

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
ĐẠI HỌC THÁI NGUYÊN**

**NGUYỄN THỊ MAI HƯƠNG**

**NGHIÊN CỨU XÂY DỰNG THUẬT TOÁN ĐIỀU KHIỂN  
DỰ BÁO THEO MÔ HÌNH CHO ĐỐI TƯỢNG  
PHI TUYẾN LIÊN TỤC**

**LUẬN ÁN TIẾN SĨ KỸ THUẬT**

**THÁI NGUYÊN – NĂM 2016**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
ĐẠI HỌC THÁI NGUYÊN**

**NGUYỄN THỊ MAI HƯƠNG**

**NGHIÊN CỨU XÂY DỰNG THUẬT TOÁN ĐIỀU KHIỂN  
DỰ BÁO THEO MÔ HÌNH CHO ĐỐI TƯỢNG  
PHI TUYẾN LIÊN TỤC**

Chuyên ngành: Kỹ thuật điều khiển và Tự động hóa  
Mã số: 62 52 02 16

**LUẬN ÁN TIẾN SĨ KỸ THUẬT**

NGƯỜI HƯỚNG DẪN KHOA HỌC:

**PGS.TS Lại Khắc Lãi**

THÁI NGUYÊN – NĂM 2016

## LỜI CẢM ƠN

Trong quá trình làm luận án, tôi đã nhận được rất nhiều góp ý về chuyên môn cũng như sự ủng hộ về các công tác tổ chức của tập thể cán bộ hướng dẫn, của các nhà khoa học, của các bạn đồng nghiệp. Tôi xin được gửi tới họ lời cảm ơn sâu sắc.

Tôi xin bày tỏ lòng cảm ơn đến tập thể cán bộ hướng dẫn đã tâm huyết hướng dẫn tôi trong suốt thời gian qua.

Tôi cũng xin chân thành cảm ơn các đồng nghiệp, tập thể các nhà khoa học trường Đại học Kỹ thuật Công nghiệp, của bộ môn Điều khiển tự động trường Đại học Bách khoa Hà Nội, đã có những ý kiến đóng góp quý báu, các Phòng ban của Trường Đại học Kỹ thuật Công nghiệp đã tạo điều kiện thuận lợi cho tôi trong suốt quá trình thực hiện đề tài luận án.

*Thái Nguyên, ngày tháng 01 năm 2016*

Tác giả luận án

Nguyễn Thị Mai Hương

## MỤC LỤC

|   |            |
|---|------------|
| <b>LỜI CAM ĐOAN</b>   | <b>i</b>   |
| <b>LỜI CẢM ƠN</b>   | <b>ii</b>  |
| <b>MỤC LỤC</b>  | <b>iii</b> |
| <b>MỞ ĐẦU</b>   | <b>1</b>   |
| 1. Giới thiệu .....   | 1          |
| 2. Tính cấp thiết của luận án.....  | 2          |
| 3. Mục tiêu của luận án .....   | 4          |
| 4. Đối tượng, phạm vi và phương pháp nghiên cứu .....   | 4          |
| 5. Ý nghĩa khoa học và thực tiễn .....  | 5          |
| 5.1. Ý nghĩa khoa học.....  | 5          |
| 5.2. Ý nghĩa thực tiễn .....  | 5          |
| 6. Bố cục luận án.....  | 6          |
| <b>CHƯƠNG 1</b>   | <b>8</b>   |
| <b>TỔNG QUAN VỀ ĐIỀU KHIỂN DỰ BÁO CHO HỆ PHI TUYẾN</b>  | <b>8</b>   |
| 1.1. Tổng quan các công trình nghiên cứu về điều khiển dự báo hệ phi tuyến trên thế giới.....                                 | 9          |
| 1.2. Các phương pháp quy hoạch phi tuyến .....  | 18         |
| 1.2.2. Bài toán tối ưu hóa phi tuyến bị ràng buộc gồm: Kỹ thuật hàm phạt và hàm chặn, Phương pháp SQP [3], [5] và GA [2]..... | 19         |
| 1.3. Các phương pháp điều khiển tối ưu .....  | 19         |
| 1.4. Các công trình nghiên cứu về điều khiển dự báo hệ phi tuyến trong nước .....   | 20         |
| 1.5. Những vấn đề cần tiếp tục nghiên cứu về điều khiển dự báo cho hệ phi tuyến và hướng nghiên cứu của luận án .....         | 21         |
| 1.6. Kết luận chương 1 .....  | 23         |

|   |           |
|---|-----------|
| <b>CHƯƠNG 2</b>   | <b>24</b> |
| <b>ĐIỀU KHIỂN DỰ BÁO HỆ PHI TUYẾN TRÊN NỀN CÁC PHƯƠNG PHÁP QUY HOẠCH PHI TUYẾN</b>  | <b>24</b> |
| 2.1. Nguyên lý làm việc của điều khiển dự báo phi tuyến.....  | 24        |
| 2.1.1. Cấu trúc bộ điều khiển dự báo .....  | 26        |
| 2.1.2. Kỹ thuật cài đặt bộ điều khiển dự báo trên nền các phương pháp quy hoạch phi tuyến .....   | 29        |
| 2.2. Áp dụng vào điều khiển dự báo lớp hệ song tuyến.....   | 31        |
| 2.2.1. Thuật toán điều khiển dự báo phi tuyến cho hệ song tuyến.....  | 32        |
| 2.2.2. ĐKDB trên nền tối ưu hóa theo sai lệch tín hiệu điều khiển.....  | 36        |
| 2.3. Kết luận chương 2 .....  | 42        |
| <b>CHƯƠNG 3</b>   | <b>43</b> |
| <b>ĐỀ XUẤT MỘT PHƯƠNG PHÁP MỚI ĐỂ ĐIỀU KHIỂN DỰ BÁO HỆ PHI TUYẾN LIÊN TỤC TRÊN NỀN BIẾN PHÂN</b>  | <b>43</b> |
| 3.1. Nội dung cơ bản của phương pháp biến phân .....  | 44        |
| 3.1.1. Nguyên lý biến phân.....   | 45        |
| 3.1.2. Bộ điều khiển LQR (Linear Quadratic Regulator) .....   | 46        |
| 3.1.3. Điều kiện đủ cho tính ổn định của hệ LQR .....   | 46        |
| 3.1.4. Áp dụng nguyên tắc điều khiển LQR để điều khiển tối ưu hệ tuyến tính bám ổn định theo giá trị đầu ra cho trước.....                                | 47        |
| 3.2. Phương pháp đề xuất để điều khiển dự báo với cửa sổ dự báo vô hạn cho hệ song tuyến liên tục không dừng, bám theo được giá trị đầu ra cho trước..... | 49        |
| 3.2.1. Tư tưởng chính của phương pháp .....   | 49        |
| 3.2.2. Xây dựng thuật toán điều khiển .....   | 51        |
| 3.2.3. Khả năng xử lý điều kiện ràng buộc.....  | 53        |
| 3.2.4. Chứng minh tính bám ổn định của phương pháp được đề xuất .....   | 54        |
| 3.2.5. Khả năng áp dụng cho hệ phi tuyến affine không dừng.....   | 56        |

|   |            |
|---|------------|
| <b>CHƯƠNG 4</b>   | <b>58</b>  |
| <b>THỰC NGHIỆM KIỂM CHỨNG CHẤT LƯỢNG PHƯƠNG<br/>PHÁP ĐÃ ĐỀ XUẤT TRÊN ĐỐI TƯỢNG TRMS</b>                     | <b>58</b>  |
| 4.1. Mô hình toán của hệ TRMS.....  | 58         |
| 4.1.1. Mô tả vật lý hệ TRMS .....   | 58         |
| 4.1.2. Mô hình tựa Newton.....  | 59         |
| 4.2. Thiết kế bộ điều khiển dự báo trên nền quy hoạch phi tuyến .....                                       | 64         |
| 4.2.1. Thiết kế và cài đặt bộ điều khiển dự báo cho hệ TRMS .....   | 64         |
| 4.2.2. Mô phỏng trên MatLab.....  | 65         |
| 4.3. Thiết kế bộ điều khiển dự báo trên nền biến phân (phương pháp<br>điều khiển được luận án đề xuất)..... | 69         |
| 4.3.1. Thiết kế và cài đặt bộ điều khiển .....  | 69         |
| 4.3.2. Mô phỏng trên MatLab và so sánh, đánh giá chất lượng .....   | 70         |
| 4.4. Thí nghiệm trên mô hình vật lý của hệ TRMS.....  | 74         |
| 4.4.1. Cài đặt bộ quan sát Kalman .....   | 75         |
| 4.4.2. Các kết quả thực nghiệm.....   | 82         |
| 4.5. Kết luận chương 4 .....  | 90         |
| <b>DANH MỤC CÔNG TRÌNH ĐÃ CÔNG BỐ LIÊN QUAN ĐẾN ĐỀ<br/>TÀI.....</b>   | <b>92</b>  |
| <b>TÀI LIỆU THAM KHẢO</b>   | <b>93</b>  |
| Tiếng Việt .....  | 93         |
| Tiếng Anh .....   | 93         |
| <b>PHỤ LỤC</b> .....  | <b>102</b> |

## DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

### Các kí hiệu:

| Ký hiệu        | Diễn giải nội dung đầy đủ                                   |
|----------------|---|
| $N_p$          | Miền (phạm vi) dự báo                                       |
| $N_c$          | Miền (phạm vi) điều khiển                                   |
| $l_t(m)$       | Chiều dài của phần đuôi của cánh tay đòn ( $m$ )            |
| $l_m(m)$       | Chiều dài của phần chính của cánh tay đòn ( $m$ )           |
| $l_b(m)$       | Chiều dài cánh tay đòn đối trọng ( $m$ )                    |
| $l_{cb}(m)$    | Khoảng giữa cánh tay đòn đối trọng và khớp (bộ nối) ( $m$ ) |
| $r_{ms/ts}(m)$ | Bán kính của hộp bảo vệ cánh quạt chính/đuôi                |
| $m_{tr}(kg)$   | Khối lượng của động cơ một chiều đuôi ( $kg$ )              |
| $m_{mr}(kg)$   | Khối lượng của động cơ một chiều chính ( $kg$ )             |
| $m_{cb}(kg)$   | Khối lượng của đối trọng ( $kg$ )                           |
| $m_t(kg)$      | Khối lượng của phần đuôi của cánh tay đòn ( $kg$ )          |
| $m_m(kg)$      | Khối lượng phần chính của cánh tay đòn ( $kg$ )             |
| $m_b(kg)$      | Khối lượng của cánh tay đòn đối trọng ( $kg$ )              |
| $m_{ts}(kg)$   | Khối lượng của lưới chắn đuôi ( $kg$ )                      |
| $m_{ms}(kg)$   | Khối lượng của lưới chắn chính ( $kg$ )                     |
| $k_g$          | Hệ số con quay  |

|  |   |
|--|---|
| $R_{av/h}(\Omega)$   | Điện trở phản ứng của ĐCMC cánh quạt chính/đuôi ( $\Omega$ )                              |
| $L_{av/h}(mH)$   | Điện cảm phản ứng của ĐCMC cánh quạt chính/đuôi ( $H$ )                                   |
| $k_a \varphi(Nm/A)$  | Từ thông  |
| $J_{mr/tr}(gcm^2)$   | Mômen quán tính của ĐCMC chính/đuôi ( $kgm^2/s$ )   |
| $B_{mr/tr}(kgm^2/s)$   | Hệ số ma sát nhớt của ĐCMC chính và ĐCMC đuôi ( $kgm^2/s$ )                               |
| $F_{v/h}$  | Hàm phi tuyến của lực khí động học từ cánh quạt chính và cánh quạt đuôi ( $N$ )           |
| $g$  | Gia tốc trọng trường ( $m/s^2$ )  |
| $J_v$  | Mômen quán tính của trục ngang (trục hoành) ( $kgm^2$ )                                   |
| $M_{fric,v} / M_{fric,h}$  | Mômen của lực ma sát trong mặt phẳng thẳng đứng/ mặt phẳng ngang                          |
| $k_{ah/v}, k_{flp}, k_{fln}, k_{fvp}, k_{fvn}, k_{th/v}, k_v, k_m$ | Các hệ số dương ( $Nm/AWb$ )  |
| $\omega_{v/h}$   | Vận tốc góc của cánh quạt chính và cánh quạt đuôi ( $rad/s$ )                             |
| $\Omega_{h/v}$   | Vận tốc góc của cánh tay đòn TRMS trong mặt phẳng ngang/ mặt phẳng thẳng đứng ( $rad/s$ ) |
| $U_{v/h}$  | Điện áp ĐCMC cánh quạt chính/đuôi ( $V$ )   |
| $E_{av/h}$   | Sức điện động của ĐCMC cánh quạt chính/đuôi ( $V$ )                                       |



|                      |   |
|----------------------|---|
| $i_{av/h}$           | Dòng điện phân ứng của ĐCMC cánh quạt chính/đuôi ( $A$ )  |
| $\Phi_{v/h}$         | Từ thông của ĐCMC cánh quạt chính/đuôi ( $Wb$ )   |
| $M_{ev/h}$           | Mômen điện từ của ĐCMC cánh quạt chính/đuôi ( $Nm$ )  |
| $M_{tv/h}$           | Mômen tải của ĐCMC cánh quạt chính/đuôi ( $Nm$ )  |
| $\gamma_m, \gamma_t$ | Các hệ số biến dạng của chiều dài cánh tay đòn chính và đuôi  |
| $S_v$                | Vận tốc góc của cánh tay đòn TRMS trong mặt phẳng thẳng đứng mà không bị ảnh hưởng bởi cánh quạt đuôi ( $rad/s$ ) |
| $S_h$                | Vận tốc góc của cánh tay đòn TRMS trong mặt phẳng ngang mà không bị ảnh hưởng bởi cánh quạt chính ( $rad/s$ )     |
| $\hat{y}(k+i k)$     | Đầu ra dự báo ở thời điểm thứ $k+i$ so với thời điểm thứ $k$  |
| $u(k+i k)$           | Tín hiệu điều khiển ở thời điểm thứ $k+i$ so với thời điểm thứ $k$  |
| $y_{ref}$            | Tín hiệu đặt hoặc đầu ra quá trình  |
| $x_k$                | Vector của $n$ giá trị trạng thái của hệ tính tại thời điểm $t = kT$  |
| $u_k$                | Vector của $m \leq n$ giá trị tín hiệu điều khiển (tín hiệu đầu vào)  |
| $y_k$                | Vector của $r \leq m$ giá trị tín hiệu đáp ứng (tín hiệu đầu ra)  |

|                       |  |
|-----------------------|--|
| $e_{k+i}$             | Sai lệch   |
| $T$                   | Chu kỳ trích mẫu tín hiệu  |
| $J(\mathcal{U})$      | Hàm mục tiêu   |
| $\mathcal{U}^*$       | Nghiệm của bài toán tối ưu   |
| $q_i$                 | Trọng số sai lệch  |
| $r_j$                 | Trọng số điều khiển  |
| $Q$                   | Ma trận trọng số sai lệch  |
| $R$                   | Ma trận trọng số điều khiển  |
| $\Delta \mathbf{u}_k$ | Sai lệch tín hiệu điều khiển   |
| $\Theta$              | Ma trận có tất cả các phần tử đều bằng 0                                   |
| $I$                   | Ma trận đơn vị   |
| $\delta$              | Sai lệch giữa tham số trạng thái hiện thời và tham số trạng thái xác lập   |
| $\rho$                | Sai lệch giữa tín hiệu điều khiển hiện thời và tín hiệu điều khiển xác lập |
| $s(\mathcal{U})$      | Hàm phạt   |

