(\tau-t)}[X(\tau)^T Q X(\tau) + u(\tau)^T R u(\tau)]d\tau \tag{2.12}$$

$$Q = \begin{bmatrix} Q_e & 0 \\ 0 & 0 \end{bmatrix} \tag{2.13}$$

Hệ số suy giảm λ trong hàm chi phí là cần thiết theo [15] [14] [9], nếu quỹ đạo mong muốn không tiến về 0 khi thời gian tiến đến vô cùng, hàm chi phí với $\lambda = 0$ sẽ không bị chặn vì thành phần feedforward $u_d(t)$ phụ thuộc vào quỹ đạo trạng thái mong muốn. Để loại bỏ nhược điểm đó, thành phần $e^{-\lambda(\tau-t)}$ [18] được thêm vào để xử lí những trường hợp như trên, ví dụ như quỹ đạo mong muốn là một tín hiệu tuần hoàn bị chặn.

Định nghĩa 2.2.1. *Một tín hiệu $u(X)$ được định nghĩa là có thể chấp nhận được trong miền $\mathcal{X}$, kí hiệu $u(X) \in \mathcal{U}(\mathcal{X})$ nếu $u(X)$ liên tục trên miền $\mathcal{X}$, $u(0) = 0$, $u(X)$ ổn định hệ (2.9) trên $\mathcal{X}$ và $V(X)$ bị giới hạn $\forall X \in \mathcal{X}$*

Đạo hàm $V(X(t))$ trong (2.12) theo thời gian ta được:

$$\begin{aligned} \dot{V}(X) &= \lambda \int_t^\infty e^{-\lambda(\tau-t)}[X(\tau)^T Q X(\tau) + u(\tau)^T R u(\tau)]d\tau - X^T Q X - u^T R u \\ &= \lambda V(X) - X^T Q X - u^T R u \end{aligned}$$

Từ đó phương trình HJB có thể viết lại thành:

$$H(X, u^*, \nabla V^*(X)) = X^T Q X + u^{*T} R u^* - \lambda V^*(X) + \nabla V^*(X)^T (F(X) + G(X)) u^*) = 0 \tag{2.14}$$

Với $\nabla V^*(X) = \partial V^*(X)/\partial X$ và $V^*(X)$ là hàm chi phí tối ưu:

$$V^*(X(t)) = \min_{u \in \mathcal{U}(\Omega)} \int_t^\infty e^{-\lambda(\tau-t)}[X(\tau)^T Q X(\tau) + u(\tau)^T R u(\tau)]d\tau \tag{2.15}$$

Và tín hiệu điều khiển tối ưu:

$$u^*(X) = \underset{u \in \mathcal{U}(\Omega)}{argmin}\, H(X, u, \nabla V^*) = -\frac{1}{2} R^{-1} G(X)^T \nabla V^*(X) \tag{2.16}$$

Tín hiệu điều khiển tối ưu có thể giải được bằng giải thuật PI tiêu chuẩn như sau:

1. Policy Evaluation (Đánh giá luật điều khiển)

$$[\nabla V^{i+1}(X)]^T (F(X) + G(X)u^i) + X^T Q X + [u^i]^T R u^i - \lambda V^{i+1}(X) = 0 \tag{2.17}$$

2. Policy Improvement (Cải thiện luật điều khiển)

$$u^{i+1}(X) = -\frac{1}{2} R^{-1} G(X)^T \nabla V^{i+1}(X) \tag{2.18}$$

Định lí 2.2.2. *$u^i(X) \in \mathcal{U}(\mathcal{X})$ và $V^{i+1}(X) \in \mathcal{V}(\mathcal{X})$ thỏa mãn (2.17) với điều kiện chặn $V^{i+1}(0) = 0$. Khi đó, tín hiệu điều khiển (2.18) là có thể chấp nhận được trong $\mathcal{X}$, $\forall i \geq 1$. Hơn nữa, nếu $V^{i+1}(X)$ là hàm xác định dương duy nhất thỏa mãn (2.17) với điều kiện $V^{i+1}(0) = 0$ thì $V^*(X) \leq V^{i+1}(X) \leq V^i(X)$*

Chứng minh có thể tham khảo từ bài [12]. Như vậy theo định lí **2.2.2** ta chỉ cần một tín hiệu điều khiển ban đầu u^0 có thể chấp nhận được thì nghiệm của phương trình (2.17) và (2.18) sẽ hội tụ về nghiệm tối ưu: $\lim\limits_{i\to\infty} V^i = V^*$ và $\lim\limits_{i\to\infty} u^i = u^*$

2.2.2 Ứng dụng Data-driven PI để giải quyết bài toán trên

Từ chú ý **2.1.1** lời giải của bài toán OTCP cần biết cả $f(x(t))$ và $g(x(t))$, có rất nhiều phương pháp khác chỉ cần biết một phần nào đó của mô hình, chẳng hạn $g(x(t))$, có thể giải quyết được bài toán nhưng phương pháp dưới đây hoàn toàn không cần đến mô hình động học của hệ.
Từ phương trình: $\dot{X} = F(X) + G(X)u$ ta viết lại thành

$$\dot{X} = F(X) + G(X)u^i + G(X)[u - u^i] \tag{2.19}$$

Ở đây $u \in \mathbb{R}^m$. Đạo hàm $V^{i+1}(X)$ theo thời gian, kết hợp với (2.17) và (2.18) ta được:

$$\begin{aligned} \frac{dV^{i+1}(X)}{dt} &= [\nabla V^{i+1}]^T(F + Gu^i) + [\nabla V^{i+1}]^T G[u - u^i] \\ &= -X^T QX - [u^i]^T Ru^i + \lambda V^{i+1} + 2[u^{i+1}]^T R[u^i - u] \end{aligned} \tag{2.20}$$

Tích phân hai vế của (2.20) trong khoảng từ $[t, t + \delta t]$ ta được:

$$\begin{aligned} V^{i+1}(X(t+\delta t)) - V^{i+1}(X(t)) = &- \int_t^{t+\delta t} [X(\tau)^T QX(\tau) \\ &+ [u^i(X(\tau))]^T Ru^i(X(\tau))]d\tau \\ &+ \int_t^{t+\delta t} \lambda V^{i+1}(X(\tau))d\tau \\ &+ 2\int_t^{t+\delta t} [u^{i+1}(X(\tau))]^T R[u^i(X(\tau)) - u(\tau)]d\tau \end{aligned} \tag{2.21}$$

Nhìn vào biểu thức trên có thể thấy $f(x(t))$ và $g(x(t))$ không xuất hiện, do đó ta không cần biết đến mô hình động học của hệ thống. Tuy nhiên trạng thái hệ thống X và tín hiệu điều khiển u cần phải biết (cần đo hoặc tính toán được). Như vậy nghiệm của OTCP có thể đạt được từ phương trình trên.

Định lí 2.2.3. *Nếu $V^{i+1}(X) \in \mathcal{V}(\mathcal{X})$, ở đây $\mathcal{V}(\mathcal{X})$ là không gian hàm trên $\mathcal{X}$ khả vi liên tục, $V^{i+1}(X) \geq 0, V^{i+1}(0) = 0$ và $u^{i+1}(X) \in \mathcal{U}(\mathcal{X})$ thì nghiệm của (2.21) tương đương với nghiệm của (2.17) và (2.18)*

Chứng minh: **Phụ lục 1**

Theo định lí xấp xỉ Weierstrass [6] một hàm số liên tục có thể được biểu diễn bởi một tập vô hạn các hàm cơ sở độc lập tuyến tính. Từ đó, cấu trúc Actor-Critic Neural Network (A-C NN) được dùng để ước lượng các hàm $V(X)$ và $u(X)$ trên một tập đóng $\mathcal{X}$ được đưa ra như sau:

$$\begin{aligned} \hat{V}^i(X) &= [w_V^i]^T \varphi(X) \\ \hat{u}^i(X) &= [w_u^i]^T \psi(X) \end{aligned} \tag{2.22}$$

Ở đây $\varphi(X) \in \mathbb{R}^{l_\varphi}$ và $\psi(X) \in \mathbb{R}^{l_\psi}$ là 2 vecto hàm cơ sở dùng để xấp xỉ, tương ứng các trọng số $w_V \in \mathbb{R}^{l_\varphi}$ và $w_u \in \mathbb{R}^{l_\psi \times m}$ (lưu ý $u \in \mathbb{R}^m$).
Ta định nghĩa sai số ước lượng $\epsilon_u^i(X) = \hat{u}^i(X) - u^i(X)$. Hệ kín với luật điều khiển $\hat{u}^i(X)$ được viết lại như sau:

$$\dot{X} = F(X) + G(X)\hat{u}^i(X) \tag{2.23}$$

Định lí 2.2.4. *Xét hệ thống (2.23) với luật điều khiển xấp xỉ $\hat{u}^i(X)$, với sai số ước lượng $\epsilon_u^i(X)$ và hệ số suy giảm λ tiến về 0, tín hiệu điều khiển $\hat{u}^i(X)$ sẽ đảm bảo đưa sai lệch bám $e_d(t)$ ổn định tiệm cận.*

Chứng minh: **Phụ lục 2**

Bây giờ, đặt $t_{k-1} = t$ và $t_k = t + \delta t$. Ta xem xét phương trình (2.21) để tiện cho việc giải phương trình Off-Policy ta có thể viết lại thành:

$$\begin{aligned} \sigma(X(t_{k-1}), u, X(t_k)) =& [\varphi(X(t_{k-1})) - \varphi(X(t_k))]^T w_V^{i+1} + \int_{t_{k-1}}^{t_k} \lambda [\varphi(X(\tau))]^T d\tau w_V^{i+1} \\ &+ 2\sum_{p=1}^{m} r_p \int_{t_{k-1}}^{t_k} [u_p^i - u_p(\tau)]^T \theta^T(X(\tau)) w_{u,p}^{i+1} d\tau \\ &- \int_{t_{k-1}}^{t_k} [X(\tau)^T Q X(\tau) d\tau - \int_{t_{k-1}}^{t_k} [u^i(\tau)]^T R u^i(\tau) d\tau \end{aligned} \tag{2.24}$$

$\sigma(X(t_{k-1}), u, X(t_k))$ là sai số giữa 2 vế của phương trình (2.21)

$$\begin{aligned} \xi_{\Delta\varphi k} &= \varphi(X(t_{k-1})) - \varphi(X(t_k)) \\ \xi_{Qk} &= \int_{t_{k-1}}^{t_k} [X(\tau)^T Q X(\tau) d\tau \\ \xi_{\lambda k} &= \int_{t_{k-1}}^{t_k} \lambda \varphi(X(\tau))) d\tau \\ \xi_{u\theta k,p} &= r_p \int_{t_{k-1}}^{t_k} [u_p^i - u_p(\tau)]^T \theta^T(X(\tau)) d\tau \\ \xi_{uk} &= \int_{t_{k-1}}^{t_k} [u^i(\tau)]^T R u^i(\tau) d\tau \end{aligned} \tag{2.25}$$

Từ phương trình (2.24) và (2.25) suy ra:

$$\begin{aligned}\sigma(X(t_{k-1}),u,X(t_k)) &= \xi_{\Delta\varphi k}^T w_V^{i+1} + \xi_{\lambda k} w_V^{i+1} \\ &= 2\sum_{p=1}^{m} \xi_{u\theta k,p} w_{u,p}^{i+1} - \xi_{Qk} - \xi_{uk} \\ &= \Lambda_k w_{Vu}^{i+1} - \Upsilon_k^i\end{aligned} \tag{2.26}$$

Ở đây:
$\Lambda_k = [(\xi_{\Delta\varphi k}^T + \xi_{\lambda k}^T), \xi_{u\theta k,p,1}, \xi_{u\theta k,p,m}]$
$\Upsilon_k^i = \xi_{Qk} + \xi_{uk}$
$w_{Vu}^{i+1} = [(w_V^{i+1})^T, (w_{u,1}^{i+1})^T ... (w_{u,m}^{i+1})^T]^T$

w_{Vu}^{i+1} có thể đạt được bằng cách cực tiểu hóa $\sigma(X(t_{k-1}),u,X(t_k))$:

$$w_{Vu}^{i+1} = [\sum_{k=1}^{M} (\Lambda_k w_{Vu}^{i+1})^T (\Lambda_k w_{Vu}^{i+1})]^{-1} \sum_{k=1}^{M} (\Lambda_k w_{Vu}^{i+1})^T \Upsilon_k^i \tag{2.27}$$

M là số khoảng lấy mẫu.
Lưu ý rằng số lượng dữ liệu thu thập được là phải đủ lớn và nhiễu u_e thêm vào là rất quan trọng để thỏa mãn điều kiện PE. Thực tế ra ngoài việc thêm nhiễu để thỏa mãn điều kiện PE, còn có một cách khác là dùng nhiều tín hiệu điều khiển u khác nhau để lấy dữ liệu.
Ngoài ra cần lưu ý rằng điều kiện chặn trên của hệ số suy giảm λ, $\lambda \leq \bar{\lambda} = 2\|(B_1 R^{-1} B_1^T Q_e)^{1/2}\|$ để sai số cục bộ ổn định tiệm cận, cụ thể xem trong **Phụ lục 3**
Từ đó giải phương trình (2.21) bằng phương pháp Data-driven với M khoảng thời gian thu thập dữ liệu ta tìm được nghiệm tối ưu. Thuật toán tóm tắt lại như sau:

Thuật toán 1. *Thuật toán Data-driven PI:*

---

1. *Khởi tạo*
   Bắt đầu với tín hiệu điều khiển ổn định u^0 và phần nhiễu u_e thêm vào để đảm bảo điều kiện PE. Thu thập dữ liệu và xác định ngưỡng ϵ

