

**XÂY DỰNG HỆ THỐNG ĐIỀU KHIỂN TRÊN MATLAB-SIMULINK  
THEO KẾT CẤU MA TRẬN**  
DESIGNING CONTROL SYSTEM ON MATLAB-SIMULINK  
UNDER MATRIX STRUCTURE

TS. TRẦN ANH DŨNG

*Khoa Điện - ĐTTB, Trường ĐHHH*

**Tóm tắt**

*Các phương pháp điều khiển trạng thái luôn được thiết kế theo cấu trúc ma trận với nhiều phép toán, phương trình dạng ma trận rất phức tạp. Phần mềm Matlab-Simulink cho phép giải các bài toán điều khiển được viết ở dạng ma trận, làm đơn giản hóa quá trình xây dựng và tránh được nhiều nhầm lẫn không đáng có.*

**Abstract**

*The state - control methods are always designed in matrix structure with many complicated matrix operations and equations. Software Matlab-Simulink allows solving control problems composed in matrix form to simplify designing process and to avoid unnecessary errors.*

**1. Đặt vấn đề**

Trong lĩnh vực điều khiển tự động, mô hình trạng thái là dạng mô hình toán mô tả đầy đủ nhất đối tượng điều khiển. Có thể nói, mô hình trạng thái là một kho động học của đối tượng. Từ mô hình trạng thái, các lý thuyết điều khiển trạng thái, cả kinh điển và hiện đại, đã phát triển rất mạnh. Có thể kể ra một vài phương pháp điều khiển trạng thái điển hình như: điều khiển gán điểm cực, điều khiển tối ưu, điều khiển bám, điều khiển thích nghi, điều khiển bền vững... Trong hầu hết các trường hợp, người ta không thể đo được hết các biến trạng thái của đối tượng, cho nên để thực hiện thuật toán điều khiển cần phải có các bộ quan sát trạng thái mà điển hình trong số đó là bộ quan sát Luenberger và bộ quan sát (bộ lọc) Kalman. Đặc điểm chung và nổi bật nhất của mô hình trạng thái, bộ điều khiển trạng thái và bộ quan sát trạng thái là chúng đều được tổng hợp theo kết cấu ma trận bằng những phương trình ma trận phức tạp, gây nhiều khó khăn cho việc khảo sát, mô phỏng hệ thống.

Phần mềm Matlab (Matrix Laboratory) hiện nay được sử dụng rất rộng rãi để lập trình, khảo sát, phân tích và xây dựng các hệ thống điều khiển tự động. Matlab đặc biệt mạnh khi xử lý, thao tác và thực hiện các phép toán ma trận, kể cả phần ứng dụng Simulink của nó. Khai thác thế mạnh này của Matlab-Simulink để làm đơn giản hóa lời giải cho các bài toán điều khiển trạng thái là vấn đề hiện nay ít được đề cập đến và là nội dung chính của bài báo này.

**2. Mô hình trạng thái và các phương pháp điều khiển trạng thái**

Một đối tượng điều khiển MIMO (nhiều đầu vào, nhiều đầu ra) thường được miêu tả bằng một mô hình trạng thái có dạng như sau:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du}, \end{aligned} \quad (1)$$

trong đó:  $\mathbf{u} = [u_1 \ u_2 \dots \ u_m]^T$  - véc tơ đầu vào,  $\mathbf{y} = [y_1 \ y_2 \dots \ y_r]^T$  - véc tơ đầu ra,  $\mathbf{x} = [x_1 \ x_2 \dots \ x_n]^T$  - véc tơ trạng thái, A - ma trận hệ thống, B - ma trận điều khiển, C, D - các ma trận đầu ra.

Các phương pháp điều khiển trạng thái dựa trên mô hình trạng thái của đối tượng đã được nghiên cứu và trình bày trong rất nhiều tài liệu về điều khiển tự động, trong đó điển hình là một số phương pháp có khả năng ứng dụng mạnh sau:

- *Điều khiển gán điểm cực*: điều khiển gán điểm cực theo phương pháp Modal là loại điều khiển phản hồi trạng thái, trong đó cần phải xác định các hệ số phản hồi sao cho hệ thống đạt

được các điểm cực cho trước theo mong muốn. Bộ điều khiển Modal K phải thỏa mãn phương trình:

$$\det(s\mathbf{I} - \mathbf{A} + \mathbf{BK}) = (s-s_1)(s-s_2)\dots(s-s_n), \quad (2)$$

trong đó:  $s_1, s_2, \dots, s_n$  - các điểm cực mong muốn.