**Các chữ viết tắt:**

|       |  |
|-------|--|
| ANFIS | Adaptive Neural Fuzzy Inference System |
| BB    | Branch and Bound                       |
| BFO   | Bacterial Foraging Optimization        |
| ĐCMC  | Động cơ một chiều                      |
| ĐKDB  | Điều khiển dự báo                      |
| DMC   | Dynamical Matrix Control               |
| EKF   | Extended Kalman Filter                 |
| FSMC  | Fuzzy Sliding Mode Control             |
| GA    | Genetic Algorithm                      |
| GPC   | Generalized Predictive Control         |
| IIO   | Increment Input Output models          |
| IO    | Direct Input Output models             |
| IOM   | Input Output Models                    |
| LP    | Linear programming                     |
| LQG   | Linear Quadratic Gaussian              |
| LQR   | Linear Quadratic Regulator             |
| LRPC  | Long-Range Predictive Control          |
| LTI   | Linear time - invariant                |

|         |                                    |
|---------|------------------------------------|
| MIMO    | Multiple Input Multiple Output     |
| MPC     | Model Prediction Control           |
| MPCS    | Thuật toán MPC                     |
| NMPC    | Nonlinear Model Prediction Control |
| NNs     | Neural Networks                    |
| PID AFC | PID Active force control           |
| QP      | Quadratic Programing               |
| RHC     | Receding horizon control           |
| SISO    | Single Input Single Output         |
| SQP     | Sequential Quadratic Programing    |
| TRMS    | Twin rotor MIMO system             |
| UKF     | Unscented Kalman Filter            |

## DANH MỤC CÁC HÌNH ẢNH, ĐỒ THỊ

|  |    |
|--|----|
| Hình 2.1. Cấu trúc cơ bản của một hệ thống điều khiển dự báo   | 37 |
| Hình 2.2. Sơ đồ khối của MPC để điều khiển hệ song tuyến   | 44 |
| Hình 3.1: Hệ kín với bộ điều khiển phản hồi trạng thái tối ưu LQR  | 60 |
| Hình 3.2: Mô tả tư tưởng của phương pháp   | 63 |
| Hình 3.3. Điều khiển dự báo hệ phi tuyến liên tục với cửa sổ dự báo vô hạn                                   | 68 |
| Hình 4.1. Cấu hình vật lý của hệ TRMS  | 72 |
| Hình 4.2. Cấu trúc bộ ĐKDB áp dụng cho thuật toán SQP  | 79 |
| Hình 4.3. Đáp ứng của góc chao dọc khi tín hiệu đặt là xung vuông  | 79 |
| Hình 4.4. Đáp ứng của góc đảo lái khi tín hiệu đặt là xung vuông   | 80 |
| Hình 4.5. Đáp ứng của góc chao dọc khi tín hiệu đặt là substep   | 80 |
| Hình 4.6. Đáp ứng của góc đảo lái khi tín hiệu đặt là substep  | 81 |
| Hình 4.7. Sơ đồ cấu trúc bộ ĐKDB phản hồi trạng thái để tín hiệu ra bám theo tín hiệu đầu ra mẫu cho hệ TRMS | 83 |
| Hình 4.8. Đáp ứng đầu ra góc đảo lái khi tín hiệu đặt là xung vuông  | 84 |
| Hình 4.9. Đáp ứng đầu ra góc chao dọc khi tín hiệu đặt là xung vuông   | 84 |
| Hình 4.10. Đáp ứng đầu ra góc đảo lái khi tín hiệu đặt là substep  | 84 |
| Hình 4.11. Đáp ứng đầu ra góc chao dọc khi tín hiệu đặt là substep   | 85 |
| Hình 4.12. Lưu đồ của phương pháp quan sát Kalman mở rộng  | 89 |
| Hình 4.13. Sơ đồ mô phỏng kiểm tra bộ quan sát trạng thái  | 92 |

|  |     |
|--|-----|
| Hình 4.14. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ nhất ( $\Omega_h$ )   | 92  |
| Hình 4.15. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ hai ( $S_h$ )         | 93  |
| Hình 4.16. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ ba ( $\alpha_h$ )     | 93  |
| Hình 4.17. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ tư ( $\Omega_v$ )     | 94  |
| Hình 4.18. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ tư năm ( $S_v$ )      | 94  |
| Hình 4.19. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ tư sáu ( $\alpha_v$ ) | 95  |
| Hình 4.20. Hình ảnh thí nghiệm điều khiển hệ thống TRMS  | 96  |
| Hình 4.21. Bộ điều khiển dSPACE1103  | 98  |
| Hình 4.22. Phần mềm giám sát và điều khiển ControlDesk   | 99  |
| Hình 4.23. Đáp ứng đầu ra của góc chao dọc khi sử dụng bộ điều khiển dự báo tối ưu hóa trên nền qui hoạch phi tuyến              | 101 |
| Hình 4.24. Đáp ứng đầu ra của góc đảo lái khi sử dụng bộ điều khiển dự báo tối ưu hóa trên nền qui hoạch phi tuyến               | 101 |
| Hình 4.25. Đáp ứng đầu ra của góc chao dọc khi sử dụng bộ điều khiển dự báo bám ổn định theo tín hiệu mẫu ở đầu ra               | 102 |
| Hình 4.26. Đáp ứng đầu ra của góc đảo lái khi sử dụng bộ điều khiển dự báo bám ổn định theo tín hiệu mẫu ở đầu ra                | 102 |

## MỞ ĐẦU

### 1. Giới thiệu

Điều khiển dự báo dựa trên mô hình (Model Predictive Control - MPC), hay còn thường được gọi ngắn gọn là điều khiển dự báo, ra đời vào cuối thập niên 70 và đầu thập niên 80 của thế kỉ trước, là một xu hướng điều khiển được ưa chuộng. Trong hai thập kỷ trở lại đây, điều khiển dự báo đã có những bước phát triển rất đáng kể, đóng góp khá nhiều các phương pháp về mặt học thuật cũng như đẩy mạnh khả năng ứng dụng của MPC trong thực tế, điều đó được thể hiện trong các tài liệu [13], [14], [15], [27], [68] với hơn 3000 ứng dụng vào điều khiển quá trình, điều khiển các hệ cơ, điều khiển robot, điều khiển các hệ bay... Bản chất của điều khiển dự báo là sử dụng mô hình tường minh của đối tượng để tính toán tối ưu các biến được điều khiển thông qua các phương pháp tối ưu hóa. Để thiết kế, cài đặt bộ điều khiển dự báo cho một đối tượng cụ thể, cần thực hiện 3 công việc chính đó là:

- Xây dựng mô hình dự báo;
- Xác định hàm mục tiêu và các điều kiện ràng buộc;
- Giải bài toán tối ưu.

Đối với hệ tuyến tính việc thực hiện các công việc này khá dễ dàng do đã có những nghiên cứu tương đối hoàn chỉnh. Song, đối với hệ phi tuyến, việc thực hiện các công việc này còn gặp nhiều khó khăn, đặc biệt là việc xây dựng mô hình dự báo phi tuyến và tìm thuật toán nghiệm toàn cục của bài toán quy hoạch phi tuyến. Vì vậy, xu hướng nghiên cứu hiện nay của MPC là khai thác, áp dụng bộ điều khiển này để điều khiển các đối tượng thực tế có tính phi tuyến mạnh, thời gian đáp ứng nhanh, các yêu cầu về điều kiện bị chặn nghiêm ngặt... mà các bộ điều khiển dự báo tuyến tính truyền thống như GPC, DMC... khó thực hiện được.

## 2. Tính cấp thiết của luận án

Điều khiển dự báo dựa trên mô hình cho hệ tuyến tính đã được phát triển, chấp nhận và ứng dụng cho các ngành công nghiệp quá trình và một số lĩnh vực khác. Tuy nhiên đối với quá trình phi tuyến (đối tượng điều khiển phi tuyến) đặc biệt là vừa phi tuyến và vừa có nhiễu thì các phương pháp MPC áp dụng cho hệ tuyến tính hoàn toàn không sử dụng được. Có hai vấn đề khó khăn chính đối với điều khiển dự báo dựa trên mô hình phi tuyến đó là:

- Nhận dạng đối tượng điều khiển hay xây dựng mô hình dự báo sao cho đạt được mức độ chính xác cao nhất để xác định được giá trị đầu ra tương lai và giá trị đó ít bị ảnh hưởng bởi nhiễu quá trình, và
- Giải một bài toán tối ưu phi tuyến với rất nhiều ràng buộc.

Bài toán tối ưu hóa với các điều kiện ràng buộc đôi khi không tìm được lời giải, những trường hợp như vậy thuật toán điều khiển trở nên không khả thi. Trong trường hợp này cần loại bớt hoặc giới hạn các điều kiện ràng buộc để thuật toán có thể tìm được lời giải phù hợp, tạo ra tính khả thi cho bài toán tối ưu. Ở MPC, việc làm này thường được gọi là tạo ra tính khả thi (*feasibility*) cho bài toán.

Tóm lại, vì không có một giải pháp mang tính tổng quát cho nên trong các nghiên cứu thường dùng các phương pháp quy hoạch phi tuyến phổ biến như SQP (Sequential Quadratic programming), giải thuật di truyền (Genetic Algorithms - GA), v.v. Do vậy khối lượng tính toán của NMPC sử dụng phương pháp số cũng nặng hơn nhiều so với MPC tuyến tính, và thậm chí nó còn tăng theo cấp số nhân khi ta tăng tầm dự báo.

Nếu sử dụng mô hình dự báo phi tuyến thì với bài toán nhận dạng cho hệ phi tuyến, đặc biệt là cho hệ phi tuyến có tham số bất định sẽ gặp rất nhiều khó khăn, thậm chí khi giả thiết chúng ta có mô hình dự báo phi tuyến cho đối tượng thì khi thực hiện giải bài toán tối ưu để tìm tín hiệu điều khiển dự báo



lại gặp khó khăn hơn nữa vì lại tiếp tục phải đụng chạm đến vấn đề giải bài toán tối ưu phi tuyến với nhiều ràng buộc và hạn chế, do đó cần phải trả lời cho các câu hỏi sau đây:

- Bài toán tối ưu phi tuyến đó có giải được không? Hiện nay cũng chưa có phương pháp giải bài toán tối ưu phi tuyến tổng quát, hiện tại mới chỉ có phương pháp điều khiển tối ưu là quy hoạch động Bellman, nguyên lý cực đại của Pontragin, phương pháp biến phân.
- Tầm dự báo của bộ điều khiển dự báo là bao nhiêu để hệ thống kín còn đảm bảo tính ổn định?
- Tính ổn định của hệ kín ra sao khi cửa sổ dự báo tiến đến vô cùng?
- Hệ thống kín có đảm bảo về thời gian tính toán để thỏa mãn tính thời gian thực trong điều khiển công nghiệp?

Từ các phân tích ở trên, ta thấy rằng đối với điều khiển dự báo hệ phi tuyến nói chung còn rất nhiều vấn đề cần được tiếp tục nghiên cứu, hoàn thiện. Một số trong các vấn đề đó là:

- Xây dựng mô hình dự báo phản ánh trung thực đối tượng phi tuyến. Khi mô hình dự báo càng gần với mô hình đối tượng thì kết quả dự báo càng sát và chất lượng bộ điều khiển càng cao. Điều này rất dễ thực hiện đối với đối tượng tuyến tính, nhưng đối với hệ phi tuyến vẫn còn đang là bài toán mở;

- Chọn phiếm hàm mục tiêu phù hợp cho từng đối tượng, đặc biệt khi các mục tiêu đối nghịch nhau cần phải có giải pháp "thỏa hiệp" giữa các mục tiêu để chọn được phiếm hàm mục tiêu phù hợp nhất;

- Tìm ra các phương pháp mới giải bài toán tối ưu phi tuyến và cài đặt chúng vào bộ điều khiển dự báo.

### 3. Mục tiêu của luận án

Mục tiêu của luận án là nghiên cứu đề xuất thuật toán mới giải bài toán tối ưu trong hệ thống điều khiển dự báo phi tuyến MIMO.

#### Mục tiêu cụ thể của luận án là:

- Nghiên cứu phương pháp luận nhằm xây dựng bộ điều khiển dự báo cho hệ phi tuyến (nói chung) và hệ song tuyến (nói riêng).

- Đề xuất thuật toán mới giải bài toán tối ưu trong hệ MPC phi tuyến. Trong đó: khối tối ưu hóa xây dựng trên nền quy hoạch phi tuyến được áp dụng cho mô hình không liên tục của đối tượng. Đề xuất một khối tối ưu hóa áp dụng phương pháp biến phân để áp dụng cho mô hình liên tục. Cả hai khối tối ưu hóa này được mở rộng sang điều khiển tối ưu bám các quỹ đạo đặt cho trước, chứ không đơn thuần là điều khiển ổn định. Đưa ra các thuật toán điều khiển cho một lớp các đối tượng phi tuyến.

- Khảo sát hệ thống TRMS và cài đặt thuật toán điều khiển dự báo trên đây vào đối tượng cụ thể là hệ TRMS và mô phỏng kiểm chứng.

### 4. Đối tượng, phạm vi và phương pháp nghiên cứu

- **Đối tượng nghiên cứu:** Hệ thống điều khiển dự báo theo mô hình phi tuyến, các thuật toán giải bài toán tối ưu trong điều khiển dự báo phi tuyến; hệ thống Twin Rotor MIMO System (TRMS).

- **Phạm vi nghiên cứu:**

- + Nghiên cứu, thiết kế bộ điều khiển dự báo phản hồi trạng thái hệ phi tuyến bám theo tín hiệu đầu ra mẫu với cửa sổ dự báo hữu hạn sử dụng thuật toán SQP để giải bài toán tối ưu.

- + Nghiên cứu, thiết kế bộ điều khiển dự báo phản hồi trạng thái để tín hiệu ra bám theo tín hiệu đầu ra mẫu cho hệ phi tuyến liên tục với cửa sổ dự báo vô hạn sử dụng phương pháp biến phân (ĐK tối ưu) để giải bài toán tối ưu.

+ Mô phỏng và thực nghiệm kiểm chứng kết quả nghiên cứu lý thuyết trên đối tượng TRMS (chưa kể đến tác động của nhiễu và xử lý tác động xen kênh).

**- Phương pháp nghiên cứu:**

+ Nghiên cứu lý thuyết: Phân tích, đánh giá các nghiên cứu đã được công bố trên các bài báo, tạp chí, các tài liệu tham khảo về điều khiển dự báo cho hệ phi tuyến; Các phương pháp giải bài toán tối ưu trong điều khiển dự báo. Nghiên cứu, thiết kế bộ điều khiển dự báo phản hồi trạng thái bám tín hiệu đầu ra mẫu cho hệ phi tuyến không liên tục và liên tục khi cửa sổ dự báo hữu hạn và vô hạn.

+ Mô phỏng trên Matlab - Simulink để kiểm chứng lại lý thuyết.

+ Thực nghiệm trên hệ thống phi tuyến để kiểm chứng kết quả nghiên cứu lý thuyết.

## **5. Ý nghĩa khoa học và thực tiễn**

### **5.1. Ý nghĩa khoa học**

Luận án đưa ra phương pháp luận và đề xuất 1 thuật toán mới trong chiến lược tối ưu hóa bộ điều khiển dự báo hệ phi tuyến MIMO, góp phần bổ sung, làm phong phú thêm khối kiến thức về điều khiển hệ phi tuyến.