2. *Policy Evaluation*
   Với tín hiệu $u^i(X)$ giải được từ vòng lặp trước, giải $V^{i+1}(X)$ và $u^{i+1}(X)$

từ phương trình:

$$\begin{aligned} V^{i+1}(X(t+\delta t)) - V^{i+1}(X(t)) = &- \int_t^{t+\delta t} [X(\tau)^T Q X(\tau) \\ &+ [u^i(X(\tau))]^T R u^i(X(\tau))] d\tau \\ &+ \int_t^{t+\delta t} \lambda V^{i+1}(X(\tau)) d\tau \\ &+ 2\int_t^{t+\delta t} [u^{i+1}(X(\tau))]^T R[u^i(X(\tau)) - u(\tau)] d\tau \end{aligned} \tag{2.28}$$

3. *Policy Improvement*
   Tiếp tục lặp cho đến khi $\|u^{i+1} - u^i\| < \epsilon$
   Cập nhật $u^i = u^{i+1}$

2.3 Ví dụ

2.3.1 Hệ tuyến tính

Xét hệ thống lò xo có ma sát nhớt được mô tả bằng phương trình sau:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{k}{m}x_1 - \frac{c}{m}x_2 + \frac{1}{m}u(t) \end{aligned} \tag{2.29}$$

Ở đây x_1 và x_2 là vị trí và vận tốc tương ứng. $m = 1kg$, $c = 0.5Ns/m$ và $k = 5N/m$ Quỹ đạo điều khiển cần bám là:

$$\begin{aligned} x_{d1}(t) &= 0.5sin(\sqrt{5}t) \\ x_{d2}(t) &= 0.5\sqrt{5}cos(\sqrt{5}t) \end{aligned} \tag{2.30}$$

Hay có thể viết lại thành:

$$\dot{x}_d = \begin{bmatrix} 0 & 1 \\ -5 & 0 \end{bmatrix} x_d \tag{2.31}$$

Hệ thống mở rộng được viết lại như sau:

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -5 & -0.5 & 0 & -0.5 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -5 & 0 \end{bmatrix} X + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u = AX + Bu \tag{2.32}$$

Thông số của hàm chi phí có thể chọn như sau: $R = 1$, $Q_e = 100I_2$ và $\lambda = 0.01$. Hàm kích hoạt cho vector hàm cơ sở có thể là hàm Gaussian, Sigmoid hoặc

đa thức. Trong ví dụ này ta chọn đa thức làm hàm kích hoạt.
NN của Critic được chọn như sau:

$$\begin{aligned}\varphi(X) &= [X_1^2, X_1X_2, X_1X_3, X_1X_4, X_2^2, X_2X_3, X_2X_4, X_3^2, X_3X_4, X_4^2]^T \\ w_V &= [w_{V1}..w_{V10}]^T\end{aligned} \tag{2.33}$$

NN của Actor được chọn như sau:

$$\begin{aligned}\psi(X) = [&X_1, X_2, X_3, X_4, X_1^2, X_1X_2, X_1X_3, X_1X_4, X_2^2, X_2X_3, \\ &X_2X_4, X_3^2, X_3X_4, X_4^2, X_1^3, X_2^3, X_3^3, X_4^3]^T \\ w_u = [&w_{u1}..w_{u18}]^T\end{aligned} \tag{2.34}$$

Trọng số khởi tạo cho Actor được chọn:

$$\begin{aligned}w_u^{(0)} =[&0,0,0,0.5,0,0,0,0,0,0,0,\\ &0,0,0,0,0,0,0]^T\end{aligned} \tag{2.35}$$

Kết quả sau 24 vòng lặp trọng số hội tụ:

$$\begin{aligned}w_V^{(24)} =[&115.118, 12.070, 4.477, 0.132, 10.211, 0.143,\\ &0.956, 61.910, 0.126, 12.382]^T\end{aligned} \tag{2.36}$$

$$\begin{aligned}w_u^{(24)} =[&6.135, 10.101, 0.0105, 0.477, 0, 0, 0, 0, 0, 0, 0,\\ &0, 0, 0, 0.00022, 0.00057, 0, 0]^T\end{aligned} \tag{2.37}$$

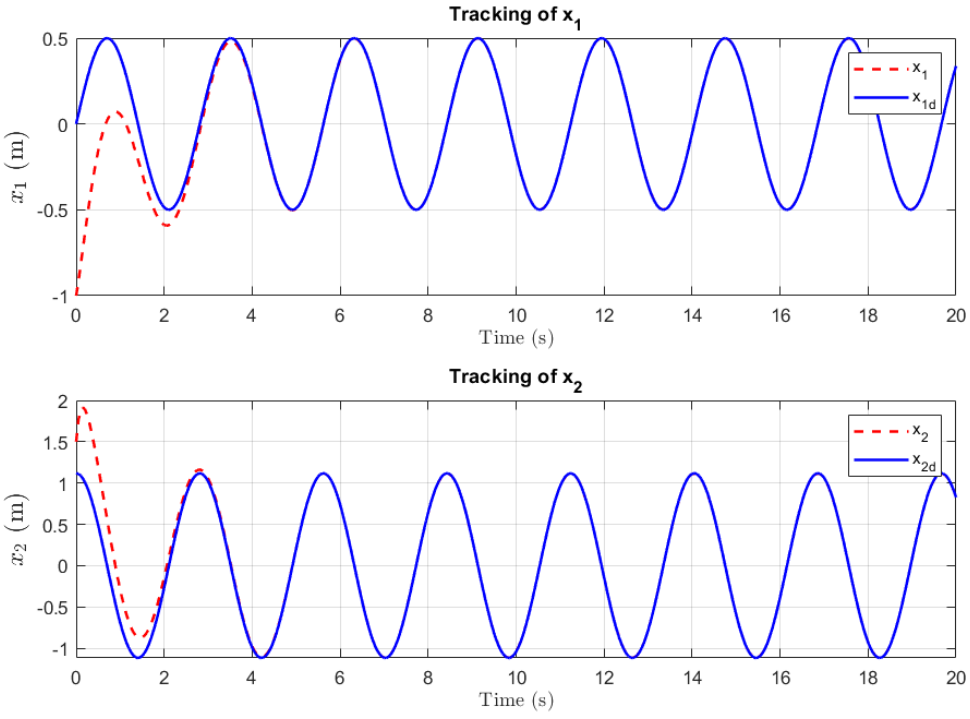


Figure 2.1: Quỹ đạo bám đối với tuyến tính

Hình (2.1) biểu diễn quỹ đạo x_1 và x_2 thực tế bám so với quỹ đạo đặt với điều kiện ban đầu $x_0 = [-1, 1.5]^T$, sử dụng bộ điều khiển tối ưu từ trọng số hội tụ. Do hệ là tuyến tính nên nghiệm bài toán, nếu biết mô hình động học của hệ, có thể thu được từ việc giải phương trình ARE dưới đây:

$$A^T P + PA - \lambda P + PBR^{-1}B^T P + Q = 0 \tag{2.38}$$

$$P^* = \begin{bmatrix} 115.5458 & 6.1266 & -2.2452 & -0.0249 \\ 6.1266 & 10.1005 & 0.0132 & -0.4771 \\ -2.2452 & 0.0132 & 62.3598 & -0.0622 \\ -0.0249 & -0.4771 & -0.0622 & 12.4730 \end{bmatrix}$$

Từ ma trận $P*$, ta thu được w_V^*:

$$\begin{aligned} w_V^* =& [115.5458, 12.2532, 4.4904, 0.0498, 10.1005, 0.0264, \\ & 0.9542, 62.3598, 0.1244, 12.4730]^T \end{aligned}$$

So sánh w_V^* và $w_V^{(24)}$, ta thấy kết quả ra khá là tương đồng với nhau, chứng tỏ tính đúng đắn của thuật toán

Để kiểm chứng ảnh hưởng của λ đến chất lượng điều khiển, giữ nguyên các tham số ban đầu, thay đổi λ lần lượt từ 0.01, 0.1 và 0.5 ta thu được đồ thị trên hình (2.2)

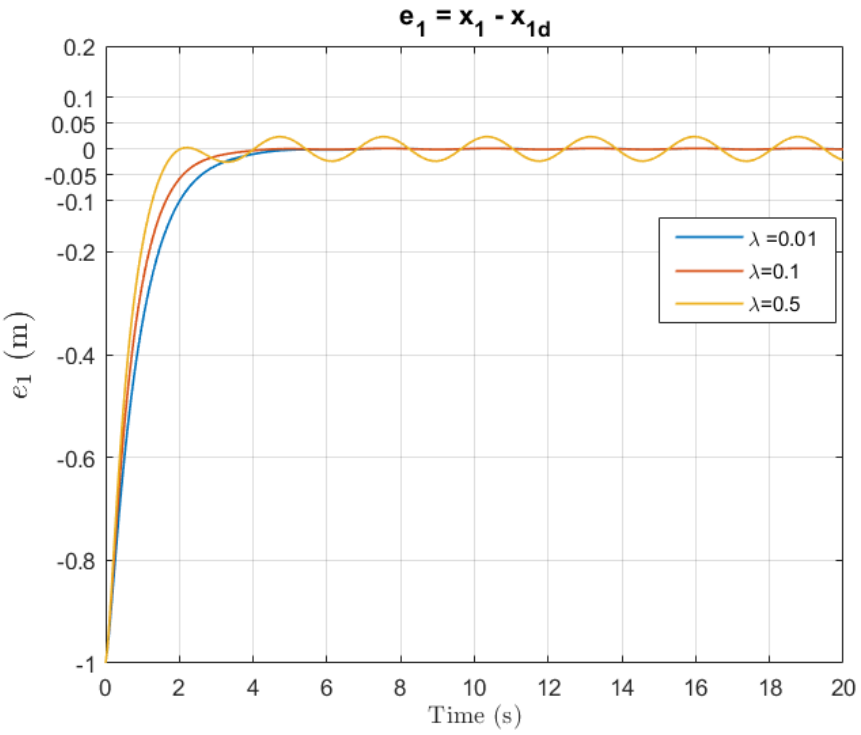


Figure 2.2: e_1 thay đổi theo λ

Nhận xét 2.3.1. *Giá trị chặn của sai lệch e_1 là 0.0001, 0.00027 và 0.02 tương ứng với giá trị λ bằng 0.01, 0.1 và 0.5. Có thể thấy, giá trị sai lệch có thể được điều chỉnh nhỏ theo mong muốn bằng việc chọn hệ số suy giảm λ đủ nhỏ.*

2.3.2 Hệ phi tuyến

Xét hệ phi tuyến affine sau:

$$\dot{x} = \begin{bmatrix} x_2 \\ -0.5(x_1 + x_2) + 0.5x_1^2 x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{2.39}$$

Quỹ đạo đặt $x_d = [0.5sin(t), 0.5cos(t)]^T$ do đó:

$$\dot{x}_d = \begin{bmatrix} x_{2d} \\ -x_{1d} \end{bmatrix}$$

Điều kiện ban đầu $x(0) = [0.2\, 0.1]^T$ và $x_d(0) = [0\, 0.5]^T$. Thông số hàm chi phí được chọn với $Q_e = 100I_2$ và $R = 1$, $\lambda = 0.01$. Critic NN được chọn là:

$$\begin{aligned} \varphi(X) = [&X_1^2, X_1X_2, X_1X_3, X_1X_4, X_2^2, X_2X_3, X_2X_4, X_3^2, \\ &X_3X_4, X_4^2, X_1^3, X_2^3, X_3^3, X_4^3, X_1^4, X_2^4, X_3^4, X_4^4]^T \\ w_V = [&w_{V1}...w_{V18}]^T \end{aligned} \tag{2.40}$$

Actor NN được chọn là:

$$\begin{aligned} \psi(X) = [&X_1, X_2, X_3, X_4, X_1^2, X_1X_2, X_1X_3, X_1X_4, X_2^2, X_2X_3, \\ &X_2X_4, X_3^2, X_3X_4, X_4^2, X_1^3, X_1X_2X_3, X_1X_2X_4, X_1^2X_2, \\ &X_1^2X_3, X_1^2X_4, X_2^3, X_2^2X_1, X_2^2X_3, X_2^2X_4, X_2X_3X_4, \\ &X_3^3, X_3^2X_1, X_3^2X_2, X_3^2X_4, X_4^3, X_4^2X_1, X_4^2X_2, X_4^2X_3]^T \\ w_u = [&w_{u1}...w_{u34}]^T \end{aligned} \tag{2.41}$$

Trọng số khởi tạo của Actor được chọn như sau:

$$\begin{aligned} w_u^{(0)} = [&0, 0, -0.5, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0.5, \\ &0, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0.5, 0.5, 0, 0, 0, 0]^T \end{aligned} \tag{2.42}$$

Sau 60 vòng lặp, trọng số hội tụ cho kết quả như sau:

$$\begin{aligned} w_V^{(60)} = [&102.7629, 18.9965, 1.1704, 0.3302, 10.6051, 1.0414, -0.7600, 22.6232, -0.0341, \\ &22.4059, 8.6455, 0.9452, 0.0021, 0.0006, 781.7642, 1.2293, -0.3745, 0]^T \end{aligned}$$

$$\begin{aligned} w_u^{(60)} = [&-7.7546, -10.9013, -0.5733, 0.4068, 33.2184, 6.4724, -2.2191, 4.4119, \\ &-0.5362, -0.1452, 0.5718, 0.0333, -0.1610, 0.1357, -432.6140, -0.9517, \\ &51.7376, -0.6381, -143.7584, -30.0475, -121.1779, -7.1010, -41.0103, 10.2111, \\ &-9.9278, -1.2006, 0, 0, 0, 0.4219, 0, -10.4996, \\ &-0.7099, 0.3058]^T \end{aligned}$$

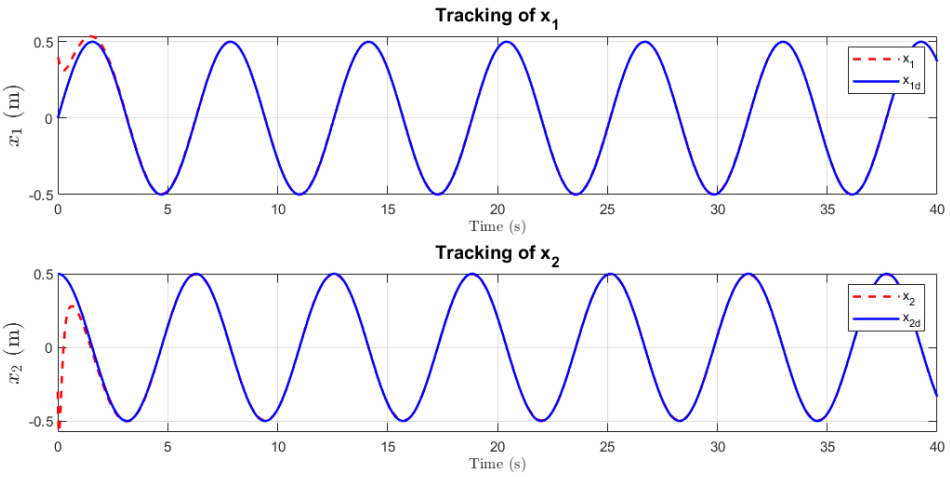


Figure 2.3: Quỹ đạo bám đối với hệ phi tuyến

Sử dụng bộ điều khiển tối ưu với bộ trọng số hội tụ trên, ta thu được kết quả như hình 2.3. Ta có thể thấy chất lượng điều khiển bám khá tốt, sai lệch bám hội tụ về 0 sau khoảng 4 (s).