Ma trận K có thể xác định dễ dàng nhờ câu lệnh Matlab sau:

$$\mathbf{K} = \text{place}(\mathbf{A}, \mathbf{B}, [s_1 \ s_2 \dots \ s_n]) \quad (3)$$

- *Điều khiển tối ưu*: cần phải tổng hợp được bộ điều khiển phản hồi trạng thái K để hệ thống

có phẩm hàm chất lượng dạng toàn phương  $\mathbf{Q} = \frac{1}{2} \int_0^{\infty} (\mathbf{x}^T \mathbf{R}_1 \mathbf{x} + \mathbf{u}^T \mathbf{R}_2 \mathbf{u}) dt$  đạt cực tiểu. Theo lý thuyết về điều khiển tối ưu [1], ma trận K có thể xác định bằng công thức:

$$\mathbf{K} = \mathbf{R}_2^{-1} \mathbf{B}^T \mathbf{P}, \quad (4)$$

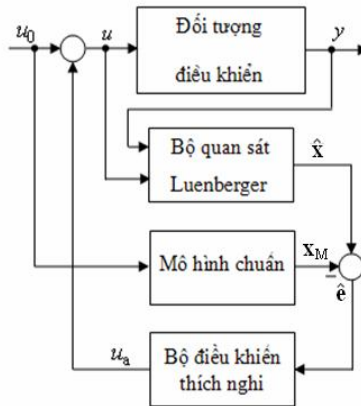
trong đó ma trận P thỏa mãn phương trình Riccati có dạng:

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{R}_1 - \mathbf{P} \mathbf{B} \mathbf{R}_2^{-1} \mathbf{B}^T \mathbf{P} = 0 \quad (5)$$

Công việc khó khăn nhất khi tổng hợp bộ điều khiển tối ưu dạng toàn phương là phải giải phương trình Riccati. Đây là dạng phương trình ma trận phi tuyến, nó sẽ càng phức tạp khi bậc của đối tượng càng cao. Tuy nhiên trong chương trình Matlab ma trận K có thể xác định một cách dễ dàng nhờ các câu lệnh sau:

$$\mathbf{P} = \text{care}(\mathbf{A}, \mathbf{B}, \mathbf{R}_1, \mathbf{R}_2); \quad \mathbf{K} = \text{inv}(\mathbf{R}_2)^* \mathbf{B}^* \mathbf{P} \quad (6)$$

- *Điều khiển thích nghi*: thuật toán điều khiển thích nghi trực tiếp với mô hình chuẩn đã được đề cập đến trong [2]. Đối với một đối tượng SISO (một đầu vào, một đầu ra), hệ điều khiển thích nghi trực tiếp sử dụng bộ quan sát trạng thái Luenberger có cấu trúc như hình 1.



Hình 1. Cấu trúc hệ điều khiển thích nghi trực tiếp với bộ quan sát trạng thái.

Thuật toán thích nghi trực tiếp với chỉnh định thông số, mô hình chuẩn và bộ quan sát có dạng sau:

$$u_a = \mathbf{k}_A \hat{\mathbf{x}} + k_b u_0 \quad (7)$$

$$\left. \begin{aligned} \dot{\mathbf{k}}_A &= -\gamma_a \mathbf{B}_M^T \mathbf{P} \hat{\mathbf{x}}^T - \lambda_a \mathbf{k}_A \\ \dot{k}_b &= -\gamma_b \mathbf{B}_M^T \mathbf{P} \hat{e} u_0^T - \lambda_b k_b \end{aligned} \right\} \quad (8)$$

trong đó:  $\mathbf{k}_A$  và  $k_b$  – véc tơ và hệ số chỉnh định luật thích nghi (7);  $\hat{\mathbf{e}} = \hat{\mathbf{x}} - \mathbf{x}_M$  – véc tơ sai lệch,  $\hat{\mathbf{x}}$  - véc tơ trạng thái của bộ quan sát Luenberger có dạng [1]:

$$\left. \begin{aligned} \dot{\hat{\mathbf{x}}} &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{L}(\hat{y} - y) + \mathbf{B}u \\ \hat{y} &= \mathbf{C}\hat{\mathbf{x}} + \mathbf{D}u, \end{aligned} \right\} \quad (9)$$

$\mathbf{x}_M$  – véc tơ trạng thái của mô hình chuẩn có dạng:

$$\dot{\mathbf{x}}_M = \mathbf{A}_M \mathbf{x}_M + \mathbf{B}_M u_0, \quad (10)$$

$\mathbf{A}_M$ ,  $\mathbf{B}_M$  – các ma trận hệ số;  $\gamma_a$ ,  $\lambda_a$ ,  $\gamma_b$ ,  $\lambda_b$  – các hệ số khuếch đại thuật toán;  $\mathbf{P}$  – ma trận đối xứng xác định dương thỏa mãn phương trình Lyapunov có dạng:

$$\mathbf{A}_M^T \mathbf{P} + \mathbf{P} \mathbf{A}_M = -\mathbf{G}, \quad (11)$$

$\mathbf{G}$  – ma trận đối xứng xác định dương bất kỳ.