### **5.2. Ý nghĩa thực tiễn**

- Thuật toán mới đề xuất đã được kiểm nghiệm qua mô phỏng và thực nghiệm trên hệ thống thực, qua đó khẳng định tính khả thi của thuật toán mà luận án đề xuất.

- Kết quả nghiên cứu của luận án đã giảm được thời gian tính toán khi giải bài toán tối ưu trong chiến lược tối ưu hóa của điều khiển dự báo đã khẳng định tính khả thi của bộ điều khiển sử dụng trong các hệ thống công nghiệp.

- Kết quả nghiên cứu của luận án sẽ là tài liệu tham khảo cho sinh viên ngành điều khiển và tự động hóa, học viên cao học và các nghiên cứu sinh quan tâm nghiên cứu về thiết kế bộ điều khiển dự báo cho hệ phi tuyến. Có khả năng bổ sung phần cài đặt thuật toán về bộ điều khiển dự báo cho hệ phi tuyến khi cửa sổ dự báo tiến ra vô cùng trong toolbox của Matlab – Simulink.

## **6. Bố cục luận án**

Ngoài phần mở đầu và kết luận, nội dung chính của luận án được trình bày trong 4 chương:

Chương 1: Tổng quan về các phương pháp điều khiển dự báo cho hệ phi tuyến. Nội dung chương này tổng hợp các nghiên cứu về điều khiển dự báo cho hệ phi tuyến. Trước tiên, chỉ ra các tác giả đã xây dựng mô hình dự báo cho hệ phi tuyến dựa trên các phương pháp khác nhau, nhận xét đánh giá kết quả của các mô hình. Tiếp theo, tập trung chủ yếu vào những công trình đã công bố về các phương pháp điều khiển dự báo cho hệ phi tuyến. Sau đó, tác giả đề cập đến các phương pháp giải bài toán tối ưu cho cả hệ phi tuyến không bị ràng buộc và hệ phi tuyến bị ràng buộc. Phân tích, nhận định và rút ra ý nghĩa về lý luận cũng như thực tiễn của các công trình đó. Cuối cùng, đưa ra các vấn đề cần tiếp tục nghiên cứu và đề xuất hướng nghiên cứu của luận án căn cứ vào những vấn đề còn mở chưa được khai thác trong các công trình đó.

Chương 2: Trong nội dung chương này tác giả đã đưa ra cấu trúc và nguyên lý làm việc của điều khiển dự báo phi tuyến, hoàn thiện các bước giải bài toán tối ưu trong điều khiển dự báo hệ phi tuyến trên nền quy hoạch phi tuyến thông qua thuật toán 2.1.

Chương 3: Xây dựng phương pháp luận về thiết kế bộ điều khiển dự báo cho hệ phi tuyến. Tác giả đề xuất một phương pháp điều khiển dự báo mới cho hệ phi tuyến liên tục khi cửa sổ dự báo tiến ra vô hạn dựa trên nền

phương pháp biến phân (điều khiển tối ưu) đảm bảo cho hệ bám ổn định toàn cục và cải thiện đáng kể thời gian tính toán thể hiện trong thuật toán 3.1.

Chương 4: Trên cơ sở lý luận đã đề xuất ở chương 2 và chương 3, để kiểm chứng các kết quả nghiên cứu lý thuyết tác giả đã thiết kế điều khiển dự báo cho hệ TRMS thông qua mô phỏng. Trước tiên, luận án đã cài đặt bộ điều khiển dự báo cho đối tượng TRMS sử dụng phương pháp SQP giải bài toán tối ưu nhằm thu được các kết quả cần thiết để so sánh với việc cài đặt bộ điều khiển dự báo cho đối tượng này khi sử dụng phương pháp biến phân là những đề xuất mới của luận án. Tiếp đó, chứng minh khả năng ứng dụng của lý thuyết đã đề xuất trong luận án tác giả tiến hành kết nối và điều khiển đối tượng TRMS thực thông qua card chuyển đổi **DSP1103** tại phòng thí nghiệm Điện - Điện tử trường Đại học Kỹ thuật Công nghiệp Đại học Thái Nguyên. Thông qua việc sử dụng bộ lọc Kalman mở rộng để quan sát các tham số trạng thái của hệ thống, luận án đưa ra ở chương này các kết quả thực nghiệm trên đối tượng TRMS thực.

## CHƯƠNG 1

### TỔNG QUAN VỀ ĐIỀU KHIỂN DỰ BÁO CHO HỆ PHI TUYẾN

*Đặt vấn đề:* Điều khiển dự báo dựa trên mô hình phi tuyến (Nonlinear Model Predictive Control - NMPC) trong thời gian qua đã thu hút những nghiên cứu của nhiều tác giả trong và ngoài nước [8], [13], [14], [27], [29], [30], [35], [36], [37], [39], [46], [47], [56], [61], [63], [64], [65], [68], [71], [72]. Ngày nay các nghiên cứu về NMPC tập trung chính vào tính ổn định, tính bền vững trong khi các vấn đề về thời gian tính toán lại chưa được quan tâm đúng mức.

Để phân tích tính toán bài toán điều khiển dự báo cho hệ phi tuyến (NMPC), việc kết hợp trực tiếp một quá trình phi tuyến và một cấu trúc của MPC tuyến tính sẽ dẫn đến bài toán tối ưu quy hoạch phi tuyến không lồi (non-convex) mà bài toán này để giải được cần phải có những ràng buộc về mặt thời gian lấy mẫu rất chặt chẽ. Trong MPC, thuật toán tối ưu hóa được áp dụng để tính toán chuỗi tín hiệu điều khiển tương lai trong phạm vi điều khiển sao cho cực tiểu hóa được hàm mục tiêu điều khiển theo các ràng buộc đã cho. Đối với bài toán tối ưu hóa tuyến tính là bài toán tối ưu hóa lồi, thời gian hội tụ của thuật toán nhanh và luôn tìm được lời giải tối ưu toàn cục. Một mô hình tuyến tính với ràng buộc tuyến tính và hàm mục tiêu chuẩn bậc hai thì việc tìm lời giải cho bài toán tối ưu hóa thường dùng là thuật toán QP (Quadratic programming), nếu hàm mục tiêu là chuẩn bậc 1 hoặc chuẩn vô cùng thì thuật toán được dùng là LP (Linear programming). Khi đối tượng điều khiển là phi tuyến, bài toán tối ưu hóa là bài toán không lồi, việc đi tìm lời giải rất dễ rơi vào trường hợp tối ưu cục bộ, do đó việc lựa chọn thuật toán điều khiển rất quan trọng. Với hệ thống phi tuyến, thuật toán quy hoạch phi tuyến thường dùng là các phương pháp phân nhánh và giới hạn (Branch and Bound), hay SQP (Sequential Quadratic Programming), trust region, hoặc các phương pháp Newton, phương pháp Levenberg - Marquardt, khi không có

điều kiện ràng buộc. Tuy nhiên trong số tất cả những phương pháp quy hoạch phi tuyến kể trên, không có một phương pháp nào có thể đảm bảo chắc chắn là nghiệm tối ưu tìm được sẽ là nghiệm toàn cục, nếu như bài toán tối ưu đó không lồi. Bởi vậy xu hướng tiếp theo để có thể đảm bảo chắc chắn cho tính toàn cục của nghiệm tìm được là sử dụng các phương pháp của điều khiển tối ưu, chẳng hạn như nguyên lý cực đại của Pontragin, phương pháp quy hoạch động của Bellman và phương pháp biến phân. Song cho tới nay kết quả nghiên cứu theo xu hướng này là chưa nhiều, nhất là khi có thêm điều kiện ràng buộc.

### **1.1. Tổng quan các công trình nghiên cứu về điều khiển dự báo hệ phi tuyến trên thế giới**

Điều khiển dự báo dựa trên mô hình (Model Predictive Control - MPC) ra đời từ cuối thập niên 70 và đầu thập niên 80 của thế kỉ trước và từ đó có những phát triển đáng kể. Đây là một cách tiếp cận điều khiển mà sử dụng mô hình tường minh của đối tượng để tính toán tối ưu các biến điều khiển thông qua việc sử dụng các phương pháp tối ưu hóa (*Optimization*). MPC tuyến tính và điều khiển ma trận động (DMC) đã rất phổ biến trong hai thập niên này. Mặc dù hầu hết các quá trình thực tế đều phi tuyến, song đa số các kỹ thuật MPC được ứng dụng trong các quá trình công nghiệp đều là các mô hình tuyến tính vì một trong các lí do sau:

- Các mô hình tuyến tính triển khai nhanh và dễ dàng so với mô hình phi tuyến.
- Tính ổn định và bền vững đang còn là thử thách đối với các mô hình phi tuyến.

Một số các mô hình phi tuyến và điều kiện ràng buộc đòi hỏi phải giải bài toán tối ưu phi tuyến không lồi nên việc giải rất phức tạp.

Trong nhiều trường hợp mô hình tuyến tính cũng đưa ra các kết quả thỏa đáng (đủ dùng trong công nghiệp).

Khoảng hơn hai thập niên trước đây, điều khiển dự báo chủ yếu được khai thác và ứng dụng vào điều khiển các đối tượng tuyến tính không có ràng buộc hoặc ràng buộc tuyến tính. Những năm gần đây, điều khiển dự báo (MPC) là một trong các kỹ thuật tính toán điều khiển tối ưu hiện đại, đang phát triển mạnh cả lĩnh vực lý thuyết và ứng dụng, và đã có được vị trí quan trọng trong lĩnh vực điều khiển nói chung và trong điều khiển các quá trình công nghiệp nói riêng là do MPC có những ưu điểm nổi bật như [9], [33], [51]:

- Phù hợp với một lớp rất rộng các bài toán điều khiển, từ quá trình có hằng số thời gian lớn và thời gian trễ lớn, đến những hệ phi tuyến biến đổi nhanh,
- Áp dụng được cho các quá trình có số lượng lớn các biến điều khiển và biến được điều khiển,
- Dễ dàng đáp ứng được các bài toán điều khiển có ràng buộc cả về trạng thái và tín hiệu điều khiển,
- Các đối tượng điều khiển thay đổi và sự cố thiết bị,
- Là bài toán dựa trên nền tối ưu nên có khả năng nâng cao tính bền vững của hệ thống đối với sai lệch mô hình và nhiễu.

Ban đầu MPC chủ yếu chỉ được áp dụng cho hệ tuyến tính, biến đổi chậm. Song do sự phát triển mạnh mẽ của công nghệ thông tin đã cải thiện đáng kể tốc độ tính toán của máy tính nên đã khắc phục hạn chế về khối lượng tính toán lớn, và từ đó mở rộng phạm vi ứng dụng cho cả những đối tượng biến đổi nhanh, phi tuyến, có khối lượng tính toán rất lớn.

Điều khiển dự báo dựa trên mô hình là sự kết hợp của một số lĩnh vực đã được phát triển trong lĩnh vực điều khiển, điển hình đó là hai lĩnh vực điều



khuyến tối ưu và nhận dạng hệ thống. Ngay như tên của nó “điều khiển dự báo dựa trên mô hình” có nghĩa là trong đó cần phải sử dụng một mô hình dự báo để ước lượng (dự báo) các giá trị đầu ra trong tương lai để phục vụ cho bài toán điều khiển. Điều khiển dự báo dựa trên mô hình có thể thỏa mãn được các điều kiện ràng buộc về mặt vật lý của quá trình (như độ mở van, các hạn chế của cơ cấu chấp hành, các giới hạn của tín hiệu điều khiển v.v) trong thiết kế bộ điều khiển và chuyển bài toán thiết kế bộ điều khiển thành một bài toán tối ưu. Theo thống kê của tác giả trong [49] đã có hơn 3000 ứng dụng của MPC đã được thương mại hóa trong các lĩnh vực khác nhau bao gồm: công nghệ lọc hóa dầu, công nghệ chế biến thực phẩm, công nghệ ô tô, công nghệ không gian, công nghệ bột giấy và giấy v.v [48].

Với những lí do trên đây việc ứng dụng các kỹ thuật điều khiển dự báo dựa trên mô hình phi tuyến trong thực tế vẫn đang còn nhiều hạn chế. Vì vậy tiềm năng nghiên cứu và ứng dụng của kỹ thuật này thực sự rất lớn [17].

Hầu hết các đối tượng cần điều khiển trong thực tế đều phi tuyến, để điều khiển đối tượng phi tuyến, việc trước tiên phải xây dựng mô hình, các mô hình phi tuyến cần phải tiến hành mô hình hóa bằng cách sử dụng giải tích xấp xỉ [13], [14], [27], [29], [37], [56], [63], [68], [72] hoặc trí tuệ nhân tạo dựa trên kinh nghiệm [11], [31], [62] như mạng *neuron* và *wavelet*. Mỗi một lớp mô hình có những ưu, nhược điểm riêng. Nhiều trường hợp các mô hình phi tuyến có thể được biểu diễn trọn vẹn khi sử dụng mô hình tuyến tính nhiều biến [13], [18], [29], [56] hoặc mô hình tuyến tính thích nghi [29], [37], [63], [72]. Để mô hình hóa một hệ thống sử dụng kỹ thuật mô hình hóa nhiều biến (đa mô hình) thì miền điều khiển của hệ thống tuyến tính phải được tách ra thành một số miền con và thực hiện tuyến tính hóa trong mỗi miền con đó. Vì vậy, dựa trên điểm làm việc hiện tại của hệ thống, mô hình tuyến tính được sử dụng để dự báo đầu ra của quá trình. Ví dụ: Aufderheide (2001) đã đề xuất mô hình điều khiển dự báo với thời gian trễ chủ yếu dựa trên chiến lược mô

hình nhiều đầu vào cho một lớp các mô hình bậc nhất có trễ [13], [14]. Một chiến lược điều khiển thích nghi đa mô hình đối với điều khiển ma trận động học nhiều biến đã được đề cập trong [27]. Tác giả này so sánh các đầu ra của bộ điều khiển DMC tuyến tính nhiều vào ứng với mô hình đáp ứng bước của chúng. Wan và Kothare (2004) đã đề cập đến sơ đồ điều khiển dự báo phản hồi đầu ra ổn định cho các hệ thống phi tuyến bị ràng buộc [68]. Các tác giả này đã đưa ra một tập hợp các bộ điều khiển dự báo phản hồi đầu ra địa phương với các miền ước lượng của chúng ổn định trong miền điều khiển mong muốn và đã cài đặt trực tuyến thành sơ đồ MPC đơn giản. Một phương pháp xấp xỉ thích nghi mô hình nhiều đầu vào đã được đưa ra trong khung chương trình điều khiển dự báo theo mô hình để tính toán sự biến đổi và điều khiển các ràng buộc [56]. Trong trường hợp các hệ thống bậc cao với nhiều đầu vào thì rất khó tìm ra mô hình tuyến tính trên toàn miền điều khiển. Vì vậy, đối với các hệ thống phi tuyến mạnh và cửa sổ dự báo rộng thì việc tuyến tính hóa mô hình MPC nhiều đầu vào sử dụng một lớp mô hình tuyến tính trong suốt phạm vi dự báo vẫn không thể đảm bảo.