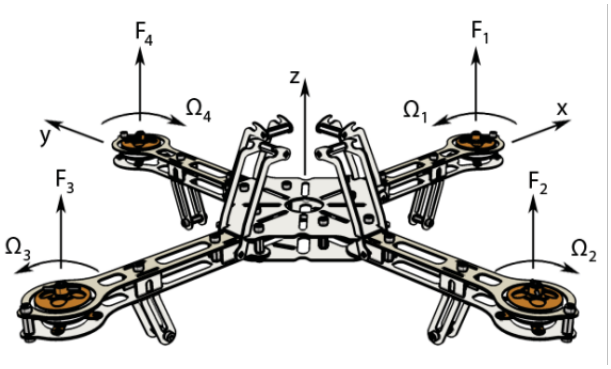
Vậy qua 2 ví dụ áp dụng thuật toán Data-driven PI cho 2 hệ tuyến tính và phi tuyến khá đơn giản, ta có thể thấy tính đúng đắn của thuật toán. Để chứng tỏ khả năng áp dụng cho các hệ phi tuyến phức tạp, chương tiếp theo nghiên cứu áp dụng thuật toán để tìm bộ điều khiển tối ưu cho đối tượng máy bay không người lái "Quadrotor".

Chapter 3

Ứng dụng thuật toán Data-Driven PI cho quadrotor

3.1 Giới thiệu về Quadrotors

Trong những năm gần đây, máy bay không người lái nhận được sự quan tâm lớn trong cộng đồng nghiên cứu nhờ tiềm năng ứng dụng to lớn của nó trong rất nhiều công việc, lĩnh vực mà ở đó sự hiện diện của con người là khó có thể đáp ứng: khảo sát thảm họa, phát hiện cháy rừng, ứng dụng trong nông nghiệp,...



Một trong những loại máy bay không người lái điển hình, quadrotor đã và đang trở nên ngày càng phổ biến nhờ khả năng cất cánh và tiếp đất theo phương thẳng đứng, khả năng ổn định vị trí và quỹ đạo linh hoạt.
Theo đó, một trong những bài toán phổ biến và quan trọng được đưa ra đó là bài toán điều khiển bám cho quadrotor. Đây là một bài toán không hề đơn giản do quadrotor có 6 bậc tự do với 4 đầu vào điều khiển, song, mô hình quadrotor còn là mô hình đối tượng có sự xen kênh mạnh. Hơn thế nữa, trong các bài toán và ứng dụng thực tế, việc biết chính xác mô hình đối tượng là khó có thể đáp ứng, kèm theo đó là bất định mô hình thường xuyên gặp phải do yêu cầu nhiệm cụ của quadrotor thường xuyên phải gắn thêm một vật nặng phức tạp, ví dụ: camera, vật phẩm cần chuyển trong ứng dụng giao hàng, ... Vậy nên yêu cầu giải quyết bất định mô hình là một yêu cầu quan trọng khi thiết kế bộ điều khiển cho quadrotor. Trong nhiều nghiên cứu gần đây, để đạt được chất lượng điều khiển tốt, nhiều cấu trúc điều khiển được đưa ra có thể kể đến: bộ điều khiển PID [7],[19], bộ điều khiển LQR [11], [20],

bộ điều khiển backstepping [4], [5], [13], bộ điều khiển sliding mode [16]-[21]. Trong [20], bộ điều khiển LQR được thiết kế để ổn định mô hình quadrotor trong giải làm việc nhỏ. Trong [16], cấu trúc điều khiển fast nonlinear sliding mode được nghiên cứu với mục đích đưa biến sai lệch về vị trí cân bằng trong khoảng thời gian hữu hạn. Bên cạnh đó, nhiều phương pháp dựa trên bộ quan sát cũng được sử dụng để bù sai lệch động học cho mô hình quadrotor lý tưởng, sau đó một bộ điều khiển phản hồi phi tuyến được thiết kế để giải quyết bài toán điều khiển bám.
Ở phần này, chúng em tiến hành áp dụng thuật toán Data-driven PI đã trình bày ở các phần trước cho đối tượng là quadrotor.

3.2 Mô hình động học của Quadrotor

Quadrotor có thể miêu tả đơn giản bằng hình dưới đây:

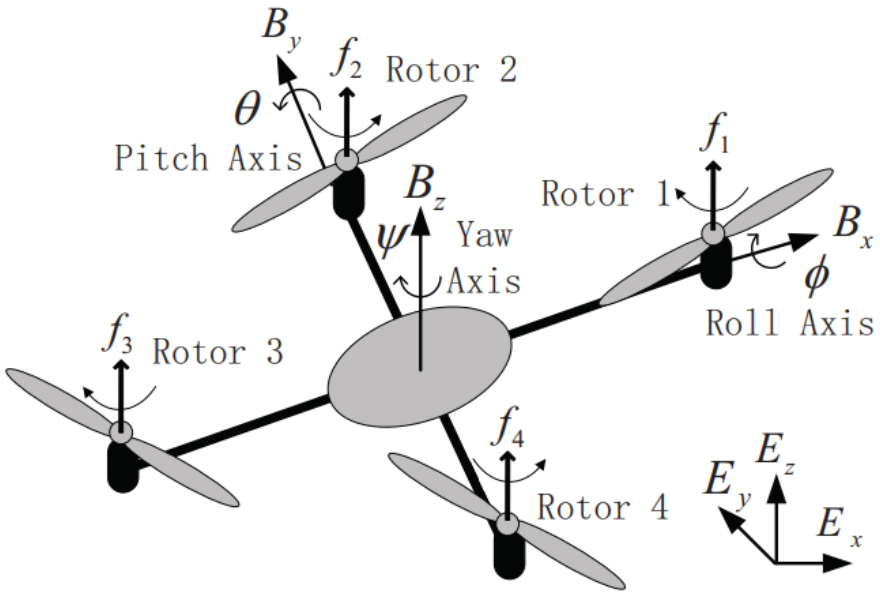


Figure 3.1: Nguyên lý động học của quadrotor

Khung tọa độ gắn với Trái Đất(Earth-fixed inertial frame):là khung tọa độ cố định gắn với Trái Đất, kí hiệu: $\alpha = E_x E_y E_z$ (như trong hình 4.1)
Khung tọa độ gắn cứng với Quadrotor(Body-fixed frame): là khung tọa độ có gốc gắn với khối tâm quadrotor, miêu tả hướng quay của quadrotor so với khung tọa độ gắn với Trái Đất, kí hiệu: $\beta = B_x B_y B_z$ (như trong hình 4.1)
Kí hiệu:
Vị trí của khối tâm quadrotor hay gốc tọa độ của khung β so với khung tọa độ α: $p = [p_x, p_y, p_z]^T \in \mathbb{R}^3$. Các góc Euler so với khung tọa độ α là $\Theta = [\phi, \theta, \psi] \in \mathbb{R}^3$.

Vận tốc góc tức thời của quadrotor trong hệ quy chiếu β là $\omega = [p, q, r]^T$, vận tốc góc này khác với đạo hàm theo thời gian của các góc Euler $\dot{\Theta} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]$ do các góc Euler là các góc quay có thứ tự, công thức liên hệ giữa chúng là:

$$\omega = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \dot{\Theta} \tag{3.1}$$

Góc Euler ZYX được dùng để miêu tả sự quay của quadrotor trong hệ tọa độ gắn với Trái Đất, $R = R_{\beta\to\alpha} = R_{Z,\psi}R_{Y,\theta}R_{X,\phi}$ là ma trận chuyển đơn vị từ tọa độ β sang α:

$$R_{Z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \; R_{Y,\theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \; R_{X,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}$$

$$R = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \tag{3.2}$$

Phương trình động lực học:

$$m\ddot{p} = T_p R e_{3,3} - mge_{3,3} \tag{3.3}$$

$$J\dot{\omega} = \tau - \omega \times (J\omega) \tag{3.4}$$

Ở đây $T_p \in \mathbb{R}$ là tổng lực nâng của các cánh quạt $T_p = T_1 + T_2 + T_3 + T_4$ trong hệ tọa độ α và $\tau = [\tau_\phi, \tau_\theta, \tau_\psi]^T \in \mathbb{R}^3$ là momen tác động vào quadrotor trong hệ toạ độ β quay quanh các trục x, y, z, $T_p = k_w u_z$ và $\tau = [l_\tau k_w u_\phi,\ l_\tau k_w u_\theta,\ k_t u_\psi]^T$. m là khối lượng của quadrotor, g là gia tốc trọng trường, $J = diag(J_x, J_y, J_z)$ với các $J_x,\ J_y,\ J_z$ lần lượt các momen quán tính quanh các trục $x,\ y,\ z$ tương ứng. l_τ là chiều dài từ tâm của quadrotor tới mỗi rotor. $k_w(Ns^2)$ và $k_t(Ns^2/m)$ là các hằng số khí động tỉ lệ. Có thể thấy từ mô hình động lực học của quadrotor, có 6 DOF với 4 đầu vào $(u_z, u_\phi, u_\theta, u_\psi)$ và là hệ phi tuyến siêu rằng buộc (super-coupling). Các tín hiệu đầu vào $u_z, u_\phi, u_\theta, u_\psi$ phụ thuộc vào vận tốc quay của các cánh quạt như sau:

$$\begin{aligned} u_z &= \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 \\ u_\phi &= \omega_2^2 - \omega_4^2 \\ u_\theta &= \omega_1^2 - \omega_3^2 \\ u_\psi &= \omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2 \end{aligned} \tag{3.5}$$

Với $\omega_j (j = 1, 2, 3, 4)$ là các vận tốc quay của các cánh quạt j tương ứng.
Ngoài ra dựa vào mối liên hệ trong công thức (3.1) và (3.4) ta thu được hệ

phương trình động lực học khác:

$$\begin{aligned} m\ddot{p} &= T_p R e_{3,3} - mge_{3,3} \\ J\ddot{\Theta} &= \tau - C(\Theta, \dot{\Theta})\dot{\Theta} \end{aligned} \tag{3.6}$$

Ở đây:

$$C(\Theta, \dot{\Theta}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

$$\begin{aligned}
c_{11} &= 0 \\
c_{12} &= (J_y - J_z)(\dot{\theta}c\phi s\phi + \dot{\psi}s^2\phi c\theta)/J_x + ((J_z - J_y)\dot{\psi}c^2\phi c\theta - J_x\dot{\psi}c\theta)/J_x, \\
c_{13} &= (J_z - J_y)\dot{\psi}c\phi s\phi c^2\theta/J_x, \\
c_{21} &= (J_z - J_y)(\dot{\theta}c\phi s\phi + \dot{\psi}s^2\phi c\theta)/J_y + ((J_y - J_z)\dot{\psi}c^2\phi c\theta + J_x\dot{\psi}c\theta)/J_y, \\
c_{22} &= (J_z - J_y)\dot{\phi}c\phi s\phi/J_y, \\
c_{23} &= (-J_x\dot{\psi}s\theta c\theta + J_z\dot{\psi}c^2\phi c\theta s\theta)/J_y + \dot{\psi}s^2\phi c\theta s\theta, \\
c_{31} &= ((J_y - J_z)\dot{\psi}c^2\theta s\phi c\phi - J_x\dot{\theta}c\theta)/J_z, \\
c_{32} &= (J_z - J_y)(\dot{\theta}c\phi s\phi s\theta + \dot{\phi}s^2\phi c\theta)/J_z + ((J_y - J_z)\dot{\phi}c^2\phi c\theta + J_x\dot{\psi}s\theta c\theta)/J_z \\
&\quad - J_y\dot{\psi}s^2\phi s\theta c\theta/J_z - \dot{\psi}c^2\phi c\theta s\theta, \\
c_{33} &= (-J_z\dot{\theta}c^2\phi s\theta c\theta - J_y\dot{\theta}s^2\phi c\theta s\theta)/J_z + ((J_y - J_z)\dot{\phi}c\phi s\phi c^2\theta + J_x\dot{\theta}s\theta c\theta)/J_z
\end{aligned}$$