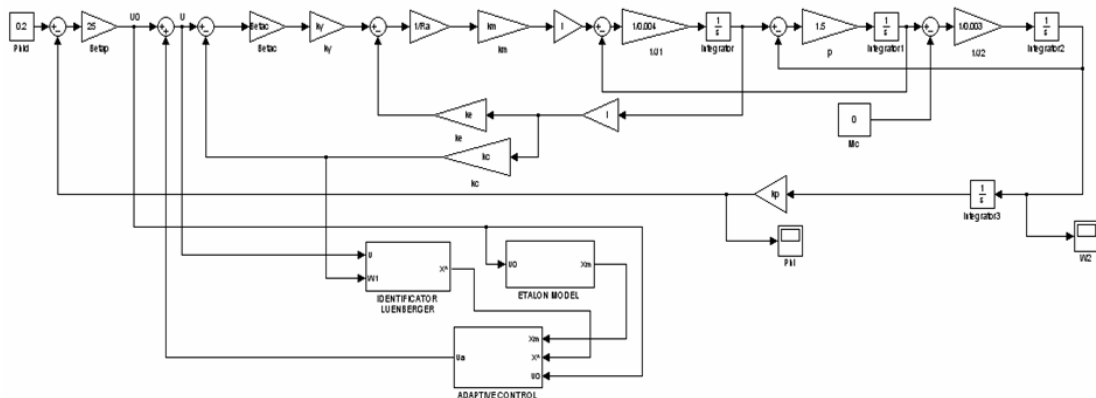
Để giải phương trình Lyapunov có thể sử dụng câu lệnh Matlab sau:

$$\mathbf{P} = \text{lyap}(\mathbf{A}_M, \mathbf{G}) \quad (12)$$

Thuật toán điều khiển thích nghi như trình bày ở trên bao gồm nhiều phép toán, phương trình ma trận phức tạp. Để thực hiện thuật toán này thuận tiện hơn cả là sử dụng công cụ mô phỏng Simulink của phần mềm Matlab.

### 3. Xây dựng mô hình Simulink theo kết cấu ma trận

Simulink là một công cụ ứng dụng của Matlab để xây dựng mô hình mô phỏng các hệ thống. Mô hình trên Simulink được xây dựng theo kết cấu sơ đồ khối bao gồm nhiều khối cơ bản ghép nối với nhau. Ưu điểm của mô hình Simulink là trực quan, dễ xây dựng, thuận tiện cho việc sử dụng, khảo sát, phân tích và tổng hợp hệ thống. Công cụ Simulink cũng có những thế mạnh của phần mềm Matlab, một trong số đó là khả năng thao tác trên các ma trận. Khả năng này của Simulink hiện nay ít được đề cập đến trong các tài liệu về điều khiển tự động cũng như về phần mềm Matlab.

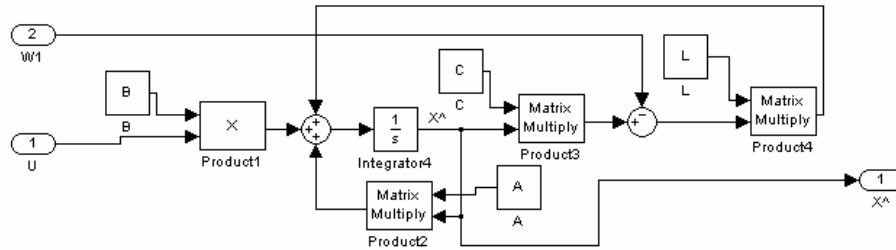


**Hình 2. Hệ thống điều khiển thích nghi trực tiếp cho đối tượng bậc 3 với bộ quan sát.**

Trên hình 2 là mô hình Simulink hệ thống điều khiển thích nghi trực tiếp với bộ quan sát theo thuật toán (7), (8), (9), (10), (11) dùng cho đối tượng liên kết đàn hồi 2 khối có bậc 3 [3], trong đó bộ quan sát Luenberger, mô hình chuẩn và bộ điều khiển thích nghi được đưa vào các khối Subsystem (hệ con) nhằm giảm độ phức tạp cho mô hình tổng thể, tránh nhầm lẫn, chông chéo trong quá trình xây dựng và làm cho hệ thống có cấu trúc tích hợp.