Mặt khác, phương pháp mô hình hóa tuyến tính thích nghi để cập nhật mô hình tuyến tính theo dữ liệu đo được hoặc tuyến tính hóa mô hình đối tượng phi tuyến cũng được một số tác giả đề cập đến. Ví dụ điều khiển dự báo theo mô hình tuyến tính thích nghi được đưa ra để cập nhật trực tuyến mô hình tuyến tính dựa trên dữ liệu đo được để điều khiển đối tượng bất định [29], [37]. Zhang và Zhang (2006) đề xuất một phương pháp, sử dụng giả định từng phần bắt nguồn từ động học tuyến tính một hệ phi tuyến tại mỗi bước điều khiển hàm dự báo để tạo thuận lợi cho phương pháp tối ưu bình phương cực tiểu tuyến tính [72]. Trong [63], tác giả so sánh các tham số ước lượng và thuật toán MPC thông qua sử dụng các hệ thống rời rạc hóa tuyến tính. Li và Biegler (1989) đã đề xuất chiến lược điều khiển Newton nhiều bước đối với các quá trình phi tuyến bị ràng buộc trong đó mô hình phi tuyến

được tuyến tính hóa xung quanh điểm làm việc và giải bài toán tối ưu toàn phương trên cửa sổ dự báo [38]. Họ đã mở rộng phương pháp đề xuất trong khoảng làm việc ưu tiên [38] để đặt chỉ số đặc tính vào bên trong dạng số gia điều khiển và thực hiện một số thay đổi như mở rộng cửa sổ dự báo đến  $\infty$  [22].

Trong [16], tác giả đã chỉ ra với hàm mục tiêu có cửa sổ dự báo hữu hạn thì sự ổn định vòng kín của MPC nói chung không được đảm bảo. Vì vậy, các nghiên cứu tiếp theo cần phải thực hiện là hàm mục tiêu có cửa sổ dự báo tiến ra vô cùng hoặc thỏa mãn các điều kiện ràng buộc. Trong điều khiển dự báo thì phương pháp ràng buộc điểm cuối đã được đề xuất rộng rãi để xét ổn định cho cả các hệ thống MPC tuyến tính và phi tuyến, cả hệ rời rạc và hệ liên tục. Bằng việc bổ sung thêm hàm phạt vào hàm mục tiêu, Bitmead (1990) đã đưa ra điều khiển dự báo ổn định cho các hệ thống tuyến tính không bị ràng buộc [16].

Rawlings và Muske (1993) đã đưa ra bộ điều khiển có phạm vi dự báo vô hạn để xét tính ổn định của các đối tượng tuyến tính ổn định và bất định [57]. Một sơ đồ mà cửa sổ dự báo gần như vô hạn được đưa ra để đảm bảo ổn định bộ điều khiển dự báo phi tuyến bất định với các ràng buộc đầu vào. Scokaert (1999) đã nghiên cứu tỉ mỉ các điều kiện dưới để các bộ điều khiển dự báo theo mô hình cận tối ưu được ổn định [58]. Các đặc tính bền vững của bộ điều khiển dịch dần về tương lai phi tuyến với các ràng buộc điểm cuối đã được nghiên cứu tỉ mỉ với mong muốn khuếch đại và bổ sung thêm nhiều. Michalska và Mayne (1993) đã đưa ra bộ điều khiển dịch dần về tương lai chế độ kép cho một lớp rộng các hệ thống phi tuyến với các ràng buộc điều khiển, ràng buộc trạng thái và sai số mô hình [44]. Một nghiên cứu đầy đủ về cả MPC tuyến tính và MPC phi tuyến khi tập trung vào các điều kiện đủ để đảm bảo ổn định và bền vững [43]. Bài báo xét đến cả sự ổn định và tính bền vững của kỹ thuật MPC phi tuyến có thể thấy trong [20]. Jadbabaie (2001) đưa ra MPC phi tuyến ổn định sử dụng điều khiển tối ưu với cửa sổ dự báo hữu hạn

không bị ràng buộc [33]. Phương pháp xấp xỉ cuối của cửa sổ dự báo vô hạn cost - to - go sử dụng hàm Lyapounov điều khiển tương thích.

Các tác giả Poignet và Gautier (2000) đã đưa ra MPC phi tuyến cho máy điều khiển robot và thí nghiệm với mô hình động học tuyến tính hóa và tối ưu hóa [47]. Wroblewski (2004) đã áp dụng MPC để điều khiển máy điều khiển 6 bậc tự do để thử nghiệm giảm sai số ở cuối phạm vi dự báo tốt hơn là tìm giải pháp tối ưu để giảm khối lượng tính toán [71]. Một sự xấp xỉ MPC phi tuyến cũng được đưa ra để điều khiển nhiều máy bay tự lái trong môi trường phức tạp; Việc xấp xỉ này phối hợp với sự ổn định của động học máy bay và sự phát sinh quỹ đạo phân tán qua việc kết hợp chức năng tiềm tàng [61]. Mô hình hóa, điều khiển và ra quyết định đối với các robot bay được thảo luận [29], trong đó các tác giả đã thiết kế điều khiển bám sử dụng MPC phi tuyến. Trong [65], tác giả đề cập việc ứng dụng MPC cho bộ dẫn động song song. Trong những nghiên cứu đó, một chiến lược ĐKDB dựa trên mô hình động học đơn giản được cài đặt. Kawati (2002) đã sử dụng sự xấp xỉ RHC (Receding Horizon Control) phi tuyến điều khiển sự truyền động robot [36]. Ở tài liệu này các tác giả đã đưa ra sơ đồ RHC ổn định dựa trên hàm Lyapunov điều khiển và luật điều khiển phản hồi tương ứng. Gu (2005) cũng sử dụng bộ điều khiển RHC để điều chỉnh robot di động không holonôm [30]. Trong nghiên cứu này sự ổn định của RHC được đảm bảo bằng việc thêm vào hàm phạt trạng thái cuối của hàm mục tiêu và dải (phạm vi) trạng thái cuối của các ràng buộc tối ưu hóa. Sự địa phương hóa và điều hướng của truyền động robot sử dụng kỹ thuật NMPC (Nonlinear Model Prediction Control) được đưa ra trong [19].

Tác giả Akbar Rahideha, b, \*, M. Hasan Shaheedb (2012) đã sử dụng sự biến đổi MPC Newton được đưa ra để điều khiển hệ thống phi tuyến bị ràng buộc MIMO [55]. Đây là một biến thể của tác giả Li và Biegler (1989) trong [38]. Sự khác biệt giữa nghiên cứu này và nghiên cứu gốc là cách lập

công thức (cách trình bày). Li và Biegler miêu tả phần đầu vào của hàm mục tiêu dựa trên sự khác nhau giữa quỹ đạo đầu vào hiện thời và xung quanh điểm làm việc. Ở đây Akbar Rahideha, b, \*, M. Hasan Shaheedb sử dụng đầu vào khác nhau giữa hai lần lấy mẫu liên tiếp trong suốt phạm vi điều khiển. Nếu tất cả các biến trạng thái của hệ thống đều đo được hoặc quan sát được thì phải sử dụng MPC phản hồi trạng thái. Tuy nhiên, MPC phi tuyến phản hồi đầu ra có thể đạt được khi liên kết với MPC phi tuyến phản hồi trạng thái và một bộ quan sát trạng thái [21], [28], [32]. Sự ổn định, sự bền vững và đặc tính của MPC phi tuyến phản hồi đầu ra đã được nghiên cứu trong [32], nhưng trong tài liệu này sự ổn định của MPC phi tuyến phản hồi đầu ra được đưa vào bên trong phép tính. Kothare và Morari (2000) đã sử dụng thêm các điều kiện ràng buộc trạng thái (còn gọi là ràng buộc rút gọn - contractive constraint), để chứng minh đặc tính ổn định mạnh xung quanh điểm làm việc của MPC đối với các hệ thống phi tuyến bị ràng buộc [21].

O'Brien (2006) đã sử dụng mô hình trạng thái phụ thuộc để thiết kế bộ điều khiển dự báo theo mô hình cho hệ thống nước thải [46]. Dựa trên mô hình không gian trạng thái phụ thuộc trạng thái (state - dependent state-space model), xây dựng mô hình thời gian biến đổi tuyến tính sử dụng quỹ đạo tương lai xác định thông qua bộ điều khiển dự báo. Một sơ đồ Newton để giải quyết việc xấp xỉ trực tuyến các bài toán tối ưu hóa cho điều khiển phản hồi tối ưu được đưa ra trong [23]. Thông qua việc xấp xỉ điều khiển phản hồi tối ưu chính xác thì ảnh hưởng của nhiễu tác động nhanh có thể đạt được. Diehl (2005) đã nghiên cứu về phương pháp Newton để tối ưu hóa trực tuyến trong NMPC được gọi là sơ đồ lặp thời gian thực, chỉ với 1 vòng lặp đã đưa ra được quy tắc mã hóa nén thời gian trích mẫu để có được thuật toán tối ưu hóa trực tuyến nhanh [24].

Hơn nữa, một số các mô hình được xây dựng trong lĩnh vực điều khiển bền vững. MPC bền vững được nghiên cứu toàn diện trong các tài liệu [34],

[43]. Một nghiên cứu dựa trên sự bền vững vốn có của MPC trong [43], trong đó bỏ qua độ bất định đã được Nicolao và Magni thực hiện lần lượt trong trường hợp MPC cho hệ thống có thời gian gián đoạn và thời gian liên tục. MPC min - max đã được đề xuất trong một số các công trình, trong [42], tác giả khẳng định vòng hở min - max được bảo toàn thực sự và vì thế mà một phương án phản hồi MPC được đề xuất trong [19], [26], [40], [58], [67]. Cuzzola (2002) đã cải tiến luật điều khiển phản hồi trạng thái nhằm cực tiểu hóa hàm mục tiêu với cửa sổ dự báo vô hạn phụ thuộc vào các điều kiện ràng buộc sử dụng hàm Lyapunov ứng với từng đáp ứng tại mỗi đỉnh khác nhau của đa diện bất định thay vì dùng hàm Lyapunov đơn [19]. Trong [40], Mao (2003) đã khắc phục lỗi của [19] và hoàn thiện nghiên cứu này. Một sự cải tiến khác dựa trên nghiên cứu của [19] được đưa ra trong [26] bằng cách bổ sung thêm  $N$  khoảng dịch chuyển điều khiển tự do trước luật phản hồi tuyến tính. Khái niệm ellipsoid bất biến ổn định tiệm cận được dùng để phát triển mô hình MPC bị ràng buộc bền vững. Phương pháp đưa ra một chuỗi các luật điều khiển tường minh (explicit) ứng với một chuỗi ellipsoid bất biến ổn định tiệm cận có cấu trúc ngoại tuyến, trong không gian trạng thái ở tài liệu [67]. Scokaert và Mayne (1998) đã phác họa chi tiết cách xây dựng MPC min - max, trong đó giới thiệu khái niệm phản hồi (presenting feedback) trong khi tiến hành điều khiển [48]. Wang và Rawlings (2004) đã bổ sung thêm phương pháp MPC bền vững để đảm bảo ổn định và offset - free điểm đặt bám trong sự hiện diện của mô hình bất định [69]. Bộ điều chỉnh bền vững sử dụng một quỹ đạo tự do để dự báo mô hình bất định thời gian biến đổi.

MPC phản hồi đầu ra bền vững gần đây cũng được nghiên cứu trong các công trình [25], [42], [66]. Wan và Kothare (2002) đã phát triển MPC phản hồi đầu ra ngoại tuyến, trước tiên là một bộ MPC phản hồi trạng thái bị ràng buộc bền vững ngoại tuyến và một bộ ước lượng trạng thái được thiết kế độc lập rồi sau đó phối hợp sự ổn định bền vững của bộ điều khiển và phân

tích hàm ước lượng [66]. Nếu sự ổn định bền vững không thảo mãn, việc thiết kế bộ điều khiển và hàm ước lượng được lặp lại thông qua một bộ tham số mới để thu được kết quả thỏa mãn. MPC phản hồi đầu ra bền vững của các hệ thống bị ràng buộc, tuyến tính và thời gian rời rạc trong bề mặt trạng thái biên và các nhiễu đầu ra được đưa ra bởi Mayne trong [42]. Phương pháp này bao gồm sự ước lượng trạng thái Lueberger ổn định trong đó các sai lệch của nó được giới hạn bởi một tập hợp bất biến và một sự ổn định bền vững MPC. Trong [25], Ding (2007) đã đưa ra phương pháp tổng hợp MPC bền vững phản hồi đầu ra cho các hệ thống được miêu tả bởi siêu diện, nhiễu trạng thái được giới hạn và đo được nhiễu. Một chuỗi luật phản hồi đầu ra được gọi là ngoại tuyến dựa trên bộ ước lượng trạng thái thông qua việc giải bài toán tối ưu LMI. Ở mỗi một bước, luật phản hồi đầu ra tiện dụng được lựa chọn trực tuyến từ những chuỗi này. Lưu ý rằng hầu hết các phương pháp MPC phản hồi đầu ra bền vững đều khó thực hiện trong các hệ điều khiển thời gian thực.

Tác giả Akbar Rahideh (2009) của luận án [10] đã đề cập tương đối đầy đủ và chi tiết về hệ phi tuyến TRMS, tác giả của luận án này đã tiến hành xây dựng mô hình toán đối tượng phi tuyến TRMS theo mô hình hộp trắng dựa trên nền phương pháp Newton và Lagrange để xây dựng mô hình toán một bậc tự do theo phương dọc và một bậc tự do theo phương ngang từ đó xây dựng mô hình hai bậc tự do của đối tượng và so sánh hai mô hình này với nhau cho kết quả sai lệch rất nhỏ. Xây dựng mô hình toán của đối tượng theo mô hình hộp xám và cả mô hình hộp đen (đây là đóng góp chính của luận án); Khi xây dựng mô hình theo mô hình hộp xám tác giả này đã sử dụng giải thuật di truyền để tính toán tối ưu các tham số trong quá trình nhận dạng; Sử dụng phương pháp xấp xỉ mạng nơron để xây dựng mô hình hộp đen; nội dung luận án này cũng xét đến tính ổn định của đối tượng phi tuyến TRMS dựa theo phương pháp ràng buộc điểm cuối; xây dựng bộ điều khiển dự báo cho đối tượng này với cửa sổ dự báo hữu hạn; cũng trong luận án này còn đưa

ra việc xây dựng bộ điều khiển dự báo phản hồi đầu ra để điều khiển hệ phi tuyến TRMS. Ngoài ra luận án đó cũng đề cập đến bộ quan sát trạng thái thông qua EKF (Extended Kalman Filter) và UKF (Unscented Kalman Filter). Từ việc tiếp cận, đến nghiên cứu các kết quả đưa ra trong luận án này, phân tích các vấn đề còn chưa được khai thác của luận án, tôi nhận thấy, khi xây dựng bộ điều khiển dự báo để điều khiển đối tượng phi tuyến TRMS trong luận án trên còn giới hạn ở phạm vi nghiên cứu cụ thể sau:

- Đã sử dụng các định luật cân bằng của Newton và Lagrange để xây dựng mô hình toán cho đối tượng phi tuyến TRMS một bậc tự do theo phương dọc, một bậc tự do theo phương ngang và hai bậc tự do, trong đó có đề cập đến hầu hết các lực tác động đến hệ thống.
- Mới sử dụng duy nhất phương pháp SQP giải bài toán tối ưu để tìm giá trị cực tiểu của hàm mục tiêu. Đây là một trong các phương pháp thuộc quy hoạch phi tuyến để giải bài toán tối ưu.
- Xét tính ổn định của hệ thống phi tuyến TRMS dựa trên phương pháp ràng buộc điểm cuối, đưa ra hàm phạt nhưng chưa chỉ ra quy luật tìm hàm phạt đó như thế nào.
- Cửa sổ dự báo hữu hạn ( $N_p = 20$ ;  $N_c = 15$ ).