Công thức (3.6) được chúng em sử dụng chính trong đồ án này

3.3 Phương pháp điều khiển Quadrotor nói chung

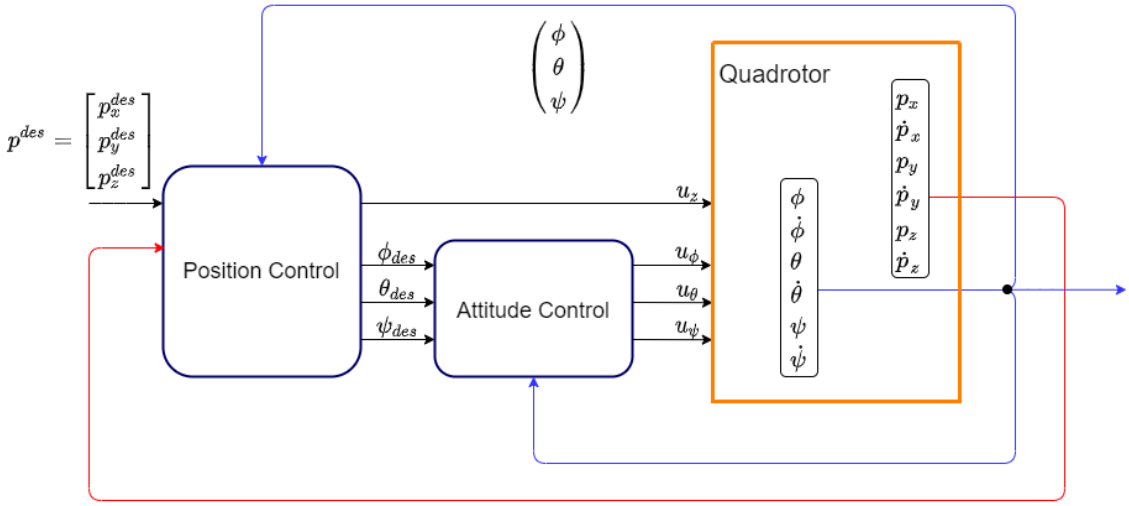


Figure 3.2: Nguyên lý điều khiển chung của quadrotor

Hình trên là nguyên tắc điều khiển một quadrotor nói chung, do quadrotor là một hệ phi tuyến, nhiều thành phần đan xen vào nhau nhưng nhìn chung đều có thể tách biệt ra làm 2 bộ điều khiển chính:

1. *Bộ điều khiển vị trí (Position Controller)*: Nhiệm vụ sinh ra quỹ đạo đặt góc hướng cho bộ điều khiển góc hướng (Attitude Controller).

2. *Bộ điều khiển góc hướng (Attitude Controller)*: Nhiệm vụ bám được quỹ đạo góc hướng đặt nhằm bắm được quỹ đạo về vị trí trong không gian.

Có thể hiểu đơn giả rằng mục đích điều khiển là điều chỉnh vận tốc quay của 4 motor sao cho quadrotor có thể nghiêng đúng hướng, kết hợp với lực nâng sẽ di chuyển bám quỹ đạo mong muốn.
Các tín hiệu đo phản hồi về là rất quan trọng, đối với mô hình quadrotor người ta hoàn toàn có thể do được toàn bộ các đại lượng vật lý của nó bao gồm:

1. *Vị trí, vận tốc và gia tốc*: Trong việc nghiên cứu ở các phòng thí nghiệm (Indoor), hệ thống Vicon System [1] được sử dụng để đo được các đại lượng trên một cách chính xác.

2. *Góc Euler, Vận tốc góc và gia tốc góc*: Sử dụng các loại cảm biến IMU (Internal Measuring Unit) quen thuộc cũng giúp đo được các đại lượng trên

3. Ngoài ra đối với trường hợp là các máy bay ngoài trời (outdoor) thường gắn thêm cảm biến áp suất (barometer), đo vị trí bằng GPS (Global Positioning System) và lưu lại các thông số qua hộp đen (blackbox).

Ở đồ án này, chúng em nghiên cứu mô phỏng quadrotor ở mức độ phòng thí nghiệm (Indoor) do đó hầu hết các đại lượng phản hồi đều đo được.

3.4 Thuật toán Data-driven PI cho quadrotor bám quỹ đạo hoàn toàn không biết về mô hình

3.4.1 Điều khiển vị trí (Position Control) với Data-driven PI

Phương trình (3.3) có thể viết lại thành:

$$\begin{aligned} \ddot{p} &= m^{-1} k_w u_z Re_{3,3} - g e_{3,3} \\ &= m^{-1} k_w u_p \end{aligned} \tag{3.7}$$

Ở đây: $u_p = u_z Re_{3,3} - \frac{m}{k_w} g e_{3,3} \in \mathbb{R}^3$, thành phần off-set $u_b = \frac{m}{k_w} g e_{3,3}$ trong thực tế có thể hoàn toàn có thể xác định được mà không cần biết m, g, k_w (xem trong **Phụ Lục 4**)
Đặt $x_p = [p_x, \dot{p}_x, p_y, \dot{p}_y, p_z, \dot{p}_z]^T \in \mathbb{R}$, phương trình (3.7) có thể viết lại như sau:

$$\dot{x}_p = A_p x_p + B_p u_p \tag{3.8}$$

Với $A_p = diag(a_p, a_p, a_p) \in \mathbb{R}^{6\times6}$, $a_p = [0_{2\times1}\ e_{2,1}]$ và $B_p = m^{-1}k_w[e_{6,2}, e_{6,4}, e_{6,6}]$
Giả sử quỹ đạo đặt mong muốn là $\dot{x}_{pd}$ có phương trình $\dot{x}_{pd} = A_{pd}x_{pd}$, và sai lệch $e_p = x_p - x_{pd}$, phương trình mở rộng có thể viết thành:

$$\dot{X}_p = \begin{bmatrix} \dot{e}_p \\ \dot{x}_{pd} \end{bmatrix} = \begin{bmatrix} A_p & A_p - A_{pd} \\ 0_{6\times6} & A_{pd} \end{bmatrix} X_p + \begin{bmatrix} B_p \\ 0_{6\times3} \end{bmatrix} u_p \tag{3.9}$$

Ma trận $Q_p = \begin{bmatrix} Q_{ep} & 0_{6\times6} \\ 0_{6\times6} & 0_{6\times6} \end{bmatrix}$, trong đó Q_{ep} là ma trận đối xứng xác định dương, $R_p \in \mathbb{R}^{3\times3}$ cũng là ma trận đối xứng xác định dương.
Với hàm chi phí được chọn là:

$$V_p(X_p(t)) = \int_t^\infty e^{-\lambda(\tau-t)}(X_p(\tau)^T Q_p X_p(\tau) + u_p(\tau)^T R_p u_p(\tau))d\tau \tag{3.10}$$

Như trong chương 2 đã trình bày, tương tự ta có thuật toán Data-driven PI cho điều khiển vị trí như sau:

Thuật toán 2. *Data-driven PI cho điều khiển vị trí:*

---

1. *Khởi tạo*
   Bắt đầu với tín hiệu điều khiển ổn định u_p^0 và phần nhiễu u_{pe} thêm vào để đảm bảo điều kiện PE. Thu thập dữ liệu và xác định ngưỡng ϵ_p

2. *Policy Evaluation*
   Với tín hiệu $u_p^i(X_p)$ giải được từ vòng lặp trước, giải $V_p^{i+1}(X_p)$ và $u_p^{i+1}(X_p)$ từ phương trình:

$$\begin{aligned} V_p^{i+1}(X_p(t+\delta t)) - V_p^{i+1}(X_p(t)) = &- \int_t^{t+\delta t} [X_p(\tau)^T Q_p X_p(\tau) \\ &+ [u_p^i(X_p(\tau))]^T R_p u_p^i(X_p(\tau))]d\tau \\ &+ \int_t^{t+\delta t} \lambda V_p^{i+1}(X_p(\tau))d\tau \\ &+ 2\int_t^{t+\delta t} [u_p^{i+1}(X_p(\tau))]^T R_p u_p^i(X_p(\tau))d\tau \\ &- 2\int_t^{t+\delta t} [u_p^{i+1}(X_p(\tau))]^T R_p[u_p^0(\tau) + u_{pe}]d\tau \end{aligned} \tag{3.11}$$

3. *Policy Improvement*
   Tiếp tục lặp cho đến khi $\|u_p^{i+1} - u_p^i\| < \epsilon_p$
   Cập nhật $u_p^i = u_p^{i+1}$

Để xấp xỉ V_p^i và u_p^i, Critic-Actor NN được xây dựng như sau:

$$V_p^i(X_p) = w_{Vp}^T \varphi_p(X_p) \tag{3.12}$$

$$u_p^i(X_p) = w_{up}^T \psi_p(X_p) \tag{3.13}$$

Ở đây $\varphi_p(X_p) \in \mathbb{R}^{l_1}$ và $\psi_p(X_p) \in \mathbb{R}^{l_2}$ là 2 vecto hàm cơ sở (l_1 và l_2 lần lượt là số neurons của $\varphi_p(X_p)$ và $\psi_p(X_p)$).$w_{Vp} \in \mathbb{R}^{l_1 \times 1}$ và $w_{up} \in \mathbb{R}^{l_2 \times 3}$ là 2 vecto trọng số tương ứng. Từ đó có thể áp dụng phương pháp Least-Square để giải phương trình (3.11).
Sau khi tìm được tín hiệu $u_p = [u_{px}, u_{py}, u_{pz}]^T$ tối ưu, từ (3.7) ta có thể giải được tín hiệu điều khiển u_z, góc hướng đặt cho vòng điều khiển trong như sau (góc Yaw ψ_d được chọn cố định):

$$\begin{aligned} u_z &= \sqrt{u_{px}^2 + u_{py}^2 + (u_{pz} + u_b)^2} \\ \psi_d &= 0 \\ \phi_d &= arcsin(\frac{u_{px} sin(\psi_d) - u_{py} cos(\psi_d)}{u_z}) \\ \theta_d &= arctan(\frac{u_{px} cos(\psi_d) + u_{py} sin(\psi_d)}{u_{pz} + u_b}) \end{aligned} \tag{3.14}$$

3.4.2 Điều khiển góc hướng (Attitude Control) với Data-driven PI

Định nghĩa $x_\Theta = [\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T$ phương trình (3.6) ta có thể viết lại thành:

$$\dot{x}_\Theta = F_\Theta x_\Theta + B_\Theta u_\Theta \tag{3.15}$$

Ở đây $F_\Theta = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -c_{11}J_x^{-1} & 0 & -c_{12} & 0 & -c_{13} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -c_{21} & 0 & -c_{22}J_y^{-1} & 0 & -c_{23} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -c_{31} & 0 & -c_{32} & 0 & -c_{33}J_z^{-1} \end{bmatrix}$

$B_\Theta = [e_{6,2}b_{\Theta 1}, e_{6,4}b_{\Theta 2}, e_{6,6}b_{\Theta 3}] \in \mathbb{R}^{6\times 3}$
$b_{\Theta 1} = J_x^{-1} l_\tau k_w$, $b_{\Theta 2} = J_y^{-1} l_\tau k_w$, $b_{\Theta 3} = J_z^{-1} k_t$.
Quỹ đạo đặt giải được từ công thức (3.14) có thể viết được thành $\dot{x}_{\Theta d} = F_{\Theta d} x_{\Theta d}$ và sai lệch $e_\Theta = x_\Theta - x_{\Theta d}$, phương trình mở rộng viết lại thành:

$$\dot{X}_{\Theta d} = \begin{bmatrix} \dot{e}_\Theta \\ \dot{x}_{\Theta d} \end{bmatrix} = \begin{bmatrix} F_\Theta & F_\Theta - F_{\Theta d} \\ 0_{6\times 6} & F_{\Theta d} \end{bmatrix} X_{\Theta d} + \begin{bmatrix} B_\Theta \\ 0_{6\times 3} \end{bmatrix} u_\Theta \tag{3.16}$$

Ma trận $Q_\Theta = \begin{bmatrix} Q_{e\Theta} & 0_{6\times6} \\ 0_{6\times6} & 0_{6\times6} \end{bmatrix}$, trong đó $Q_{e\Theta}$ là ma trận đối xứng xác định dương, $R_\Theta \in \mathbb{R}^{3\times3}$ cũng là ma trận đối xứng xác định dương.
Với hàm chi phí được chọn là:

$$V_\Theta(X_\Theta(t)) = \int_t^\infty e^{-\lambda(\tau-t)}(X_\Theta(\tau)^T Q_e X_\Theta(\tau) + u_\Theta(\tau)^T R u_\Theta(\tau))d\tau \tag{3.17}$$

Như trong chương 3 đã trình bày, tương tự ta có thuật toán Data-driven PI cho điều khiển góc hướng như sau:

Thuật toán 3. *Data-driven PI cho điều khiển góc hướng:*

---

1. *Khởi tạo*
   Bắt đầu với tín hiệu điều khiển ổn định u_Θ^0 và phần nhiễu $u_{\Theta e}$ thêm vào để đảm bảo điều kiện PE. Thu thập dữ liệu và xác định ngưỡng ϵ_Θ

2. *Policy Evaluation*
   Với tín hiệu $u_\Theta^i(X_\Theta)$ giải được từ vòng lặp trước, giải $V_\Theta^{i+1}(X_\Theta)$ và $u_\Theta^{i+1}(X_\Theta)$ từ phương trình:

$$\begin{aligned} V_\Theta^{i+1}(X_\Theta(t+\delta t)) - V_\Theta^{i+1}(X_\Theta(t)) = &- \int_t^{t+\delta t} [X_\Theta(\tau)^T Q_\Theta X_\Theta(\tau) \\ &+ [u_\Theta^i(X_\Theta(\tau))]^T R_\Theta u_\Theta^i(X_\Theta(\tau))]d\tau \\ &+ \int_t^{t+\delta t} \lambda V_\Theta^{i+1}(X_\Theta(\tau))d\tau \\ &+ 2\int_t^{t+\delta t} [u_\Theta^{i+1}(X_\Theta(\tau))]^T R_\Theta u_\Theta^i(X_\Theta(\tau))d\tau \\ &- 2\int_t^{t+\delta t} [u_\Theta^{i+1}(X_\Theta(\tau))]^T R_\Theta [u_\Theta^0(\tau) + u_{\Theta e}]d\tau \end{aligned} \tag{3.18}$$

3. *Policy Improvement*
   Tiếp tục lặp cho đến khi $\|u_\Theta^{i+1} - u_\Theta^i\| < \epsilon_\Theta$
   Cập nhật $u_\Theta^i = u_\Theta^{i+1}$

Để xấp xỉ V_Θ^i và u_Θ^i, Critic-Actor NN được xây dựng như sau:

$$V_\Theta^i(X_\Theta) = w_{V\Theta}^T \varphi_\Theta(X_\Theta) \tag{3.19}$$

$$u_\Theta^i(X_\Theta) = w_{u\Theta}^T \psi_\Theta(X_\Theta) \tag{3.20}$$

Ở đây $\varphi_\Theta(X_\Theta) \in \mathbb{R}^{l_3}$ và $\psi_\Theta(X_\Theta) \in \mathbb{R}^{l_4}$ là 2 vecto hàm cơ sở (l_3 và l_4 lần lượt là số neurons của $\varphi_\Theta(X_\Theta)$ và $\psi_\Theta(X_\Theta)$).$w_{V\Theta} \in \mathbb{R}^{l_3 \times 1}$ và $w_{u\Theta} \in \mathbb{R}^{l_4 \times 3}$ là 2 vecto trọng số tương ứng. Từ đó có thể áp dụng phương pháp Least-Square để giải phương trình (3.18).