Các hệ con đều được xây dựng trực tiếp từ các phương trình ma trận miêu tả chúng đã đưa ra ở trên mà không cần phải biến đổi về các phép toán vô hướng. Rất nhiều các khối cơ bản trong thư viện của Simulink đều có chức năng làm việc với ma trận và các phép toán ma trận như khối tổng, khối nhân, khối tích phân... Chức năng này là mặc định của các khối cơ bản, cách sử dụng

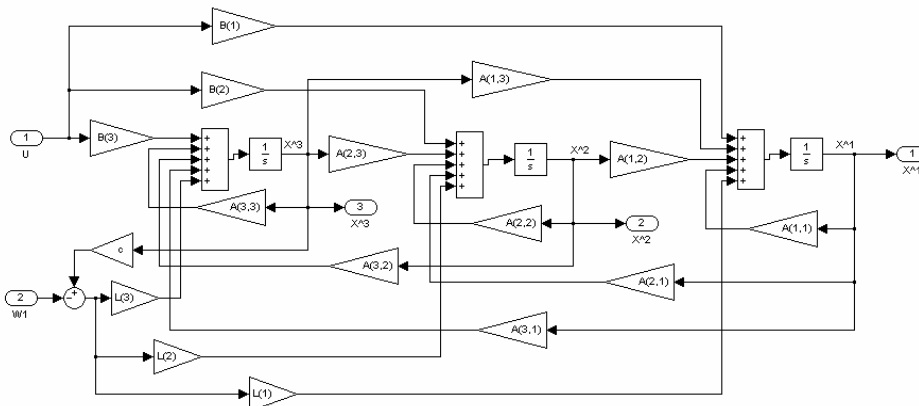
chúng khi làm việc với các phép toán ma trận không khác gì khi làm việc với các phép toán vô hướng như đa số người sử dụng vẫn thường làm, chỉ trừ một vài trường hợp phải vào cửa sổ tham số của khối (chẳng hạn như khối nhân) để chuyển thao tác vô hướng sang thao tác ma trận.



Hình 3. Bộ quan sát Luenberger theo kết cấu ma trận.

Để làm phép so sánh, trên hình 3 đưa ra mô hình Simulink bộ quan sát Luenberger của đối tượng liên kết đàn hồi 2 khối xây dựng theo kết cấu ma trận và trên hình 4 đưa ra mô hình Simulink bộ quan sát Luenberger của cùng đối tượng nhưng xây dựng theo cách thông thường.

Rõ ràng rằng, mô hình theo kết cấu ma trận gọn nhẹ hơn và dễ kiểm soát hơn rất nhiều, mà đây mới chỉ là đối tượng bậc 3, độ phức tạp chưa lớn. Khi bậc của đối tượng tăng lên, mô hình thông thường sẽ phình ra to hơn, nhiều đường nối chằng chéo hơn, trong khi mô hình theo kết cấu ma trận sẽ không thay đổi gì.



Hình 4. Bộ quan sát Luenberger xây dựng theo cách thông thường.

Một điều rất quan trọng cho người sử dụng cần phải nhấn mạnh rằng, các mô hình Simulink xây dựng theo kết cấu ma trận đều có thể dùng để điều khiển đối tượng thực trên miền thời gian thực sử dụng Card ghép nối Matlab với ngoại vi [4].

**4. Kết luận**

Phần mềm Matlab-Simulink với những công cụ xử lý, tính toán trên ma trận rất mạnh giúp người sử dụng dễ dàng kiểm nghiệm, phân tích, đánh giá và tổng hợp các hệ thống điều khiển tự động trong không gian trạng thái. Việc xây dựng mô hình các hệ thống điều khiển tự động trên Simulink theo kết cấu ma trận làm rút ngắn quá trình xây dựng, mô hình trở nên gọn nhẹ hơn, tránh được những sai sót khi xây dựng các mô hình công kênh. Xây dựng mô hình theo kết cấu ma trận góp phần làm tổng quát hóa các thuật toán điều khiển trạng thái, mô hình trở nên phổ quát không phụ thuộc vào bậc của đối tượng, dẫn đến không phải xây dựng lại mô hình đối với các đối tượng khác nhau.

**TÀI LIỆU THAM KHẢO**

[1] Nguyễn Doãn Phước - Lý thuyết điều khiển tuyến tính - Nhà xuất bản Khoa học và Kỹ thuật - 2005.