Trong điều khiển dự báo một trong hai công việc vô cùng quan trọng là phải giải bài toán tối ưu phi tuyến với các ràng buộc. Ở hầu hết các công trình nghiên cứu về điều khiển tối ưu cho hệ phi tuyến các tác giả đã sử dụng 2 chiến lược giải bài toán tối ưu cơ bản là: quy hoạch phi tuyến và điều khiển tối ưu.

## 1.2. Các phương pháp quy hoạch phi tuyến

Nội dung chi tiết của các phương pháp quy hoạch phi tuyến được đề cập chi tiết trong các tài liệu [3], [5]. Các phương pháp cơ bản như sau:



### **1.2.1. Phi tuyến không bị ràng buộc**

**1.2.1.1. Các phương pháp Line search gồm:** PP Gradient, PP Newton-Raphson (Quasi Newton), PP Gauss – Newton

+ *Ưu điểm:* đơn giản, dễ cài đặt...

+ *Nhược điểm:* có thể tìm được nghiệm tối ưu địa phương, không tìm được nghiệm tối ưu toàn cục.

**1.2.1.2. Tìm không theo hướng gồm:** Phương pháp Levenberg – marquardt, Phương pháp miền tin cậy

+ *Ưu điểm:* đơn giản, dễ cài đặt...

+ *Nhược điểm:* có thể tìm được nghiệm tối ưu địa phương, không tìm được nghiệm tối ưu toàn cục.

**1.2.2. Bài toán tối ưu hóa phi tuyến bị ràng buộc gồm: Kỹ thuật hàm phạt và hàm chặn, Phương pháp SQP [3], [5] và GA [2]**

+ *Ưu điểm:* Dễ dàng xử lý được các điều kiện ràng buộc, kể cả các điều kiện ràng buộc về giá trị tín hiệu điều khiển, về số gia của tín hiệu điều khiển và về biến trạng thái của hệ thống.

+ *Nhược điểm:* Chỉ áp dụng được cho bộ điều khiển được xây dựng từ mô hình không liên tục của đối tượng; Chỉ sử dụng được với cửa sổ dự báo hữu hạn, do đó để đảm bảo được chất lượng ổn định hoặc bám ổn định theo giá trị đặt trước cần phải chọn được một hàm phạt thích hợp.

### **1.3. Các phương pháp điều khiển tối ưu**

**Các phương pháp điều khiển tối ưu gồm: Phương pháp biến phân, nguyên lý cực đại, phương pháp quy hoạch động**

+ *Ưu điểm:* Dễ dàng áp dụng được cho hệ phi tuyến liên tục và không dừng chứ không chỉ riêng hệ song tuyến; áp dụng được với cửa sổ dự báo vô hạn nên không cần có thêm hàm phạt, vốn rất khó, thậm chí chưa có gợi ý hữu ích

nào cho việc xác định chúng. Thời gian tính toán khi giải bài toán tối ưu trong chiến lược tối ưu hóa được cải thiện đáng kể.

+ *Nhược điểm*: Khó xử lý các điều kiện ràng buộc phức tạp.

#### **1.4. Các công trình nghiên cứu về điều khiển dự báo hệ phi tuyến trong nước**

Trên thế giới điều khiển dự báo đã được rất nhiều các tác giả quan tâm nghiên cứu. Ở Việt Nam, vấn đề này cũng được một số học giả tiếp cận, nghiên cứu trong khoảng hơn một thập niên trở lại đây. Các kết quả nghiên cứu chủ yếu là các luận văn thạc sĩ của các học viên cao học thuộc Đại học Bách Khoa Hà Nội, Đại học Bách Khoa Thành phố Hồ Chí Minh, Đại học Kỹ Thuật Công nghiệp Thái Nguyên. Cùng với xu hướng phát triển của thế giới, ở trong nước việc nghiên cứu về điều khiển dự báo hệ tuyến tính như GPC, DMC... đã được thực hiện tương đối trọn vẹn. Đến nay, điều khiển dự báo hệ phi tuyến đã và đang được nghiên cứu bởi nhiều nhà khoa học, các nghiên cứu sinh và các học viên cao học tại các trường Đại học, các Viện nghiên cứu... Một số những công trình nghiên cứu cụ thể như sau:

Trong tài liệu [1], tác giả của luận án này đã xây dựng bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ phi tuyến. Nội dung của luận án này đã có những đóng góp cơ bản:

- Đưa ra tiêu chuẩn ổn định của hệ ĐKDB phản hồi trạng thái cho hệ phi tuyến mà ở đó hàm mục tiêu có cấu trúc biến đổi theo cửa sổ dự báo trên trục thời gian;
- Xây dựng bộ điều khiển dự báo phản hồi trạng thái cho hệ phi tuyến và điều kiện để hệ kín thu được là ổn định tiệm cận;
- Xây dựng thuật toán quan sát trạng thái tối ưu cho hệ phi tuyến và điều kiện để bộ quan sát đó trở thành bộ quan sát có khoảng thời gian sát hữu hạn;
- Đưa ra điều kiện đủ để hệ phi tuyến là quan sát đều và xây dựng thuật toán quan sát trạng thái tối ưu cho hệ phi tuyến;

- Đưa ra điều kiện đủ để bộ điều khiển dự báo phản hồi đầu ra, xây dựng trên nền nguyên lý tách, làm hệ phi tuyến nói chung và hệ phi tuyến nói riêng là ổn định tiệm cận.

Trong tài liệu này không tập trung vào việc nghiên cứu về chiến lược tối ưu hóa trong điều khiển dự báo mà *chủ yếu đề cập đến việc xây dựng bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ phi tuyến* để xét tính ổn định tiệm cận của hệ do đó chưa đề cập đến tính bám ổn định của hệ thống điều khiển dự báo cho hệ phi tuyến, và mô hình đối tượng mà tác giả này sử dụng vẫn là mô hình không liên tục.

Trong tài liệu [6], tác giả của luận án này đã thực hiện xây dựng mô hình thích nghi tham số trực tuyến trên cơ sở ước lượng tham số mô hình mờ cho đối tượng phi tuyến có thành phần bất định là hàm số. Luận án này cũng đã phát biểu và chứng minh tính ổn định cho hệ kín khi sử dụng bộ điều khiển thiết kế bằng giải thuật dự báo thích nghi. Luận án này cũng không nghiên cứu chiến lược tối ưu hóa trong điều khiển dự báo mà đi sâu vào việc xây dựng mô hình.

### **1.5. Những vấn đề cần tiếp tục nghiên cứu về điều khiển dự báo cho hệ phi tuyến và hướng nghiên cứu của luận án**

Thông qua việc giới thiệu và đánh giá về các công trình đã nghiên cứu về điều khiển dự báo cho hệ phi tuyến; đưa ra các phương pháp giải bài toán tối ưu trong điều khiển dự báo là quy hoạch phi tuyến và điều khiển tối ưu trong mục 1.2 và mục 1.3 ta thấy rằng mỗi phương pháp có ưu, nhược điểm riêng. Mặc dầu đã đạt được những kết quả đáng kể cả trong lý thuyết và ứng dụng thực tiễn, song điều khiển dự báo (đặc biệt đối với điều khiển dự báo hệ phi tuyến) vẫn còn một số vấn đề tồn tại cần được tiếp tục nghiên cứu hoàn thiện, đó là:

- Nâng cao độ chính xác của mô hình dự báo, các mô hình dự báo trong các công trình đã công bố tuy đã kể đến tính phi tuyến nhưng đã bỏ qua (hoặc chưa kể đến) tính chất vật lý của đối tượng điều khiển như biến dạng đàn hồi của vật liệu, ma sát giữa các bộ phận tiếp xúc, độ dư của cơ cấu truyền động như bánh răng, trục vít, bánh vít, đai truyền, v.v...

- Chưa có công trình nào đề cập đến việc chọn và thỏa hiệp giữa các phiếm hàm mục tiêu đối nghịch nhau khi thực hiện thuật toán tối ưu hóa trong điều khiển dự báo phi tuyến.

- Tìm ra thuật toán mới để giải bài toán tối ưu nhằm cải thiện tốc độ tính toán và nâng cao độ chính xác, tính ổn định, mở rộng tầm dự báo của điều khiển dự báo phi tuyến, đặc biệt đối với hệ phi tuyến.

### **Hướng nghiên cứu của luận án**

Dựa trên các phân tích và nghiên cứu của các bài báo, các công trình đã công bố trong và ngoài nước, đặc biệt dựa vào tài liệu [10], tác giả nhận thấy: Từ trước đến nay, trong điều khiển dự báo người ta bắt buộc phải sử dụng mô hình không liên tục của đối tượng cũng như cửa sổ dự báo hữu hạn, vì chỉ như vậy mới có thể áp dụng được các phương pháp tối ưu hóa (nghĩa là giải bài toán tối ưu bằng các phương pháp quy hoạch phi tuyến). Trong nội dung của luận án này tác giả đề xuất thuật toán sử dụng trực tiếp mô hình liên tục vào điều khiển dự báo và nghiệm tối ưu trong cửa sổ dự báo hiện tại được tìm theo phương pháp biến phân của điều khiển tối ưu thay vì các phương pháp tối ưu hóa nên cửa sổ dự báo mở rộng đến vô cùng. Vì vậy, tác giả đã đề xuất hướng nghiên cứu của luận án là:

*Nghiên cứu, xây dựng thuật toán mới giải bài toán tối ưu của chiến lược tối ưu hóa trong điều khiển dự báo phi tuyến với mục đích mở rộng cửa sổ dự báo đến vô cùng nhằm nâng cao tính ổn định và chính xác của hệ thống.*

*Đồng thời rút ngắn thời gian tính toán khi giải bài toán tối ưu so với các phương pháp đã đề cập trước đó.*

### **1.6. Kết luận chương 1**

Chương 1 của luận án đã tập trung nghiên cứu vấn đề tổng quan về điều khiển dự báo cho hệ phi tuyến, đề cập và phân tích các công trình, bài báo của các tác giả trong và ngoài nước xung quanh vấn đề này. Cũng trong chương này tác giả đã thống kê các phương pháp giải bài toán tối ưu trong điều khiển dự báo áp dụng cho cả hệ phi tuyến không bị ràng buộc và hệ phi tuyến bị ràng buộc, nêu ra những ưu, nhược điểm của mỗi phương pháp đó. Căn cứ vào việc phân tích, tổng hợp và nghiên cứu khá kỹ lưỡng những vấn đề mà các nhà nghiên cứu đã đưa ra, cuối chương tác giả đã đưa ra những vấn đề cần tiếp tục nghiên cứu về điều khiển dự báo cho hệ phi tuyến và hướng nghiên cứu của luận án.

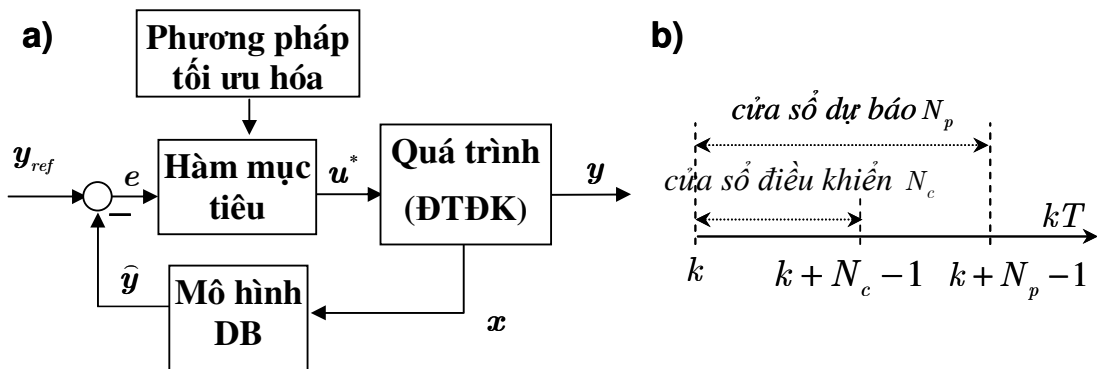
## Equation Chapter (Next) Section 1 CHƯƠNG 2

### ĐIỀU KHIỂN DỰ BÁO HỆ PHI TUYẾN TRÊN NỀN CÁC PHƯƠNG PHÁP QUY HOẠCH PHI TUYẾN

Trong chương này luận án sẽ trình bày cấu trúc điều khiển dự báo chủ đạo hiện nay đang được áp dụng cho hệ phi tuyến nói chung và hệ song tuyến nói riêng, vì nó gần như đáp ứng được các yêu cầu ràng buộc của bài toán điều khiển. Các bộ điều khiển dự báo được luận án trình bày ở mục 2.2 đã đáp ứng được những điều kiện bị chặn của bài toán điều khiển, song lại bị hạn chế về tốc độ tính toán, khó áp dụng vào điều khiển thời gian thực, và quan trọng nhất là hạn chế về tính hữu hạn của cửa sổ dự báo  $N_p$ . Do tính hữu hạn này của cửa sổ dự báo mà khả năng ổn định hệ thống của bộ điều khiển là chưa được đảm bảo, nếu như không sử dụng thêm kỹ thuật hàm phạt cho điểm trạng thái cuối. Hơn thế nữa, tất cả các bộ điều khiển dự báo này đều làm việc trên miền thời gian không liên tục, trong khi đối tượng thực tế lại luôn tồn tại ở dạng liên tục theo thời gian.

#### 2.1. Nguyên lý làm việc của điều khiển dự báo phi tuyến

Hình 2.1a) biểu diễn cấu trúc cơ bản của một hệ thống điều khiển dự báo.



Hình 2.1. Cấu trúc cơ bản của một hệ thống điều khiển dự báo

Hệ thống điều khiển dự báo làm việc theo nguyên lý:

1. Thứ nhất, xây dựng mô hình đối tượng dự báo các đầu ra tương lai cho một phạm vi (tầm) xác định trước, gọi là tầm dự báo  $N_p$ , ở mỗi thời điểm lấy mẫu  $k$ . Những đầu ra dự báo này, ký hiệu bởi  $\hat{y}(k+i|k), i=1,2, \dots, N_p$ , tính từ thời điểm  $k$ , sẽ phụ thuộc vào tín hiệu điều khiển tương lai  $u(k+i|k), i=1,2, \dots, N_p-1$  và  $u(k+i|k) = u(k+N_c|k)$ , trong đó  $i > N_c$  với  $N_c$  là tầm (phạm vi, giới hạn) điều khiển.
2. Thứ hai, các tín hiệu điều khiển tương lai được tính toán tối ưu để có đầu ra  $y$  của quá trình bám quỹ đạo đặt  $y_{ref}$  khi mà bản thân tín hiệu đặt hoặc đầu ra quá trình đều được xấp xỉ. Hàm mục tiêu thường sử dụng là hàm toàn phương sai lệch giữa tín hiệu đầu ra dự báo và quỹ đạo tham chiếu dự báo. Trong mọi trường hợp mục tiêu điều khiển là cực tiểu hóa hoặc cực đại hóa hàm mục tiêu.
3. Thứ ba, dựa trên khái niệm chiến lược dịch dần về tương lai, phần đầu tiên của tín hiệu điều khiển,  $u(k|k)$ , được gửi tới quá trình, các thành phần chính của tín hiệu điều khiển được đưa ra trong [17].