3.5 Kết quả mô phỏng

Xét đối tượng với các thông số mô phỏng như sau:
$m = 2.0\,(kg),\ k_w = 1\,(Ns^2),\ k_t = 1\,(Ns^2/m),\ g = 9.8\,(m/s^2), l_\tau = 0.2\,(m)$
$J = diag(5.1, 5.1, 5.2)\ \ (10^{-3} kg.m^2)$
Quỹ đạo đặt mong muốn trong mô phỏng được chọn là quỹ đạo xoắn ốc $p_d = [2sin(at),\ 2cos(at),\ 0.8t]^T$ với $a = 0.5$ là tần số góc. Với quỹ đạo được chọn dễ dàng tính được ma trận A_{pd} thỏa mãn phương trình $\dot{x}_{pd} = A_{pd} x_{pd}$ như sau:

$$A_{pd} = \begin{bmatrix} 0 & 0 & a & 0 & 0 & 0 \\ 0 & 0 & 0 & a & 0 & 0 \\ -a & 0 & 0 & 0 & 0 & 0 \\ 0 & -a & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Từ đó thu được ma trận của hệ sau khi mở rộng theo phương trình 3.9 với A và B tương ứng là:

$$A = \begin{bmatrix} A_p & A_p - A_{pd} \\ 0_{6\times6} & A_{pd} \end{bmatrix} \quad B = \begin{bmatrix} B_p \\ 0_{6\times3} \end{bmatrix}$$

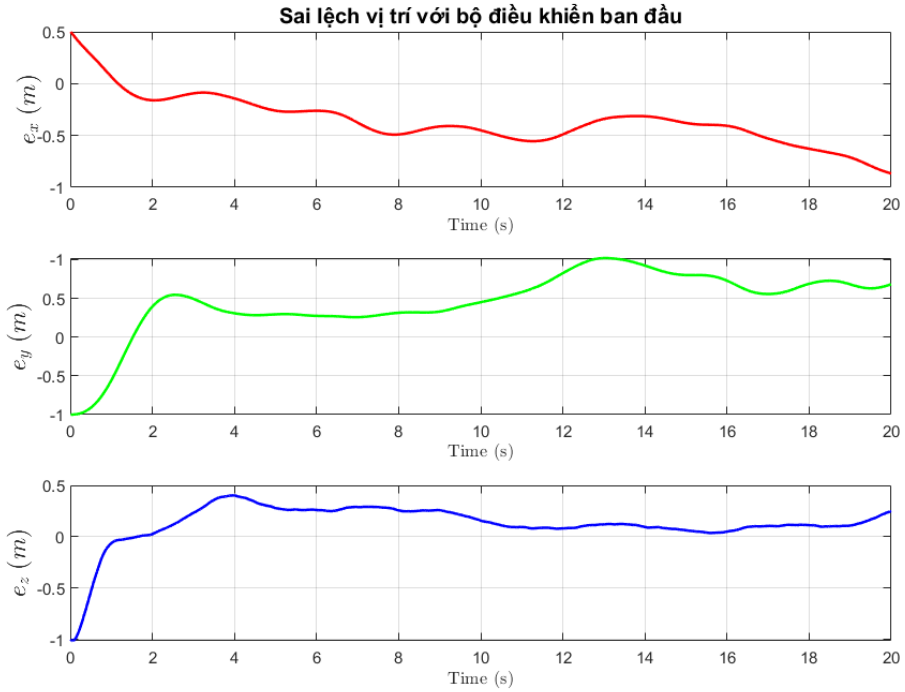


Figure 3.3: Sai lệch bám vị trí với bộ điều khiển ban đầu

Đầu tiên, ở giai đoạn thu thập dữ liệu, ta sử dụng 2 bộ điều khiển PD (Proportion-Derivative) đơn giản cho cả 2 vòng điều khiển vị trí và góc hướng

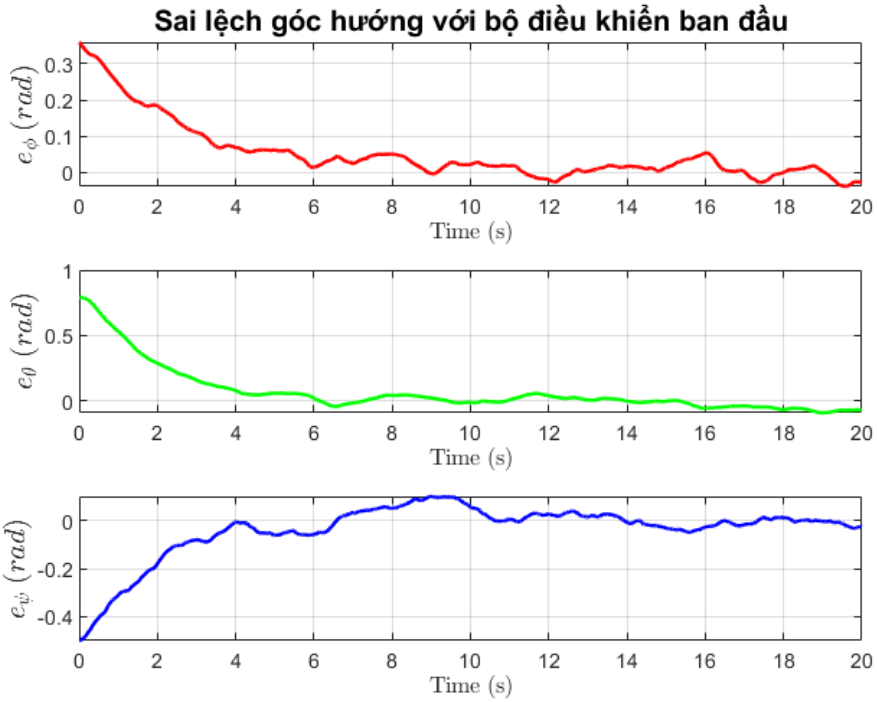


Figure 3.4: Sai lệch bám góc hướng với bộ điều khiển ban đầu

để giữ hệ ổn định trong thời gian đầu. Ngoài ra, để đảm bảo điều kiện PE, tín hiệu nhiễu được thêm vào ở 2 bộ điều khiển, cụ thể $u_{pe} = \sum_{m=1}^{100} 0.01 sin(w_m t)$ với vòng điều khiển vị trí, và $u_{\Theta e} = \sum_{m=1}^{500} 0.002 sin(w_m t)$, trong đó w_m là các tần số được chọn ngẫu nhiên trong khoảng $[-100, 100]$.
Với bộ điều khiển ban đầu dùng để thu thập dữ liệu, ta có sai lệch bám của vị trí và góc hướng như hình (3.3) và (3.4).

Sau khi thu thập dữ liệu, áp dụng 2 thuật toán (3.11) cho vòng điều khiển vị trí và (3.18) cho vòng điều khiển góc hướng với các thông số như sau:

1. Với bộ điều khiển vị trí, chọn ma trận $Q_{ep} = 100 I_6$, $R_p = I_3$ và hệ số $\lambda = 0.01$

2. Với bộ điều khiển góc hướng, chọn ma trận $Q_{e\Theta} = 100 I_6$, $R_\Theta = I_3$ và hệ số $\lambda = 0.01$

Đối với vecto hàm cơ sở được chọn ở cả 2 thuật toán, vecto hàm cơ sở của Actor là hàm bậc nhất, vecto hàm cơ sở của Critic là hàm bậc hai.
Khoảng thời gian thu thập dữ liệu được chọn đối với cả hai thuật toán là $T_{step} = 0.01s$. Kết quả sau khi train, trọng số hội tụ thể hiện trên hình (3.5) và (3.6). Có thể thấy trọng số hội tụ khá nhanh ở cả 2 bộ điều khiển. Ở bộ điều khiển vị trí, trọng số của Actor và Critic hầu như hội tụ sau 4 vòng lặp. Còn với bộ điều khiển góc hướng, sự hội tụ đạt được sau 8 vòng lặp. Sau 20 vòng lặp thì thuật toán dừng lại với sai số $\epsilon_p = \epsilon_\Theta = 10^{-8}$

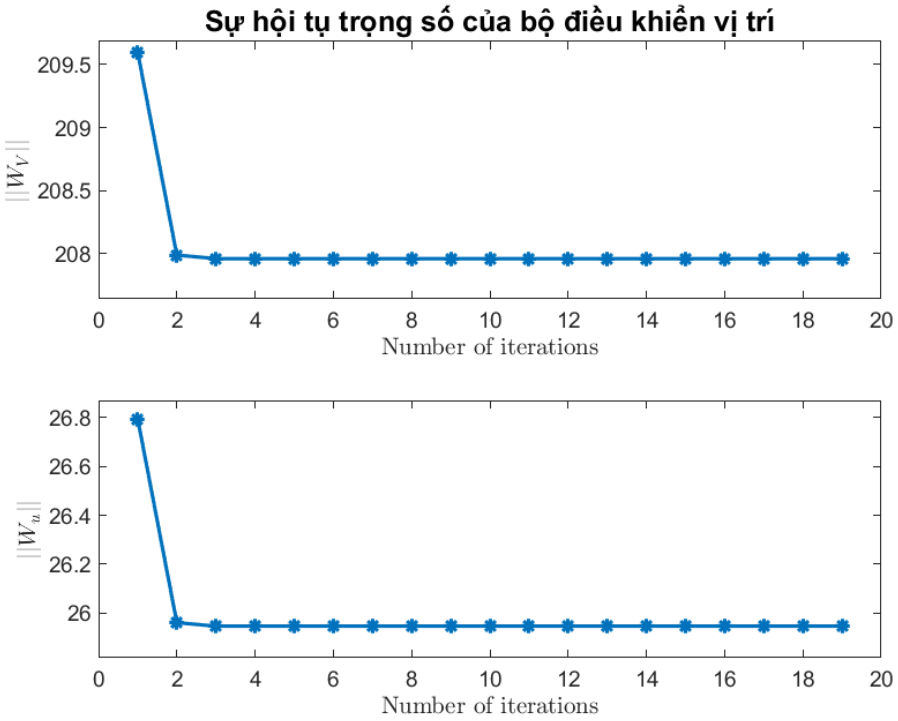


Figure 3.5: Sự hội tụ của norm trọng số ở bộ điều khiển vị trí

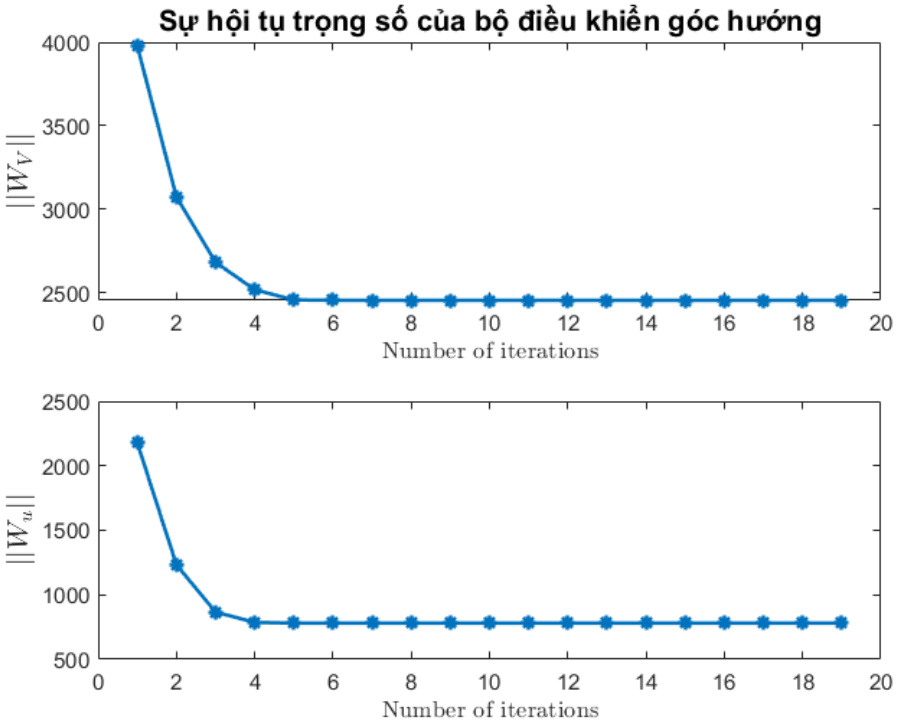


Figure 3.6: Sự hội tụ của norm trọng số ở bộ điều khiển góc hướng

Trọng số hội tụ của Actor ở bộ điều khiển vị trí là:

$$W_{up}^{*} = \begin{bmatrix} -9.9410 & -0.0000 & 0.0000 \\ -11.8122 & -0.0000 & 0.0000 \\ 0.0000 & -9.9410 & -0.0000 \\ 0.0000 & -11.8122 & 0.0000 \\ -0.0000 & -0.0000 & -9.9410 \\ -0.0000 & -0.0000 & -11.8122 \\ -2.3833 & -4.6813 & 0.0000 \\ -9.3901 & 0 & 0 \\ 4.6813 & -2.3831 & 0.0000 \\ -3.7302 & -9.3941 & 0 \\ -0.0000 & -0.0000 & -0.0000 \\ 0.0000 & 0.0000 & -0.0006 \end{bmatrix}$$