- [2] Trần Anh Dũng - *Điều khiển thích nghi trực tiếp hệ điện-cơ phi tuyến ba khối có biến dạng đàn hồi* - Tạp chí Khoa học - Công nghệ Hàng hải - Số 18, 2009.
- [3] Trần Anh Dũng - *Điều khiển nở ron-mờ hệ điện-cơ phi tuyến có biến dạng đàn hồi* - Tạp chí Khoa học - Công nghệ Hàng hải - Số 15, 2008.
- [4] Trần Anh Dũng, Phạm Tuấn Anh - *Thiết kế điều khiển mờ lai cho hệ truyền động điện một chiều* - Tạp chí Khoa học - Công nghệ Hàng hải - Số 20, 2009.

Người phản biện: TS. Hoàng Đức Tuấn

## CHẾ ĐỘ LÁI VƯỢT CẤP TRONG HỆ THỐNG LÁI TỰ ĐỘNG TÀU THUYỀN AUTO- OVERRIDING MODE OF AUTO PILOTAGING ON SHIPS

ThS. BÙI VĂN DŨNG  
Khoa Điện - ĐTTB, Trường ĐHHH

### Tóm tắt

Hiện nay trong vận hành khai thác hệ thống lái tự động trên tàu thuyền, một số Sĩ quan, Thủy thủ hàng hải và một số người sử dụng hệ thống lái tự động vẫn còn quan niệm sai và chưa hiểu sâu sắc chức năng của chế độ lái vượt cấp "Auto override mode", cũng như khi nào thì hệ thống lái sử dụng góc giới hạn của bánh lái ở góc  $35^\circ$  hay góc  $45^\circ$ . Bài báo giới thiệu về khái niệm, cấu trúc cũng như việc sử dụng chế độ lái vượt cấp của hệ thống lái tự động hiện nay được trang bị trên tàu thuyền.

### Abstract

At present, when operating auto pilotaging system on ships, some officers, sailors and users of auto pilot system still make misconceptions and do not understand thoroughly the function of Auto override mode, as well as when using limit switches of rudder angle at  $35^\circ$  or  $45^\circ$ . The article introduces the concept, structure and the usage of Auto overriding mode of auto pilotaging on ships.

### 1. Khái niệm và bản chất của chế độ lái vượt cấp (Auto override steering Mode)

Chúng ta biết rằng, hệ thống lái tàu thuyền hiện đại và đầy đủ đều có các chế độ lái sau đây:

#### a. Chế độ lái tay (Hand) bao gồm 2 chế độ nhỏ là:

- Chế độ lái đơn giản (Non Follow-up, hay Lever);
- Chế độ lái lặp hay còn gọi là lái truy theo (Follow-up), trong chế độ này có thể có chế độ lái lặp tại chỗ (local) và chế độ lái lặp từ xa (Remote).

#### b. Chế độ lái tự động (Auto).

#### c. Chế độ lái sự cố (Emergency)

Vậy chế độ lái vượt cấp (Override Mode) thuộc chế độ lái nào?

Một số người cho rằng chế độ này là chế độ lái sự cố (Emergency steering Mode), và là một trong các chế độ lái hoạt động độc lập của hệ thống lái tự động. Quan niệm như vậy là sai lệch, vì chế độ lái vượt cấp không phải là chế độ lái sự cố của hệ thống lái, mà là chế độ lái bình thường mà bản chất của nó khác hoàn toàn với chế độ lái sự cố (Emergency steering Mode). Vậy bản chất của các chế độ lái như thế nào?

Bản chất của chế độ lái vượt cấp (Override) là chế độ lái tương tự như chế độ lái đơn giản (Non Follow-up, hay Lever), tức là: ở chế độ lái này tín hiệu điều khiển từ cần điều khiển lái được tác động trực tiếp đến phần tử thực hiện là van điện từ (Solenoid valve) giống như chế độ lái đơn giản, điều này hoàn toàn khác với chế độ lái sự cố là chế độ lái mà khi điều khiển không cần sử dụng đến van điện từ mà tín hiệu điều khiển được tác động trực tiếp đến hệ thống thủy lực thông qua bơm thủy lực. Điểm khác biệt nữa giữa chế độ lái vượt cấp với chế độ lái đơn giản là chế độ lái vượt cấp được thực hiện hoàn toàn tự động khi có tín hiệu lái vượt cấp trong khi công tắc chọn chế độ lái đang đặt tại vị trí tự động hoặc chế độ lái lặp, tức là hệ thống lái đang công tác ở chế độ tự động (Auto) hoặc chế độ lái lặp (Follow-up), nhưng vì cần điều khiển khẩn cấp ở góc bẻ lái lớn hơn và tác động nhanh hơn để nhằm điều động tránh va thật an toàn của Sĩ quan đi ca khi gặp