Ba nguyên lý làm việc trên luôn được thực hiện lặp để dịch dần về tương lai, tức là tại thời điểm  $k$  hiện tại, sau khi đã có tín hiệu điều khiển tối ưu  $u(k|k)$  đưa vào điều khiển đối tượng trong khoảng thời gian  $kT \leq t < (k+1)T$ , với  $T$  là bước dịch chuyển của khoảng cửa sổ điều khiển  $N_c$  và cũng là chu kỳ trích mẫu tín hiệu, cửa sổ điều khiển cùng cửa sổ dự báo được dịch về tương lai tới thời điểm  $k+1$ , lúc này đã trở thành thời điểm hiện tại. Khi đó ba nguyên lý trên lại được thực hiện lại để có  $u(k+1|k+1)$ , như được mô tả ở hình 2.1b).

Trong mục này, luận án sẽ trình bày các bước thực hiện ba nguyên lý làm việc trên của điều khiển dự báo dưới dạng tổng quan cho hệ song tuyến để từ đó tiến tới việc thực hiện thiết kế cụ thể và chi tiết các bộ điều khiển dự báo khác nhau cho hệ song tuyến TRMS, đối tượng ứng dụng chính của luận án (sẽ được trình bày cụ thể trong chương 4).

### 2.1.1. Cấu trúc bộ điều khiển dự báo

Hình 2.1 cho thấy trong một bộ điều khiển dự báo luôn có ba khối chính. Đó là các khối mô hình dự báo, khối hàm mục tiêu và khối tối ưu hóa. Trong khi khối mô hình dự báo được xác định ngay từ mô hình đối tượng điều khiển thì hai khối còn lại là khối hàm mục tiêu và tối ưu hóa giữ vai trò quyết định chính, mang tính chủ quan, về chất lượng của hệ thống. Hai khối này cũng là các thành phần cơ bản để phân biệt những bộ điều khiển dự báo khác nhau, áp dụng cho cùng một đối tượng điều khiển.

#### *Khối mô hình dự báo*

Đây là khối có chức năng sử dụng mô hình mô tả đối tượng để dự báo các tín hiệu đầu ra trong tương lai của nó. Hãy xét một đối tượng, hay quá trình phi tuyến, có mô hình toán dạng không liên tục như sau:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{y}_k = \mathbf{g}(\mathbf{x}_k) \end{cases} \quad \text{Equation Section (Next)(2.1)}$$

trong đó:

$\mathbf{x}_k = (x_1(kT), x_2(kT), \dots, x_n(kT))^T$  là vector của  $n$  giá trị trạng thái của hệ tính tại thời điểm  $t = kT, k = 0, 1, \dots$ , với  $T$  là chu kỳ trích mẫu tín hiệu,

$\mathbf{u}_k = (u_1(kT), u_2(kT), \dots, u_m(kT))^T$  là vector của  $m \leq n$  giá trị tín hiệu điều khiển (tín hiệu đầu vào),

$\mathbf{y}_k = (y_1(kT), y_2(kT), \dots, y_r(kT))^T$  là vector của  $r \leq m$  giá trị tín hiệu đáp ứng (tín hiệu đầu ra).

Nếu sử dụng ngay mô hình (2.1) của đối tượng làm mô hình dự báo, thì trong cửa sổ dự báo hiện tại  $i = 1, 2, \dots, N_p$  ta sẽ có các tín hiệu đầu ra dự báo được như sau:



$$\begin{aligned}
\hat{\mathbf{y}}(k+i|k) &= \mathbf{g}(\mathbf{x}_{k+i}) = \mathbf{g}(\mathbf{x}(k+i|k)) \\
&= \mathbf{g}(\mathbf{f}(\mathbf{x}(k+i-1|k), \mathbf{u}(k+i-1|k))) \\
&= \mathbf{g}(\mathbf{f}(\mathbf{f}(\mathbf{x}(k+i-2|k), \mathbf{u}(k+i-2|k)), \mathbf{u}(k+i-1|k))) \\
&\vdots \\
&= \mathbf{g}(\mathbf{f}(\dots (\mathbf{f}(\mathbf{x}(k|k), \mathbf{u}(k|k)), \mathbf{u}(k+1|k)), \dots), \mathbf{u}(k+i|k)) \\
&= \mathbf{g}(\mathbf{f}(\dots (\mathbf{f}(\mathbf{x}_k, \mathbf{u}(k|k)), \mathbf{u}(k+1|k)), \dots), \mathbf{u}(k+i|k)) \quad (2.2)
\end{aligned}$$

Nhưng do giá trị trạng thái  $\mathbf{x}_k$  ở thời điểm  $k$  hiện tại được giả thiết là đã đo được, nên thực chất  $\hat{\mathbf{y}}(k+i|k)$  ở công thức (2.2) chỉ còn là hàm phụ thuộc theo:

$$\mathcal{U} = \text{col}(\mathbf{u}(k|k), \mathbf{u}(k+1|k), \dots, \mathbf{u}(k+N_p-1|k)) \quad (2.3)$$

Do đó ta có thể viết nó ngắn gọn lại thành:

$$\hat{\mathbf{y}}(k+i|k) = \mathbf{p}_i(\mathcal{U}), \quad i = 0, 1, \dots, N_p \quad (2.4)$$

là ký hiệu của hàm dự báo (predictive function) tại thời điểm  $k+i$ , trong đó:

$$\mathbf{p}_0(\mathcal{U}) = \mathbf{g}(\mathbf{x}_k)$$

*Khối hàm mục tiêu*

Với mục đích là để tín hiệu ra  $\mathbf{y}_k$  của hệ bám theo được tín hiệu mẫu  $\mathbf{y}_{ref}$  nên trong điều khiển dự báo, người ta thường sử dụng hàm mục tiêu có chứa thành phần sai lệch:

$$\mathbf{e}_{k+i} = \mathbf{y}_{ref}((k+i)T) - \hat{\mathbf{y}}(k+i|k)$$

Do  $\mathbf{y}_{ref}$  là cho trước nên  $\mathbf{y}_{ref}((k+i)T)$  cũng là đã biết. Bởi vậy sai lệch trên thực chất là hàm của  $\mathcal{U}$  cho bởi (2.3), tức là:

$$\mathbf{e}_{k+i} = \mathbf{e}_{k+i}(\mathcal{U})$$

Khi đó, nếu sử dụng hàm mục tiêu có cấu trúc toàn phương:

$$J(\mathcal{U}) = \sum_{i=1}^{N_p} q_i |\mathbf{e}_{k+i}(\mathcal{U})|^2 + \sum_{j=0}^{N_p-1} r_j |\mathbf{u}(k+j|k)|^2 + s(\mathbf{x}(k+N_p|k)) \quad (2.5)$$

trong đó  $q_i, r_j$  là những hằng số dương (được gọi là trọng số), và  $s(\mathbf{x}(k + N_p | k))$  là hàm phạt điểm cuối, thì hiển nhiên cùng với mô hình dự báo (2.4) cho tín hiệu ra, và:

$$\begin{aligned} \mathbf{x}(k + N_p | k) &= \mathbf{f}(\mathbf{x}(k + N_p - 1 | k), \mathbf{u}(k + N_p - 1 | k)) \\ &= \mathbf{f}(\mathbf{f}(\mathbf{x}(k + N_p - 2 | k), \mathbf{u}(k + N_p - 2 | k)), \mathbf{u}(k + N_p - 1 | k)) \\ &\quad \vdots \\ &= \mathbf{f}(\dots (\mathbf{f}(\mathbf{x}_k, \mathbf{u}(k | k)), \mathbf{u}(k + 1 | k)), \dots), \mathbf{u}(k + N_p - 1 | k)) \end{aligned} \quad (2.6)$$

hàm mục tiêu trên trở thành hàm chỉ còn phụ thuộc vào vector  $\mathcal{U}$  cho bởi (2.3).

Gọi  $\mathcal{U}^*$  là nghiệm của:

$$J(\mathcal{U}^*) = \min_{\mathcal{U} \in U} J(\mathcal{U}) \quad (2.7)$$

với  $U$  là tập ràng buộc của các tín hiệu điều khiển, thì ứng với giá trị tối ưu tìm được:

$$\mathcal{U}^* = (\mathbf{u}^*(k | k), \mathbf{u}^*(k + 1 | k), \dots, \mathbf{u}^*(k + N_p - 1 | k))$$

ta sẽ có tổng các sai lệch dự báo  $e_{k+i}$ ,  $i = 1, 2, \dots, N_p - 1$  là nhỏ nhất.

*Khối tối ưu hóa*

Nhiệm vụ của khối này là thực thi bài toán (2.7), tức là tìm nghiệm tối ưu của:

$$\mathcal{U}^* = \arg \min_{\mathcal{U} \in U} J(\mathcal{U}) \quad (2.8)$$

Sau khi tìm được nghiệm tối ưu:

$$\mathcal{U}^* = \text{col}(\mathbf{u}^*(k | k), \mathbf{u}^*(k + 1 | k), \dots, \mathbf{u}^*(k + N_p - 1 | k)) \quad (2.9)$$

giá trị đầu tiên trong nó là:

$$\mathbf{u}_k^* = \mathbf{u}^*(k | k)$$

sẽ được đưa vào điều khiển đối tượng trong đúng khoảng thời gian trích mẫu và cũng là khoảng dịch chuyển của cửa sổ dự báo  $kT \leq t < (k + 1)T$ .

### 2.1.2. Kỹ thuật cài đặt bộ điều khiển dự báo trên nền các phương pháp quy hoạch phi tuyến

Có rất nhiều các phương pháp tối ưu hóa phục vụ việc cài đặt thuật toán tìm nghiệm tối ưu cho bài toán (2.8) của bộ điều khiển dự báo. Chẳng hạn:

1. Với bài toán (2.8) không bị ràng buộc ( $U = \mathbb{R}^{mN_p}$ ), ta có thể sử dụng các thuật toán như Gradient, Newton hay Quasi Newton, Gauss - Newton, Levenberg - Marquardt ....
2. Khi có thêm điều kiện ràng buộc ( $U \subset \mathbb{R}^{mN_p}$ ), những thuật toán thích hợp sẽ là kỹ thuật hàm chặn, hàm phạt, hay QP (quadratic programming) hoặc SQP (sequential quadratic programming) hay giải thuật di truyền (Genetic Algorithm) ...

Tất cả những thuật toán này, có tên thường gọi chung là quy hoạch phi tuyến, đã được trình bày chi tiết trong các tài liệu [5], [45]. Tuy nhiên trong điều khiển dự báo, khi phải xử lý các điều kiện ràng buộc kèm theo, thì những phương pháp quy hoạch phi tuyến thường được sử dụng để tìm nghiệm bài toán tối ưu (2.8) có thể kể đến là:

- a. Phương pháp QP
- b. Phương pháp SQP
- c. Phương pháp điểm bên trong (interior point)

#### *Cài đặt với phương pháp QP (Quadratic Programming)*

Đây là phương pháp được áp dụng cho bài toán tối ưu (2.8) mà ở đó hàm mục tiêu  $J(U)$  có dạng toàn phương và điều kiện ràng buộc  $U$  được mô tả bằng phương trình và bất phương trình tuyến tính:

$$J(U) = \frac{1}{2} U^T A U + \mathbf{b}^T U \xrightarrow{U \in U} \min \quad (2.10)$$

$$U = \left\{ U \in \mathbb{R}^{mN_p} \mid \begin{array}{l} \mathbf{c}_i^T U = d_i, \quad i = 1, 2, \dots, r' \\ \mathbf{c}_j^T U \leq d_j, \quad j = r' + 1, \dots, r \end{array} \right\}$$

QP là phương pháp lặp tìm nghiệm qua nhiều bước. Tại mỗi bước tính, phương pháp dựa vào định lý Karush - Kuhn - Tucker để thay thế miền các

điều kiện ràng buộc cho ở trên thành miền con mô tả bởi các phương trình tuyến tính, được gọi là tập tích cực (active set). Chi tiết các bước tính của phương pháp này có thể được tìm thấy trong những tài liệu tham khảo như [5] và [45].

*Cài đặt với phương pháp SQP (Sequential Quadratic Programming)*

Phương pháp SQP là sự mở rộng của QP để có thể tìm nghiệm của bài toán tối ưu phi tuyến bị ràng buộc  $U$ :

$$\mathcal{U}^* = \arg \min_{\mathcal{U} \in U} J(\mathcal{U}) \quad (2.11)$$

$$U = \left\{ \mathcal{U} \in \mathbb{R}^{mN_p} \mid \begin{array}{l} g_i(\mathcal{U}) = 0, \quad i = 1, 2, \dots, r' \\ g_j(\mathcal{U}) \leq 0, \quad j = r' + 1, \dots, r \end{array} \right\}$$

tức là bài toán tối ưu hóa mà ở đó có cả hàm mục tiêu  $J(\mathcal{U})$  và các hàm mô tả điều kiện ràng buộc  $g_i(\mathcal{U})$ ,  $i = 1, 2, \dots, r$  đều là những hàm phi tuyến.

Tư tưởng chính của SQP là từng bước chuyển bài toán tối ưu phi tuyến (2.11) về dạng toàn phương có điều kiện ràng buộc là tuyến tính (2.10) rồi sau đó áp dụng phương pháp QP. Công cụ chuyển từng bước (2.11) về dạng (2.10) trong mỗi bước tính lặp là khai triển chuỗi Taylor.

Giả sử tại vòng lặp thứ  $k$  đã có nghiệm trung gian  $\mathcal{U}_k$ . Để tìm lượng hiệu chỉnh  $\mathbf{h}$  vào  $\mathcal{U}_k$  sao cho ở bước tiếp theo với  $\mathcal{U}_{k+1} = \mathcal{U}_k + \mathbf{h} \in U$  có  $J(\mathcal{U}_{k+1}) < J(\mathcal{U}_k)$ , người ta khai triển Taylor hàm  $J(\mathcal{U}_{k+1})$  thành:

$$\begin{aligned} J(\mathcal{U}_{k+1}) &= J(\mathcal{U}_k) + \frac{\partial J(\mathcal{U}_k)}{\partial \mathcal{U}} \mathbf{h} + \frac{1}{2} \mathbf{h}^T \frac{\partial^2 J(\mathcal{U}_k)}{\partial \mathcal{U}^2} \mathbf{h} \\ &= J(\mathcal{U}_k) + \mathbf{b}_k^T \mathbf{h} + \frac{1}{2} \mathbf{h}^T A_k \mathbf{h} \quad \text{với } \mathbf{b}_k^T = \frac{\partial J(\mathcal{U}_k)}{\partial \mathcal{U}} \quad \text{và } A_k = \frac{\partial^2 J(\mathcal{U}_k)}{\partial \mathcal{U}^2} \end{aligned}$$

cũng như các hàm  $g_i(\mathcal{U})$ ,  $i = 1, 2, \dots, r$  thành:

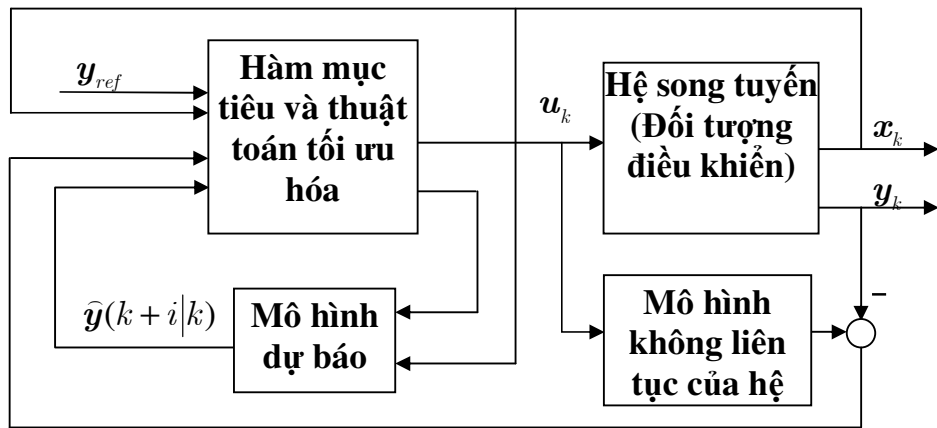
$$g_i(\mathcal{U}) = g_i(\mathcal{U}_k) + \frac{\partial g_i(\mathcal{U}_k)}{\partial \mathcal{U}} \mathbf{h} = -d_i + \mathbf{c}_i^T \mathbf{h}$$

với  $d_i = -g_i(\mathcal{U}_k)$  và  $\mathbf{c}_i^T = \frac{\partial g_i(\mathcal{U}_k)}{\partial \mathcal{U}}$ .