Có thể so sánh trọng số của W_{up}^{*} giải được bằng phương pháp Data-driven PI so với việc giải phương trình ARE với riêng bài toán về vị trí này (trong trường hợp biết các thông số về mô hình):

$$A^T P + PA - \lambda P + PBR_p^{-1}B^T P + Q_p = 0 \tag{3.21}$$

Sau khi giải được ma trận P^* ta thu được $K^* = -R_p^{-1}B^T P^*$
như ở dưới, w_{up}^{*} và K^* thu được gần như là tương đồng nhau, một phần do ta chọn hàm kích hoạt $\varphi_p(X_p)$ là hàm bậc nhất, điều đó chứng minh tính đúng đắn của thuật toán đề xuất.

$$K^* = \begin{bmatrix} -9.9401 & 0 & 0 \\ -11.8113 & 0 & 0 \\ 0 & -9.9401 & 0 \\ 0 & -11.8113 & 0 \\ 0 & 0 & -9.9401 \\ 0 & 0 & -11.8113 \\ -2.3843 & -4.6842 & 0 \\ -9.3934 & 3.7319 & 0 \\ 4.6842 & -2.3843 & 0 \\ -3.7319 & -9.3934 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Tương tự như vậy trọng số $W_{u\Theta}^{*}$ tối ưu cho bộ điều khiển góc hướng (Attitude Controller) giải được như sau:

$$W^*_{u\Theta} = \begin{bmatrix} -299.9024 & 0.9506 & 4.6424 \\ -338.1839 & 1.0954 & 1.3983 \\ -3.8254 & -298.8637 & -0.9325 \\ 1.1142 & -338.8784 & 0.4628 \\ -0.3346 & -0.6826 & -301.0987 \\ 0.1909 & 0.2886 & -324.5006 \\ 12.1182 & -22.1137 & -12.5393 \\ -14.4522 & -19.2389 & 3.7538 \\ 7.7199 & 2.0119 & -2.7786 \\ 34.3998 & -45.8008 & -22.3954 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Sử dụng trọng số hội tụ của Actor, áp dụng điều khiển vào hệ ban đầu, ta được kết quả như hình (3.7) và (3.8). Từ (3.7) và (3.8), ta thấy chất lượng bám quỹ đạo khá tốt, sai lệch bám vị trí xấp xỉ 0 sau khoảng 5 s và sai lệch bám góc hướng xấp xỉ 0 sau khoảng 6 s.
Để quan sát rõ hơn chất lượng điều khiển, hình 3.9 thể hiện đồ thị bám vị trí của 3 trục x, y, z và hình 3.10, 3.11 thể hiện quỹ đạo 3D với bộ điều khiển tối ưu.

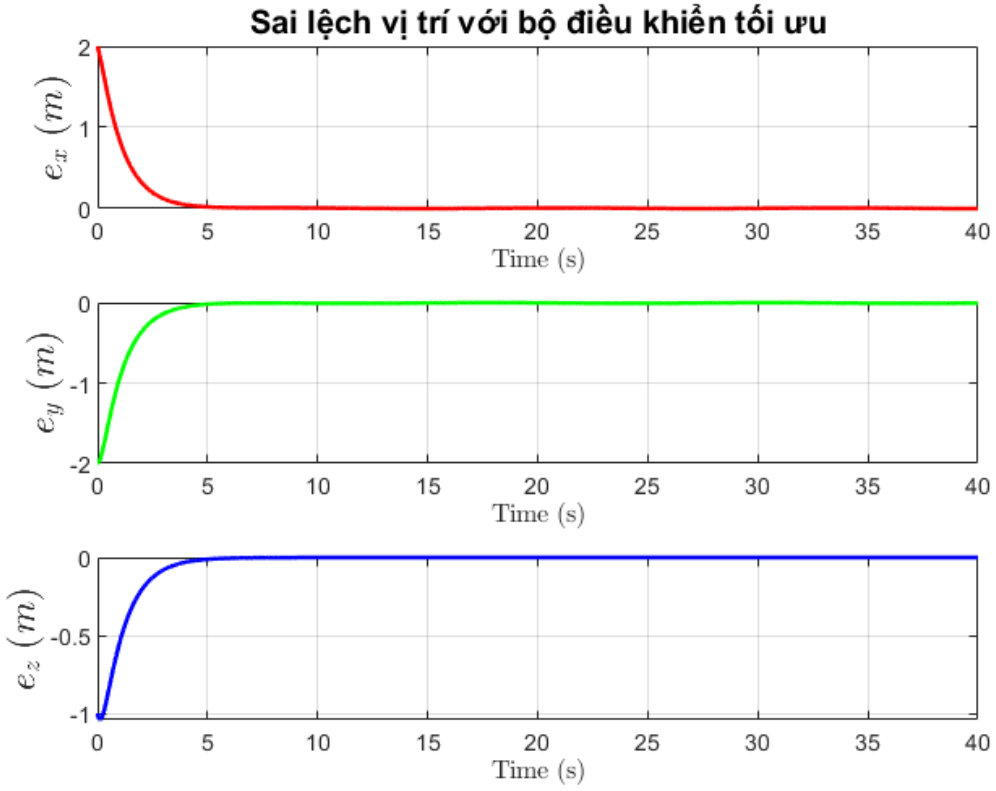


Figure 3.7: Sai lệch bám của vị trí với bộ điều khiển tối ưu

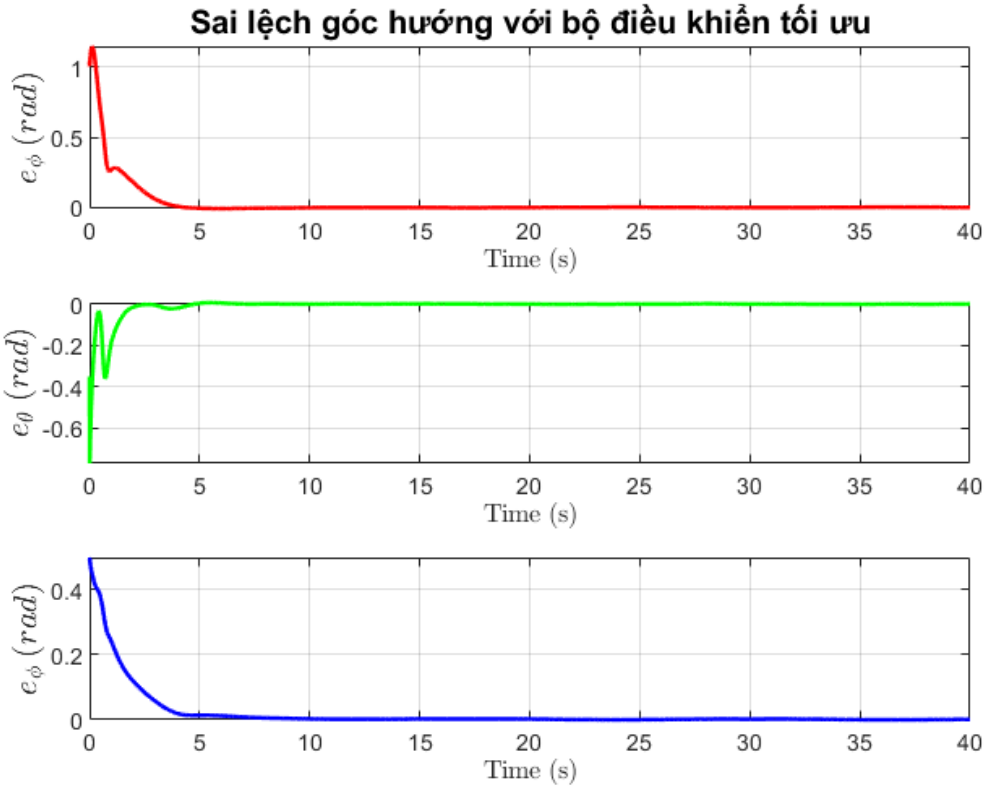


Figure 3.8: Sai lệch bám của góc hướng với bộ điều khiển tối ưu

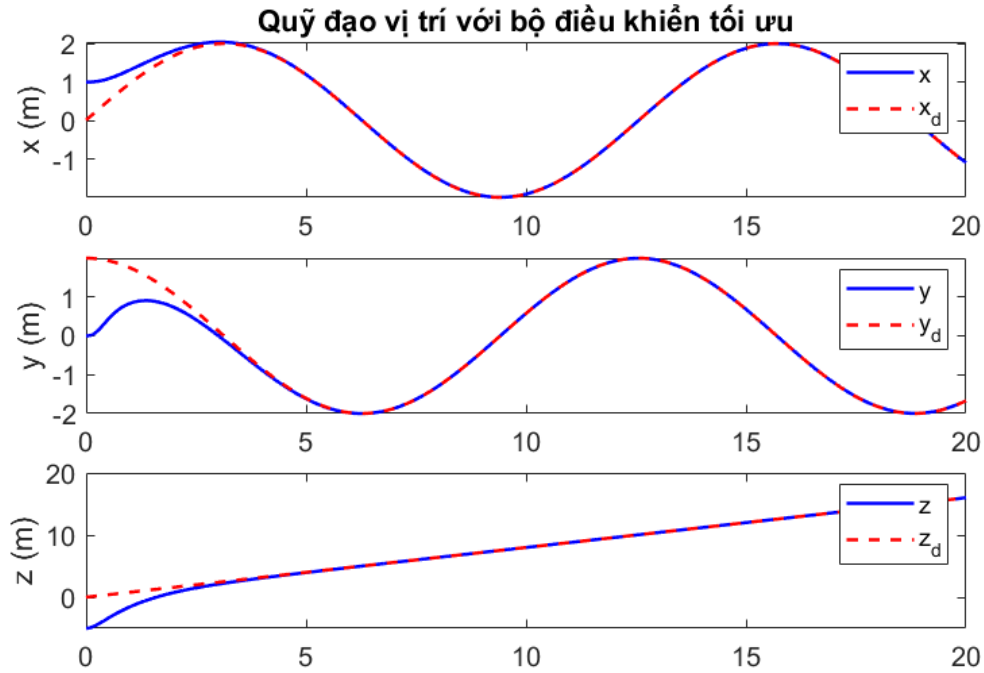


Figure 3.9: Quỹ đạo bám vị trí với bộ điều khiển tối ưu

Ở hình 3.9 và 3.10, đường màu xanh là quỹ đạo thực tế của quadrotor, đường đứt nét màu đỏ là quỹ đạo đặt. Hình chữ nhật nhỏ và hình ellipse nhỏ bao quanh điểm bắt đầu của quỹ đạo thực tế và quỹ đạo đặt của quadrotor.

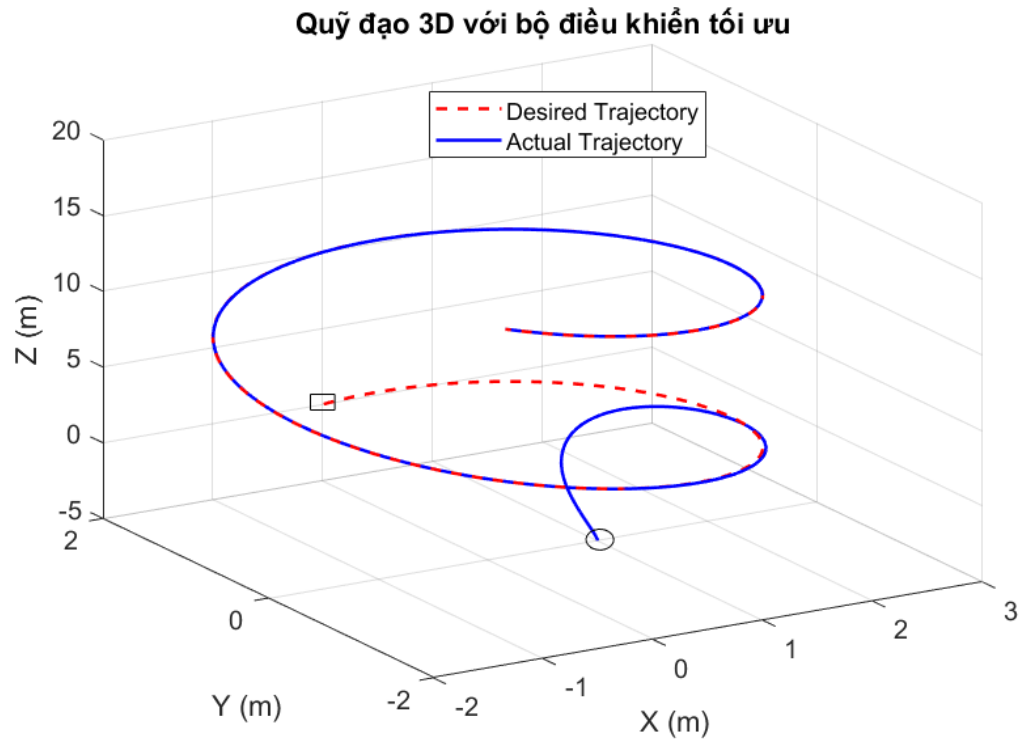


Figure 3.10: Quỹ đạo 3D bám vị trí với bộ điều khiển tối ưu

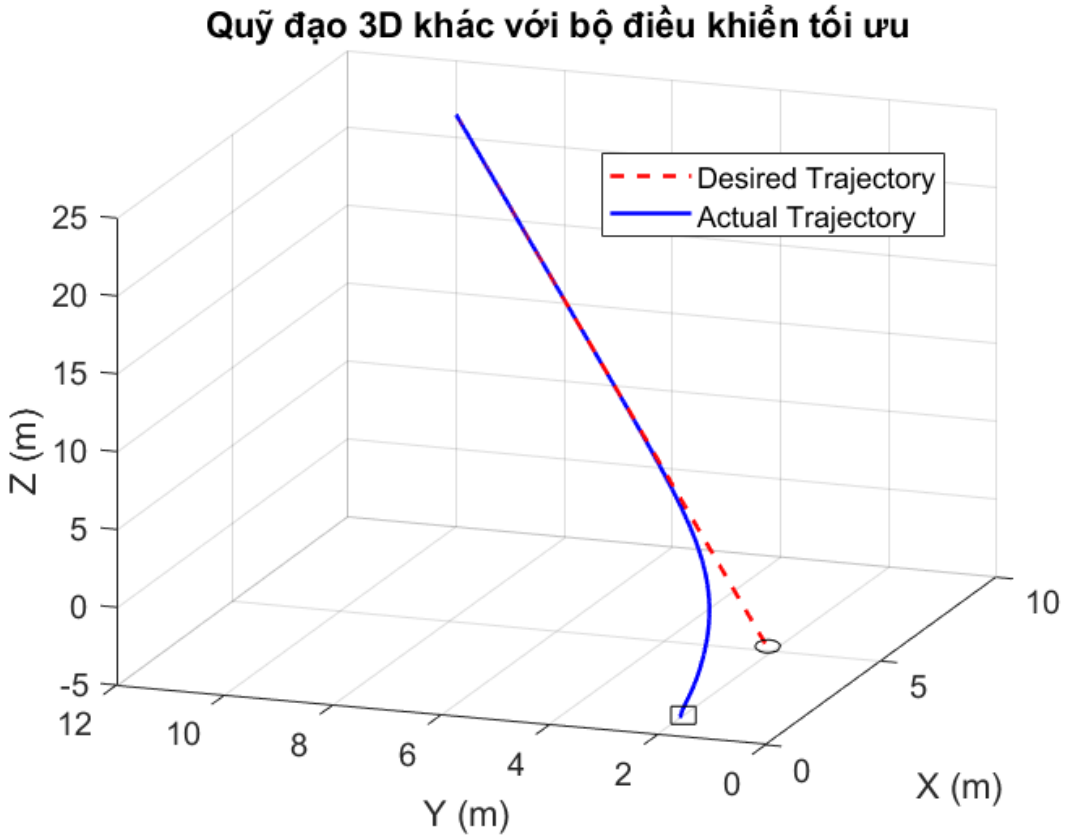


Figure 3.11: Quỹ đạo 3D bám vị trí với bộ điều khiển tối ưu

Để kiểm chứng ảnh hưởng của λ đến chất lượng điều khiển, mô phỏng được tiến hành hoàn toàn tương tự như trên, với 1 tham số thay đổi là $\lambda = 0.5$ và

$\lambda = 1$. Kết quả được thể hiện trên hình (3.11). Từ hình (3.11), ta thấy khi λ tăng, sai lệch bám tăng theo nghĩa dao động với biên độ lớn hơn, điều này phù hợp với phân tích lý thuyết ở chương 2.

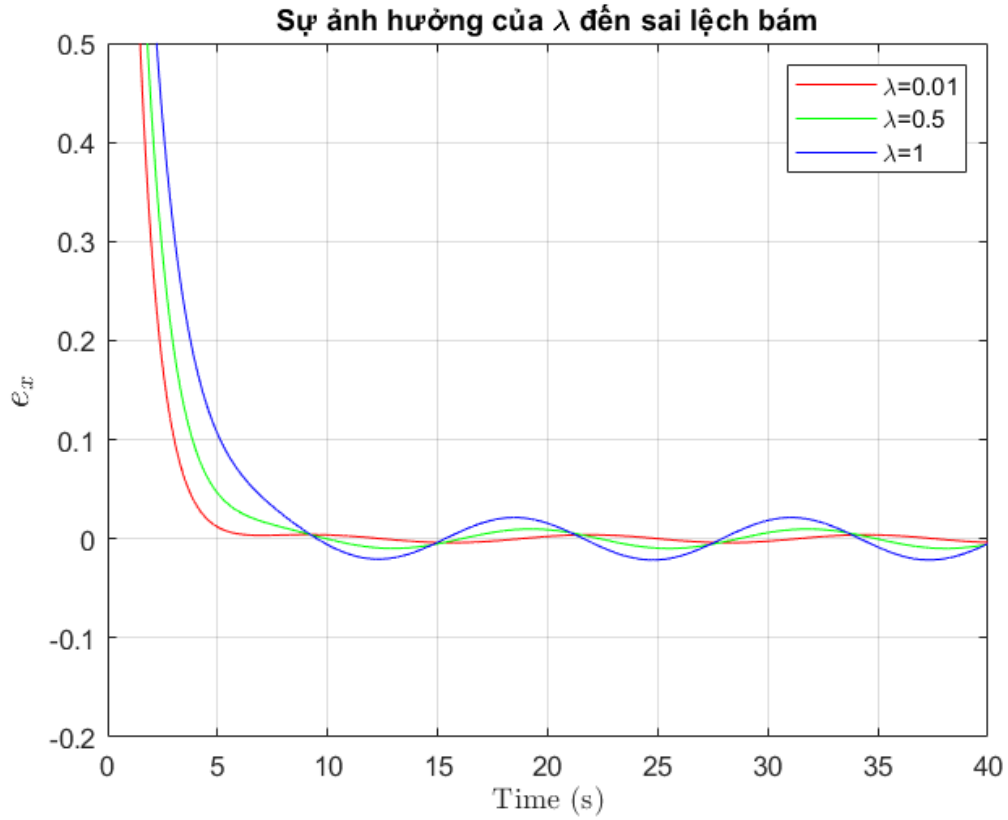


Figure 3.12: Sự ảnh hưởng của λ đến sai lệch bám vị trí

Kết luận

Qua đồ án, chúng ta có thể thấy được sự hiệu quả của thuật toán Data-driven PI, sử dụng dữ liệu để giải quyết bài toán điều khiển bám tối ưu cho hệ tuyến tính và phi tuyến không biết mô hình. Sự hội tụ của nghiệm tối ưu và ổn định của hệ được chứng minh qua các định lý một cách chặt chẽ, rõ ràng (chứng minh chi tiết ở phần phụ lục). Mô phỏng cho quadrotor cho thấy tính khả thi khi áp dụng thuật toán với các đối tượng thực tế.

Tuy nhiên, thuật toán được trình bày còn hạn chế khi chưa xét đến ảnh hưởng của nhiễu tác động và ràng buộc giới hạn đầu vào của hệ thống. Song song với đó, sự phát triển của các hệ bầy đàn đã và đang giải quyết nhiều bài toán điều khiển phức tạp có tính ứng dụng cao. Do đó hướng phát triển tương lai của đồ án là xem xét ảnh hưởng của nhiễu và ràng buộc giới hạn đầu vào, giải bài toán với phương trình HJI, kết hợp điều khiển đội hình và điều khiển tối ưu cho hệ nhiều quadrotor nhằm đạt được tính ứng dụng cao hơn trong thực tế.

Bibliography

[1] https://www.vicon.com/.

[2] *Bellman's Principle of Optimality and its Generalizations*, pages 135–161. Springer US, Boston, MA, 2002.

[3] *REINFORCEMENT LEARNING AND OPTIMAL ADAPTIVE CONTROL*, chapter 11, pages 461–517. John Wiley and Sons, Ltd, 2012.

[4] E. Altug, J.P. Ostrowski, and R. Mahony. Control of a quadrotor helicopter using visual feedback. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 1, pages 72–77 vol.1, 2002.

[5] Abhijit Das, Frank Lewis, and Kamesh Subbarao. Backstepping approach for controlling a quadrotor using lagrange form dynamics. *Journal of Intelligent and Robotic Systems*, 56:127–151, 09 2009.

[6] Bruce Finlayson and L.E. Scriven. The method of weighted residuals - a review. *Appl. Mech. Rev.*, 19:735–748, 01 1966.

[7] Gabriel Hoffmann, Haomiao Huang, Steven Waslander, and Claire Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. 08 2007.

[8] Rushikesh Kamalapurkar, Huyen Dinh, Shubhendu Bhasin, and Warren E. Dixon. Approximate optimal trajectory tracking for continuous-time nonlinear systems. *Automatica*, 51:40–48, 2015.

[9] Bahare Kiumarsi and Frank L. Lewis. Actor–critic-based optimal tracking for partially unknown nonlinear discrete-time systems. *IEEE Transactions on Neural Networks and Learning Systems*, 26(1):140–151, 2015.

[10] Frank Lewis and Draguna Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *Circuits and Systems Magazine, IEEE*, 9:32 – 50, 01 2009.

[11] Yibo Li and Shuxi Song. A survey of control algorithms for quadrotor unmanned helicopter. In *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, pages 365–369, 2012.

[12] D. Liu, Xiong Yang, and H. Li. Adaptive optimal control for a class of continuous-time affine nonlinear systems with unknown internal dynamics. *Neural Computing and Applications*, 23:1843–1850, 2012.

[13] Ashfaq Mian and Wang Daobo. Modeling and backstepping-based nonlinear control strategy for a 6 dof quadrotor helicopter. *Chinese Journal of Aeronautics - CHIN J AERONAUT*, 21:261–268, 06 2008.

[14] Hamidreza Modares and Frank Lewis. Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning. *Automatica*, 50, 07 2014.

[15] Hamidreza Modares and Frank L. Lewis. Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning. *IEEE Transactions on Automatic Control*, 59(11):3051–3056, 2014.

[16] Chaoxu Mu, Changyin Sun, and Wei Xu. Fast sliding mode control on air-breathing hypersonic vehicles with transient response analysis. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 230, 11 2015.

[17] Nguyễn Doãn Phước. *Tối ưu hóa và điều khiển tối ưu*. NXB Bách Khoa Hà Nội, 2010.

[18] Richard Sutton and AG Barto. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11:126–134, 01 1999.

[19] A. Tayebi and S. McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, 14(3):562–571, 2006.

[20] Wei Wang, Hao Ma, and C.-Y Sun. Control system design for multirotor mav. *Journal of Theoretical and Applied Mechanics*, 51:1027–1038, 01 2013.

[21] Haojian Xu, Petros Ioannou, and Majdedin Mirmirani. Adaptive sliding mode control design for a hypersonic flight vehicle. *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM*, 27:829–838, 09 2004.

Phụ lục 1. ***Chứng minh định lí 2.2.3***

Từ (2.20) và (2.21) ta thấy rõ ràng $V^{i+1}(X)$ và $u^{i+1}(X)$ là nghiệm của (2.17) và (2.18) nên để chứng minh định lí trên ta chỉ cần chứng minh sự tồn tại duy nhất của nghiệm phương trình (2.21).

Theo (2.20) và (2.21) ta có:

$$\begin{aligned}\frac{V^{i+1}(X)}{dt} &= \lim_{\Delta t\to 0}\frac{1}{\Delta t}(V^{i+1}(X(t+\Delta t)) - V^{i+1}(X(t))) \\ &= 2\lim_{\Delta t\to 0}\frac{1}{\Delta t}\int_t^{t+\Delta t}[u^{i+1}(X(\tau))]^T R[u^i(X(\tau)) - u(\tau)]d\tau \\ &\quad + \lim_{\Delta t\to 0}\frac{1}{\Delta t}\int_t^{t+\Delta t}\lambda V^{i+1}(X(\tau))d\tau \\ &\quad - \lim_{\Delta t\to 0}\frac{1}{\Delta t}\int_t^{t+\Delta t}[X^T(\tau)QX(\tau) + u^T(\tau)Ru(\tau)]d\tau \\ &= 2[u^{i+1}(X(t))]^T R[u^i(X(t)) - u(t)] + \lambda V^{i+1}(X(t)) \\ &\quad - X^T(t)QX(t) - [u^i(X(t))]^T Ru^i(X(t))\end{aligned} \tag{22}$$

Giả sử rằng tồn tại nghiệm khác $(S(X), v(X))$ của phương trình trên. Ở đây $S(X) \in \mathcal{V}(\mathcal{X})$ và $S(X) \geq 0$, $S(0) = 0$, $v(X) \in \mathcal{U}(\mathcal{X})$. Rõ ràng $(S(X), v(X))$ thỏa mãn:

$$\begin{aligned}\frac{dS(X)}{dt} &= 2[v(X(t))]^T R[u^i(X(t)) - u(t)] + \lambda S(X(t)) \\ &\quad - X^T(t)QX(t) - [u^i(X(t))^T Ru^i(X(t))\end{aligned} \tag{23}$$

Trừ vế với vế của (22) cho (23) ta được:

$$\begin{aligned}&\frac{d}{dt}(V^{i+1}(X) - S(X)) - \lambda(V^{i+1}(X(t)) - S(X(t))) \\ &\quad = 2[u^{i+1}(X(t)) - v(X(t))]^T R[u^i(X(t)) - u(t)]\end{aligned} \tag{24}$$

Nhân cả 2 vế của phương trình trên với $e^{-\lambda t}$ ta được:

$$\begin{aligned}&e^{-\lambda t}\frac{d}{dt}(V^{i+1}(X) - S(X)) - \lambda(V^{i+1}(X(t)) - S(X(t))) \\ &= \frac{d}{dt}(e^{-\lambda t}(V^{i+1}(X)) - S(X)) \\ &= 2e^{-\lambda t}[u^{i+1}(X(t)) - v(X(t))]^T R[u^i(X(t)) - u(t)]\end{aligned} \tag{25}$$

Có thể thấy (25) đúng $\forall u(t) \in \mathbb{R}^m$. Khi $u(t) = u^i(X(t))$, ta được:

$$\frac{d}{dt}[e^{-\lambda t}(V^{i+1}(X) - S(X))] = 0 \tag{26}$$

Do đó:

$$e^{-\lambda t}(V^{i+1}(X) - S(X)) = C \tag{27}$$

Ở đây C là hằng số thực. Như đã đề cập đến $S(0) = 0$, do vậy $C = e^{-\lambda t}(V^{i+1}(0) - S(0))$, vậy $V^{i+1}(X) = S(X)$. Thay kết quả trên vào (25) ta

thấy: $2e^{-\lambda t}[u^{i+1}(X(t)) - v(X(t))]^T R[u^i(X(t)) - u(t)]$ *đúng với mọi* $u(t) \in \mathbb{R}^m$. *Do* R *là ma trận xác định dương và* $u^{i+1}(X(t)) - u(t)$ *sẽ không bằng 0 trong toàn bộ thời gian do đó ta có* $u^{i+1}(X) = v(X(t))$.