Khi đó bài toán tối ưu hóa phi tuyến (2.11) ở bước tính lặp thứ  $k$  đã được chuyển về dạng toàn phương với các điều kiện ràng buộc là tuyến tính như ở (2.10) để tìm  $h$ .

## 2.2. Áp dụng vào điều khiển dự báo lớp hệ song tuyến

Mục này sẽ trình bày các nghiên cứu của luận án về việc xây dựng bộ điều khiển dự báo phi tuyến (Nonlinear Model Predictive Control - NMPC) phản hồi trạng thái đầy đủ để điều khiển đối tượng song tuyến bằng việc sử dụng các phương pháp quy hoạch phi tuyến.



Hình 2.2. Sơ đồ khối của MPC để điều khiển hệ song tuyến

Thông thường, sử dụng các phương pháp MPC tuyến tính khi mà mô hình đối tượng và các điều kiện ràng buộc là tuyến tính, thuật toán điều khiển trở nên đơn giản, bài toán tối ưu (2.8) luôn cho ra nghiệm toàn cục, và điều này hoàn toàn không phụ thuộc vào thuật toán tối ưu hóa được lựa chọn để cài đặt. Do đó MPC tuyến tính đã trở nên hấp dẫn với chiến lược điều khiển phản hồi, đặc biệt đối với các quá trình tuyến tính. Chính vì lý do này mà đối với hệ song tuyến, thừa ban đầu người ta đã xấp xỉ mô hình của nó xung quanh các điểm cân bằng để ứng dụng được những phương pháp điều khiển MPC tuyến tính phản hồi đầu ra, chẳng hạn như các phương pháp MAC (Model Algorithmic), DMC (Dynamic Matrix Control), GPC (Generalized Predictive Control) ..., hoặc phản hồi trạng thái tuyến tính.

Tuy nhiên, các mô hình tuyến tính là chưa đủ để miêu tả đặc tính động học quá trình khi yêu cầu sản phẩm chất lượng cao, tăng năng suất, các quy

định môi trường chặt chẽ và yêu cầu về tính kinh tế trong các quá trình công nghiệp... đòi hỏi các hệ thống phải hoạt động khép kín trong phạm vi cho phép, vì thế các mô hình phi tuyến được sử dụng. Đối với hệ song tuyến cũng vậy, hiện nay đã có một vài bộ điều khiển dự báo được xây dựng ngay từ mô hình phi tuyến của đối tượng mà không phải thông qua tuyến tính hóa xấp xỉ. Những bộ điều khiển dự báo phi tuyến này chủ yếu được phân biệt với nhau ở phương pháp cài đặt thuật toán quy hoạch phi tuyến (nonlinear programming) thực hiện bài toán tối ưu hóa (2.8). Sơ đồ khối của MPC nói chung để điều khiển đối tượng song tuyến được biểu diễn trong hình 2.2.

### 2.2.1. Thuật toán điều khiển dự báo phi tuyến cho hệ song tuyến

Xét mô hình hệ song tuyến dưới dạng liên tục như sau:

$$\begin{cases} \dot{\mathbf{x}} = A(\mathbf{x})\mathbf{x} + B(\mathbf{x})\mathbf{u}, & \mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m \\ \mathbf{y} = C(\mathbf{x})\mathbf{x}, & \mathbf{y} \in \mathbb{R}^m \end{cases} \quad (2.12)$$

Nếu viết lại mô hình đó bằng cách trích mẫu tín hiệu với chu kỳ trích mẫu  $T_a$  và sử dụng phép xấp xỉ cho công thức tính đạo hàm:

$$\frac{d\mathbf{x}}{dt} \approx \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{T_a}, \quad \mathbf{x}_k = \mathbf{x}(kT_a), \quad \mathbf{u}_k = \mathbf{u}(kT_a), \quad \mathbf{y}_k = \mathbf{y}(kT_a) \quad (2.13)$$

ta sẽ được mô hình không liên tục:

$$\begin{cases} \mathbf{x}_{k+1} = \mathcal{A}(\mathbf{x}_k)\mathbf{x}_k + \mathcal{B}(\mathbf{x}_k)\mathbf{u}_k \\ \mathbf{y}_k = \mathcal{C}(\mathbf{x}_k)\mathbf{x}_k \end{cases} \quad (2.14)$$

trong đó:

$$\mathcal{A}(\mathbf{x}_k) = I + T_a A(\mathbf{x}_k), \quad \mathcal{B}(\mathbf{x}_k) = T_a B(\mathbf{x}_k), \quad \mathcal{C}(\mathbf{x}_k) = C(\mathbf{x}_k) \quad (2.15)$$

Từ đây ta có được mô hình dự báo cho hệ song tuyến trong toàn bộ cửa sổ dự báo hiện tại  $[k, k + N_p)$  như sau:

$$\begin{cases} \mathbf{x}(k+i+1|k) = \mathcal{A}(k+i)\mathbf{x}(k+i|k) + \mathcal{B}(k+i)\mathbf{u}(k+i|k) \\ \mathbf{y}(k+i|k) = \mathcal{C}(k+i)\mathbf{x}(k+i|k) \end{cases} \quad (2.16)$$

Ở công thức (2.16), ta đã sử dụng các ký hiệu:

$$\begin{aligned}\mathcal{A}(k+i) &= \mathcal{A}(\mathbf{x}(k+i|k)), \quad \mathcal{B}(k+i) = \mathcal{B}(\mathbf{x}(k+i|k)) \\ \mathcal{C}(k+i) &= \mathcal{C}(\mathbf{x}(k+i|k))\end{aligned}$$

cũng như nhất quán sử dụng ký hiệu:

$$\mathbf{x}(k+i|k), \quad \mathbf{u}(k+i|k) \quad \text{và} \quad \mathbf{y}(k+i|k)$$

thay vì  $\mathbf{x}_{k+i}$ ,  $\mathbf{u}_{k+i}$ ,  $\mathbf{y}_{k+i}$  khi  $i \geq 1$  nhằm để nhấn mạnh rằng chúng là những giá trị dự báo tại thời điểm  $k+i$  trong tương lai được tính từ thời điểm hiện tại tại  $k$ . Nói cách khác, đây là những giá trị chưa có từ đối tượng.

*Khởi mô hình dự báo*

Giả sử tất cả các biến trạng thái  $\mathbf{x}_k = \mathbf{x}(k|k)$  của hệ thống đều có thể đo được hoặc quan sát được ở thời điểm hiện tại  $k$ . Khi đó, tương tự như đã thực hiện tổng quát ở (2.2) và (2.6), các biến trạng thái  $\mathbf{x}(k+i|k)$  và đầu ra  $\hat{\mathbf{y}}(k+i|k)$  tương lai thuộc cửa sổ dự báo  $[k, k+N_p)$  với các biến trạng thái hiện thời phụ thuộc giá trị các đầu vào tương lai có thể biểu diễn dưới dạng hàm phụ thuộc  $\mathbf{u}(k+i|k)$ ,  $i=0, 1, \dots, N_p-1$  của cửa sổ dự báo đó, nhờ phép tính biểu diễn truy hồi thông qua mô hình dự báo (2.16) như sau:

$$\begin{aligned}\mathbf{x}(k+1|k) &= \mathcal{A}(k)\mathbf{x}(k|k) + \mathcal{B}(k)\mathbf{u}(k|k) \\ \mathbf{x}(k+2|k) &= \mathcal{A}(k+1)\mathbf{x}(k+1|k) + \mathcal{B}(k+1)\mathbf{u}(k+1|k) \\ &= \mathcal{A}(k+1)[\mathcal{A}(k)\mathbf{x}(k|k) + \mathcal{B}(k)\mathbf{u}(k|k)] + \mathcal{B}(k+1)\mathbf{u}(k+1|k) \\ &= \mathcal{A}(k+1)\mathcal{A}(k)\mathbf{x}(k|k) + \mathcal{A}(k+1)\mathcal{B}(k)\mathbf{u}(k|k) + \mathcal{B}(k+1)\mathbf{u}(k+1|k)\end{aligned}\tag{2.17}$$

⋮

và

$$\begin{aligned}\mathbf{x}(k+N_p|k) &= \mathcal{A}(k+N_p-1) \cdots \mathcal{A}(k+1)\mathcal{A}(k)\mathbf{x}(k|k) + \\ &\quad + \mathcal{A}(k+N_p-1) \cdots \mathcal{A}(k+1)\mathcal{B}(k)\mathbf{u}(k|k) + \\ &\quad + \mathcal{A}(k+N_p-1) \cdots \mathcal{A}(k+2)\mathcal{B}(k+1)\mathbf{u}(k+1|k) + \\ &\quad \vdots \\ &\quad + \mathcal{A}(k+N_p-1)\mathcal{B}(k+N_p-2)\mathbf{u}(k+N_p-2|k) + \\ &\quad + \mathcal{B}(k+N_p-1)\mathbf{u}(k+N_p-1|k)\end{aligned}\tag{2.18}$$

Để ngắn gọn trong trình bày, ta sẽ sử dụng ký hiệu:

$$\begin{aligned}\mathcal{A}_i^j &= \mathcal{A}(k+j)\mathcal{A}(k+j-1) \cdots \mathcal{A}(k+i) \\ &= \prod_{l=i}^j \mathcal{A}(k+l), \quad j \geq i\end{aligned}$$

Khi đó, các phương trình (2.17) trên trở thành:

$$\begin{aligned}\mathbf{x}(k+1|k) &= \mathcal{A}_k^k \mathbf{x}(k|k) + \mathcal{B}(k)\mathbf{u}(k|k) \\ \mathbf{x}(k+2|k) &= \mathcal{A}_k^{k+1} \mathbf{x}(k|k) + \mathcal{A}_{k+1}^{k+1} \mathcal{B}(k)\mathbf{u}(k|k) + \mathcal{B}(k+1)\mathbf{u}(k+1|k) \\ &\vdots \\ \mathbf{x}(k+N_p|k) &= \mathcal{A}_k^{k+N_p-1} \mathbf{x}(k|k) + \mathcal{A}_{k+1}^{k+N_p-1} \mathcal{B}(k)\mathbf{u}(k|k) + \cdots + \\ &\quad + \mathcal{A}_{k+N_p-1}^{k+N_p-1} \mathcal{B}(k+N_p-2)\mathbf{u}(k+N_p-2|k) + \\ &\quad + \mathcal{B}(k+N_p-1)\mathbf{u}(k+N_p-1|k)\end{aligned}$$

Suy ra:

$$\begin{aligned}\hat{\mathbf{y}}(k+1|k) &= \mathcal{C}(k+1)\mathcal{A}_k^k \mathbf{x}(k|k) + \mathcal{C}(k+1)\mathcal{B}(k)\mathbf{u}(k|k) \\ \hat{\mathbf{y}}(k+2|k) &= \mathcal{C}(k+2) \left[ \mathcal{A}_k^{k+1} \mathbf{x}(k|k) + \mathcal{A}_{k+1}^{k+1} \mathcal{B}(k)\mathbf{u}(k|k) + \mathcal{B}(k)\mathbf{u}(k+1|k) \right] \\ &\vdots \\ \hat{\mathbf{y}}(k+N_p|k) &= \mathcal{C}(k+N_p) \left[ \mathcal{A}_k^{k+N_p-1} \mathbf{x}(k|k) + \mathcal{A}_{k+1}^{k+N_p-1} \mathcal{B}(k)\mathbf{u}(k|k) + \right. \\ &\quad \left. + \cdots + \mathcal{B}(k+N_p-1)\mathbf{u}(k+N_p-1|k) \right]\end{aligned}\tag{2.19}$$

và:

$$\begin{aligned}\begin{pmatrix} \hat{\mathbf{y}}(k+1|k) \\ \hat{\mathbf{y}}(k+2|k) \\ \vdots \\ \hat{\mathbf{y}}(k+N_p|k) \end{pmatrix} &= \begin{pmatrix} \mathcal{C}(k+1)\mathcal{A}_k^k \\ \mathcal{C}(k+1)\mathcal{A}_k^{k+1} \\ \vdots \\ \mathcal{C}(k+N_p)\mathcal{A}_k^{k+N_p-1} \end{pmatrix} \mathbf{x}(k|k) + \\ &+ \begin{pmatrix} \mathcal{C}(k+1)\mathcal{B}(k) & 0 & \cdots & 0 \\ \mathcal{C}(k+2)\mathcal{A}_{k+1}^{k+1}\mathcal{B}(k) & \mathcal{C}(k+2)\mathcal{B}(k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{C}(k+N_p)\mathcal{A}_{k+1}^{k+N_p-1}\mathcal{B}(k) & \mathcal{C}(k+N_p)\mathcal{A}_{k+2}^{k+N_p-1}\mathcal{B}(k) & \cdots & \mathcal{C}(k+N_p)\mathcal{B}(k+N_p-1) \end{pmatrix} \mathbf{u}\end{aligned}$$

Ký hiệu:



$$M(\mathcal{U}) = \begin{pmatrix} \mathcal{C}(k+1)\mathcal{A}_k^k \\ \mathcal{C}(k+1)\mathcal{A}_k^{k+1} \\ \vdots \\ \mathcal{C}(k+N_p)\mathcal{A}_k^{k+N_p-1} \end{pmatrix}$$

$$N(\mathcal{U}) = \begin{pmatrix} \mathcal{C}(k+1)\mathcal{B}(k) & 0 & \cdots & 0 \\ \mathcal{C}(k+2)\mathcal{A}_{k+1}^{k+1}\mathcal{B}(k) & \mathcal{C}(k+2)\mathcal{B}(k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{C}(k+N_p)\mathcal{A}_{k+1}^{k+N_p-1}\mathcal{B}(k) & \mathcal{C}(k+N_p)\mathcal{A}_{k+2}^{k+N_p-1}\mathcal{B}(k) & \cdots & \mathcal{C}(k+N_p)\mathcal{B}(k+N_p-1) \end{pmatrix}$$

$$\mathcal{Y} = \text{col}(\hat{\mathbf{y}}(k+1|k), \hat{\mathbf{y}}(k+2|k), \dots, \hat{\mathbf{y}}(k+N_p|k))$$

ta sẽ được:

$$\mathcal{Y} = M(\mathcal{U})\mathbf{x}(k|k) + N(\mathcal{U})\mathcal{U} = M(\mathcal{U})\mathbf{x}_k + N(\mathcal{U})\mathcal{U} \quad (2.20)$$

và đây chính là dãy các giá trị đầu ra dự báo có được trong cửa sổ dự báo hiện tại  $[k, k+N_p)$ . Do trạng thái  $\mathbf{x}_k = \mathbf{x}(k|k)$  của hệ thống tại thời điểm  $k$  là đã có nhờ đo bằng cảm biến hoặc quan sát được nên  $\mathcal{Y}$  chỉ còn phụ thuộc  $\mathcal{U}$ .