Phụ lục 2. ***Chứng minh định lí 2.2.4***

Hệ mở rộng:

$$\dot{X} = F(X) + G(X)\hat{u}^i(X) \tag{28}$$

Đạo hàm nghiệm của phương trình Policy Evaluation, ta được:

$$\begin{aligned}
\dot{V}^i(X) &= (\nabla V^i)^T(F + G\hat{u}^i) \\
&= (\nabla V^i)^T(F + G(u^i + \epsilon_u^i)) + X^T QX \\
&+ (u^{i-1} - u^i)^T R(u^{i-1} - u^i) - \lambda V^i - X^T QX \\
&- (u^{i-1} - u^i)^T R(u^{i-1} - u^i) + \lambda V^i \\
&= (\nabla V^i)^T F - 2(u^i)^T Ru^i \\
&+ (\nabla V^i)^T G\epsilon_u^i + X^T QX + (u^{i-1})^T Ru^{i-1} \\
&- 2(u^i)^T Ru^{i-1} + (u^i)^T Ru^i - \lambda V^i - X^T QX \\
&- (u^{i-1} - u^i)^T R(u^{i-1} - u^i) + \lambda V^i \\
&= (\nabla V^i)^T(F + Gu^{i-1}) + X^T QX + (u^{i-1})^T Ru^{i-1} \\
&- \lambda V^i + (\nabla V^i)^T G\epsilon_u^i - (u^i)^T Ru^i \\
&- X^T QX - (u^{i-1} - u^i)^T R(u^{i-1} - u^i) + \lambda V^i \\
&= (\nabla V^i)^T G\epsilon_u^i - (u^i)^T Ru^i - X^T QX \\
&- (u^{i-1} - u^i)^T R(u^{i-1} - u^i) + \lambda V^i
\end{aligned} \tag{29}$$

Hiển nhiên hệ mở rộng sẽ ổn định tiệm cận nếu điều kiện sau đây được thỏa mãn

$$(u^i)^T Ru^i + X^T QX + (u^{i-1} - u^i)^T R(u^{i-1} - u^i) \geq (\nabla V^i)^T G\epsilon_u^i + \lambda V^i \tag{30}$$

Bởi vì ϵ_u^i *và* λ *đều tiến tới 0, vậy nên điều kiện trên trở thành*

$$(u^i)^T Ru^i + X^T QX + (u^{i-1} - u^i)^T R(u^{i-1} - u^i) \geq 0 \tag{31}$$

Do đó, $\dot{V}^i(X) \leq 0$, *vậy hệ kín ổn định tiệm cận* $\rightarrow$ *sai lệch bám ổn định tiệm cận.*

HỆ QUẢ 1: Xét hệ mở rộng trên với tín hiệu điều khiển xấp xỉ, nếu cả ϵ_u^i *và* λ *đều không bằng* 0, *sai lệch bám có thể bị chặn nhỏ tùy ý bởi việc chọn bộ hàm kích hoạt phù hợp, hệ số* λ *đủ nhỏ và ma trận phạt* Q_e *đủ lớn.*

GIẢ SỬ: Nghiệm của phương trình (2.17) và (2.18) thỏa mãn điều kiện chặn trên $V^i(X) \leq V_M$ *và* $||\nabla V^i|| \leq \nabla V_M$, *ở đây* V_M *và* ∇V_M *là các hằng số dương.*

CHỨNG MINH HỆ QUẢ:

Thành phần đầu tiên trong vế phải của điều kiện (30) phụ thuộc chủ yếu bởi

sai lệch xấp xỉ ϵ_u^i, nó có thể nhỏ tùy ý bởi việc chọn bộ hàm kích hoạt phù hợp dựa theo tính chất xấp xỉ đơn điệu của mạng Neural, ví dụ: $||\epsilon_u^i|| \leq \epsilon_{uM}$, trong đó ϵ_{uM} là một hằng số dương.
Thành phần thứ hai chịu ảnh hưởng bởi cả λ và hàm V^i. Ta chia thành 2 trường hợp sau
- Trường hợp $\lambda = 0$, điều đó có nghĩa rằng tín hiệu điều khiển cũng hội tụ về 0 khi trạng thái hệ hội tụ về 0, điều kiện (30) trở thành

$$(u^i)^T R u^i + X^T Q X + (u^{i-1} - u^i)^T R (u^{i-1} - u^i) \geq \nabla V_M G_M \epsilon_{uM} \tag{32}$$

Có thể dễ dàng thỏa mãn điều kiện trên bởi việc chọn bộ hàm kích hoạt phù hợp để làm cho ϵ_{uM} đủ nhỏ và đạt được điều kiện chặn của sai lệch bám. - Trường hợp $\lambda \neq 0$, điều kiện (30) trở thành:

$$(u^i)^T R u^i + X^T Q X + (u^{i-1} - u^i)^T R (u^{i-1} - u^i) \geq \nabla V_M G_M \epsilon_{uM} + \lambda V_M \tag{33}$$

Từ đó, hệ số λ có thể được chọn đủ nhỏ để thỏa mãn điều kiện chặn của sai lệch bám
Hơn nữa, nhân hai vế của (29) với $e^{\lambda t}$, ta được

$$\frac{\partial}{\partial}(e^{-\lambda t} V^i(X)) = e^{-\lambda t}((\nabla V^i)^T G \epsilon_u^i - (u^i)^T R u^i - X^T Q X - (u^{i-1} - u^i)^T R (u^{i-1} - u^i)) \tag{34}$$

Thành phần đạo hàm bên vế trái của điều kiện trên là âm với $Q_e > 0$ và điều kiện (32) được thỏa mãn, do đó sai lệch bám sẽ giảm cho đến khi thành phần $e^{-\lambda t}$ bằng 0 hoặc điều kiện (32) bất thỏa mãn. Thực tế, chọn Q_e càng lớn sẽ tăng tốc độ giảm của thành phần đạo hàm về 0 và sai lệch bám càng nhỏ. Điều này cũng có nghĩa $V^i(X)$ không đạt tới chặn trên V_M trong điều kiện (33). Do đó, việc chọn Q_e càng lớn cũng khiến sai lệch bám càng nhỏ.

Phụ lục 3. *Điều kiện chặn trên của* λ

Ta xét hệ $\dot{X} = F(X) + G(X)u$ trong công thức (2.9)-(2.11) ở dạng tuyến tính hóa một phần, viết lại thành $\dot{X} = AX + Bu + \tilde{F}(X)$.
Với $\tilde{F}(X)$ là thành phần sau khi tuyến tính hóa một phần để đảm bảo hệ vẫn là hệ phi tuyến.
Ở đây $A = [A_1, A_1 - A_2; 0, A_2]$, $B = [B_1; 0]$, A_1, A_2 là ma trận tuyến tính hóa của $f(x)$ và $r_d(x_d)$.
Ta có thể phát biểu dưới dạng định lí rằng: Để hệ thống $\dot{X} = F(X) + G(X)u$ có nghiệm tối ưu làm sai số hệ thống ổn định tiệm cận thì điều kiện chặn trên của λ phải thỏa mãn:

$$\lambda \leq \bar{\lambda} = 2\|(B_1 R^{-1} B_1^T Q_e)^{1/2}\| \tag{35}$$

Chứng minh:

Từ [3], hàm Hamilton được định nghĩa như sau:

$$\begin{aligned} H^a &= e^{\lambda t} H^b(\mu, u^*) \\ &= e^{\lambda t}(e^{-\lambda t}(X^T Q X + u^{*T} T u^*) + \mu^T (F + G u^*)) \\ &= X^T Q X + u^{*T} R u^* + \nu^T (F + G u^*) \end{aligned} \tag{36}$$

Nghiệm tối ưu u^* thỏa mãn đồng thời 2 phương trình biến trạng thái và biến đồng trạng thái sau:

$$\dot{X} = H^a_\nu(X, \nu) \tag{37}$$

$$\dot{\nu} = \lambda \nu - H^a_X(X, \nu) \tag{38}$$

Ở đây, μ là biến đồng trạng thái, $\nu = e^{\lambda t}\mu$, $H^a_\nu = \partial H^a / \partial \nu$ và $H^a_X = \partial H^a / \partial X$ Từ đó, (36) được biến đổi thành:

$$H^a = X^T Q X + u^{*T} R u^* + \nu^T (AX + Bu^* + \tilde{F}(X)) \tag{39}$$

Mặt khác nghiệm của phương trình (38) tương đương với nghiệm của phương trình HJB trong khi $\nu = \nabla V^*$, thì ν có thể miêu tả là:

$$\nu = 2PX + f_a(X) \equiv \bar{\nu} + f_a(X) \tag{40}$$

Và nghiệm tối ưu:

$$u^* = -R^{-1} B^T P X + f_b(X) \tag{41}$$

Ở đây $f_a(X)$ là phần phi tuyến và $f_b(X)$ phụ thuộc vào $f_a(X)$, $\tilde{F}(X)$ và P. Định nghĩa $P = [P_{11}, P_{12}; P_{21}, P_{22}]$ và sử dụng từ (39)-(41) sai lệch bám động học có thể viết thành:

$$\dot{e}_d = (A_1 - B_1 R^{-1} B_1^T P_{11}) e_d + \tilde{F}_a = A_m e_d + \tilde{F}_a \tag{42}$$

Do đó ta có phương trình ARE dưới đây:

$$Q_e + A_m^T P_{11} + P_{11} A_m - \lambda P_{11} + P_{11} B_1 R^{-1} B_1^T P_{11} = 0 \tag{43}$$

Nhân cả hai vế phương trình trên với e_d^T vào bên trái và e_d vào bên phải ta được:

$$2(Re(\rho) - 0.5\lambda) e_d^T P_{11} e_d = -e_d^T Q_e e_d - e_d^T P_{11} B_1 R^{-1} B_1^T P_{11} e_d \tag{44}$$

Với ρ là nghiệm riêng của A_m, do $P_{11} > 0$ nên ta được:

$$(Re(\rho) - 0.5\lambda) \leq -\|(Q_e P_{11}^{-1})^{1/2}\| \|(P_{11} B_1 R^{-1} B_1^T P_{11})^{1/2}\| \tag{45}$$

Do đó: $Re(\rho) \leq -\|(B_1 R^{-1} B_1^T Q_e)^{1/2}\| + 0.5\lambda$ (Điều phải chứng minh)

Phụ lục 4. *Xác định tín hiệu* u_b

Như ta đã biết tín hiệu u_b có giá trị bằng $\frac{m}{k_w} g e_{3,3}$ để xác định thành phần này ta cần xác định được giá trị $\frac{m}{k_w} g$ trong đó k_w là hằng số khí động thường khó xác

định và thay đổi theo bản chất của cánh quạt và động cơ lắp trên quadrotor. Giá trị m và g hoàn toàn có thể đo được tuy nhiên, phương pháp đề sau có thể giải quyết vấn đề này mà không cần đo cụ thể các đại lượng trên.
Từ phương trình (3.7) ta có thể viết chính xác:

$$u_p = \begin{bmatrix} u_{px} \\ u_{py} \\ u_{pz} \end{bmatrix} = \begin{bmatrix} cos(\phi)sin(\theta)cos(\psi) + sin(\phi)sin(\psi) \\ cos(\phi)sin(\theta)sin(\psi) - sin(\phi)cos(\psi) \\ cos(\phi)cos(\theta) \end{bmatrix} u_z - \begin{bmatrix} 0 \\ 0 \\ \frac{mg}{k_w} \end{bmatrix} \quad (46)$$

Từ công thức trên có thể hiệu chỉnh cho các góc ϕ và θ về 0, khi $u_{pz} = 0$ hay $\ddot{p}_z = 0$ (hiểu một cách đơn giản là lực nâng do các cánh quạt tạo ra cân bằng với trọng lượng của quadrotor) ta xác định được giá trị u_z khi ấy chính bằng $\frac{mg}{k_w}$. Thực nghiệm có thể thực hiện bằng cách:

1. *Đặt vật lên mặt phẳng tuyệt đối (hiệu chỉnh hai góc ϕ và θ về 0)*
2. *Tăng dần gas (throttle) cho đến khi quadrotor vừa mới nhấc lên khỏi mặt phẳng thì thả gas, đo được giá trị u_z khi ấy đúng bằng $\frac{mg}{k_w}$.*
3. *Thực hiện lại nhiều lần và lấy giá trị trung bình*