*Xây dựng hàm mục tiêu*

Nếu thay quan hệ đã có ở công thức (2.17):

$$\begin{aligned} \mathbf{x}(k+N_p|k) &= \mathcal{A}_k^{k+N_p-1}\mathbf{x}(k|k) + \mathcal{A}_{k+1}^{k+N_p-1}\mathcal{B}(k)\mathbf{u}(k|k) + \cdots + \\ &\quad + \mathcal{A}_{k+N_p-1}^{k+N_p-1}\mathcal{B}(k+N_p-2)\mathbf{u}(k+N_p-2|k) + \mathcal{B}(k+N_p-1)\mathbf{u}(k+N_p-1|k) \\ &= \mathcal{A}_k^{k+N_p-1}\mathbf{x}(k|k) + \left( \mathcal{A}_{k+1}^{k+N_p-1}\mathcal{B}(k), \dots, \mathcal{A}_{k+N_p-1}^{k+N_p-1}\mathcal{B}(k+N_p-2), \mathcal{B}(k+N_p-1) \right) \mathcal{U} \end{aligned}$$

vào công thức hàm phạt  $s(\mathbf{x}(k+N_p|k))$ , thì do  $\mathbf{x}_k = \mathbf{x}(k|k)$  được giả thiết là đã biết thông qua đo đạc bằng cảm biến hoặc quan sát, nên:

$$s(\mathbf{x}(k+N_p|k)) = s(\mathcal{U})$$

sẽ chỉ còn phụ thuộc theo  $\mathcal{U}$ . Bởi vậy khi sử dụng thêm các ký hiệu:

$$\begin{aligned}\mathcal{Y}_{ref} &= \text{col}(\mathbf{y}_{ref}((k+1)T), \mathbf{y}_{ref}((k+2)T), \dots, \mathbf{y}_{ref}((k+N_p)T)) \\ Q &= \text{diag}(q_i) \\ R &= \text{diag}(r_j)\end{aligned}$$

thì hàm mục tiêu (2.5) tương ứng cho hệ sẽ là:

$$\begin{aligned}J(\mathcal{U}) &= \sum_{i=1}^{N_p} q_i |\mathbf{e}_{k+i}(\mathcal{U})|^2 + \sum_{j=0}^{N_p-1} r_j |\mathbf{u}(k+j|k)|^2 + s(\mathbf{x}(k+N_p|k)) \\ &= (\mathcal{Y}_{ref} - \mathcal{Y})^T Q (\mathcal{Y}_{ref} - \mathcal{Y}) + \mathcal{U}^T R \mathcal{U} + s(\mathcal{U}) \\ &= [\mathcal{Y}_{ref} - (M(\mathcal{U})\mathbf{x}_k + N(\mathcal{U})\mathcal{U})]^T Q [\mathcal{Y}_{ref} - (M(\mathcal{U})\mathbf{x}_k + N(\mathcal{U})\mathcal{U})] + \\ &\quad + \mathcal{U}^T R \mathcal{U} + s(\mathcal{U})\end{aligned}\tag{2.21}$$

## 2.2.2. ĐKDB trên nền tối ưu hóa theo sai lệch tín hiệu điều khiển

*Mô hình dự báo theo sai lệch tín hiệu điều khiển*

Như đã nói ở phần trên, tại mục 2.1.2, ta có khá nhiều phương án thực thi bài toán tìm nghiệm tối ưu:

$$\mathcal{U}^* = \arg \min_{\mathcal{U} \in \mathcal{U}} J(\mathcal{U})$$

Song ở dạng bài toán tối ưu mà điều kiện ràng buộc đối với tín hiệu điều khiển  $\mathbf{u}_k$  của hệ song tuyến thường được biểu diễn dưới dạng ràng buộc của sai lệch:

$$\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$$

thì khi đó cần thiết phải viết lại hàm mục tiêu (2.21) theo sai lệch tín hiệu điều khiển  $\Delta \mathbf{u}_k$ .

Để thực hiện được điều này, trước tiên ta cần viết lại mô hình (2.14) đã cho của hệ song tuyến thành:

$$\begin{aligned}
\mathbf{x}_{k+i} &= \mathcal{A}(\mathbf{x}_k)\mathbf{x}_k + \mathcal{B}(\mathbf{x}_k)\mathbf{u}_k = \mathcal{A}(\mathbf{x}_k)\mathbf{x}_k + \mathcal{B}(\mathbf{x}_k)[\mathbf{u}_{k-1} + \Delta\mathbf{u}_k] \\
&= (\mathcal{A}(\mathbf{x}_k), \mathcal{B}(\mathbf{x}_k)) \begin{pmatrix} \mathbf{x}_k \\ \mathbf{u}_{k-1} \end{pmatrix} + \mathcal{B}(\mathbf{x}_k)\Delta\mathbf{u}_k \\
\mathbf{u}_k &= \mathbf{u}_{k-1} + \Delta\mathbf{u}_k = (\Theta, I) \begin{pmatrix} \mathbf{x}_k \\ \mathbf{u}_{k-1} \end{pmatrix} + \Delta\mathbf{u}_k \\
\mathbf{y}_k &= \mathcal{C}(\mathbf{x}_k)\mathbf{x}_k = (\mathcal{C}(\mathbf{x}_k), \Theta) \begin{pmatrix} \mathbf{x}_k \\ \mathbf{u}_{k-1} \end{pmatrix}
\end{aligned}$$

trong đó  $\Theta$  là ma trận có tất cả các phần tử đều bằng 0, và  $I$  là ma trận đơn vị có số chiều  $m \times m$  phù hợp kích thước của  $\mathbf{u}_k$ . Nếu định nghĩa tiếp vector trạng thái mới:

$$\hat{\mathbf{x}}_k = \begin{pmatrix} \mathbf{x}_k \\ \mathbf{u}_{k-1} \end{pmatrix}$$

cũng như các ma trận:

$$\begin{aligned}
\hat{\mathcal{A}}(\hat{\mathbf{x}}_k) &= \begin{pmatrix} \mathcal{A}(\mathbf{x}_k) & \mathcal{B}(\mathbf{x}_k) \\ \Theta & I \end{pmatrix}, \quad \hat{\mathcal{B}}(\hat{\mathbf{x}}_k) = \begin{pmatrix} \mathcal{B}(\mathbf{x}_k) \\ I \end{pmatrix} \\
\hat{\mathcal{C}}(\hat{\mathbf{x}}_k) &= (\mathcal{C}(\mathbf{x}_k), \Theta)
\end{aligned}$$

thì mô hình (2.14) của hệ song tuyến sẽ có dạng tương đương như sau:

$$\begin{cases} \hat{\mathbf{x}}_{k+1} = \hat{\mathcal{A}}(\hat{\mathbf{x}}_k)\hat{\mathbf{x}}_k + \hat{\mathcal{B}}(\hat{\mathbf{x}}_k)\Delta\mathbf{u}_k, & \hat{\mathbf{x}}_k \in \mathbb{R}^{n+m}, \Delta\mathbf{u}_k \in \mathbb{R}^m \\ \mathbf{y}_k = \hat{\mathcal{C}}(\hat{\mathbf{x}}_k)\hat{\mathbf{x}}_k, & \mathbf{y}_k \in \mathbb{R}^m \end{cases} \quad (2.22)$$

nhưng lúc này tín hiệu điều khiển lại là sai lệch  $\Delta\mathbf{u}_k$ .

Cùng với mô hình tương đương (2.22), nhưng bây giờ được viết theo sai lệch tín hiệu điều khiển  $\Delta\mathbf{u}_k$ , ta lại có các công thức dự báo trạng thái  $\hat{\mathbf{x}}(k+i|k)$

và đầu ra  $\hat{\mathbf{y}}(k+i|k)$  hoàn toàn tương tự như (2.17) và (2.19)

$$\begin{aligned}
\hat{\mathbf{x}}(k+i|k) &= \hat{\mathcal{A}}(k+i-1) \cdots \hat{\mathcal{A}}(k+1)\hat{\mathcal{A}}(k)\hat{\mathbf{x}}(k|k) + \\
&\quad + \hat{\mathcal{A}}(k+i-1) \cdots \hat{\mathcal{A}}(k+1)\hat{\mathcal{B}}(k)\Delta\mathbf{u}(k|k) + \\
&\quad + \hat{\mathcal{A}}(k+i-1) \cdots \hat{\mathcal{A}}(k+2)\hat{\mathcal{B}}(k+1)\Delta\mathbf{u}(k+1|k) + \\
&\quad \vdots \\
&\quad + \hat{\mathcal{A}}(k+i-1)\hat{\mathcal{B}}(k+i-2)\mathbf{u}(k+i-2|k) + \hat{\mathcal{B}}(k+i-1)\Delta\mathbf{u}(k+i-1|k) \\
&= \hat{\mathcal{A}}_k^{k+i-1}\hat{\mathbf{x}}(k|k) + \hat{\mathcal{A}}_{k+1}^{k+i-1}\hat{\mathcal{B}}(k)\Delta\mathbf{u}(k|k) + \cdots + \hat{\mathcal{B}}(k+i-1)\Delta\mathbf{u}(k+i-1|k)
\end{aligned} \tag{2.23}$$

với:

$$i=1,2, \dots, N_p, \quad \hat{\mathcal{A}}_i^j = \hat{\mathcal{A}}(k+j)\hat{\mathcal{A}}(k+j-1) \cdots \hat{\mathcal{A}}(k+i) \text{ khi } j \geq i$$

và:

$$\begin{aligned}
\hat{\mathbf{y}}(k+j|k) &= \hat{\mathcal{C}}(k+j) \left[ \hat{\mathcal{A}}_k^{k+j-1}\hat{\mathbf{x}}(k|k) + \hat{\mathcal{A}}_{k+1}^{k+j-1}\hat{\mathcal{B}}(k)\Delta\mathbf{u}(k|k) + \cdots \right. \\
&\quad \left. + \hat{\mathcal{B}}(k+j-1)\Delta\mathbf{u}(k+j-1|k) \right], \quad j=1,2, \dots, N_p
\end{aligned}$$

Từ (2.23), suy ra được dãy giá trị đầu ra dự báo trong toàn bộ cửa sổ dự báo hiện tại là:

$$\hat{\mathcal{Y}} = \hat{M}(\hat{\mathcal{U}})\hat{\mathbf{x}}_k + \hat{N}(\hat{\mathcal{U}})\hat{\mathcal{U}}$$

trong đó:

$$\begin{aligned}
\hat{\mathcal{Y}} &= \text{col}(\hat{\mathbf{y}}(k+1|k), \hat{\mathbf{y}}(k+2|k), \dots, \hat{\mathbf{y}}(k+N_p|k)) \\
\hat{\mathcal{U}} &= \text{col}(\Delta\mathbf{u}(k|k), \Delta\mathbf{u}(k+1|k), \dots, \Delta\mathbf{u}(k+N_p-1|k)) \in \mathbb{R}^{2N_p} \\
\mathbf{x}(k|k) &= \mathbf{x}_k
\end{aligned}$$

và:

$$\begin{aligned} \widehat{M}(\widehat{U}) &= \begin{pmatrix} \widehat{C}(k+1)\widehat{\mathcal{A}}_k^k \\ \widehat{C}(k+1)\widehat{\mathcal{A}}_k^{k+1} \\ \vdots \\ \widehat{C}(k+N_p)\widehat{\mathcal{A}}_k^{k+N_p-1} \end{pmatrix}, \widehat{\mathcal{A}}_i^j = \widehat{\mathcal{A}}(k+j)\widehat{\mathcal{A}}(k+j-1) \cdots \widehat{\mathcal{A}}(k+i) \text{ khi } j \geq i \\ \widehat{N}(\widehat{U}) &= \begin{pmatrix} \widehat{C}(k+1)\widehat{\mathcal{B}}(k) & 0 & \cdots & 0 \\ \widehat{C}(k+2)\widehat{\mathcal{A}}_{k+1}^{k+1}\widehat{\mathcal{B}}(k) & \widehat{C}(k+2)\widehat{\mathcal{B}}(k) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{C}(k+N_p)\widehat{\mathcal{A}}_{k+1}^{k+N_p-1}\widehat{\mathcal{B}}(k) & \widehat{C}(k+N_p)\widehat{\mathcal{A}}_{k+2}^{k+N_p-1}\widehat{\mathcal{B}}(k) & \cdots & \widehat{C}(k+N_p)\widehat{\mathcal{B}}(k+N_p-1) \end{pmatrix} \end{aligned} \quad (2.24)$$

Tương ứng, hàm mục tiêu (2.21) nay cũng được đổi thành:

$$\begin{aligned} J(\widehat{U}) &= \left[ \mathcal{Y}_{ref} - (\widehat{M}(\widehat{U})\mathbf{x}_k + \widehat{N}(\widehat{U})\widehat{U}) \right]^T Q \left[ \mathcal{Y}_{ref} - (\widehat{M}(\widehat{U})\mathbf{x}_k + \widehat{N}(\widehat{U})\widehat{U}) \right] + \\ &\quad + \widehat{U}^T R \widehat{U} + s(\widehat{U}) \end{aligned} \quad (2.25)$$

trong đó:

$$\begin{aligned} \mathcal{Y}_{ref} &= \text{col}(\mathbf{y}_{ref}((k+1)T), \mathbf{y}_{ref}((k+2)T), \dots, \mathbf{y}_{ref}((k+N_p)T)) \\ s(\widehat{U}) &= s(\widehat{\mathbf{x}}(k+N_p|k)) \\ &= s \left[ \widehat{\mathcal{A}}_k^{k+N_p-1} \widehat{\mathbf{x}}_k + (\widehat{\mathcal{A}}_{k+1}^{k+N_p-1} \widehat{\mathcal{B}}(k), \dots, \widehat{\mathcal{A}}_{k+N_p-1}^{k+N_p-1} \widehat{\mathcal{B}}(k+N_p-2), \widehat{\mathcal{B}}(k+N_p-1)) \widehat{U} \right] \end{aligned} \quad (2.26)$$

Như vậy vấn đề còn lại của điều khiển dự báo hệ song tuyến là xác định nghiệm tối ưu  $\widehat{U}^* \in U$  của hàm mục tiêu (2.25), trong đó  $U$  là tập các điều kiện ràng buộc của bài toán điều khiển.

*Các điều kiện ràng buộc*

Một cách tổng quát thì tín hiệu điều khiển  $\widehat{U}^* \in U$  của hàm mục tiêu (2.25) cho hệ song tuyến cần phải thỏa mãn các điều kiện ràng buộc sau:

$$y_l^{\min} \leq y_l(k+i|k) \leq y_l^{\max}, \quad i=1,2, \dots, N_p \quad (2.27)$$

$$u_l^{\min} \leq u_l(k+i-1|k) \leq u_l^{\max}, \quad i=1,2, \dots, N_c \quad (2.28)$$

$$\Delta u_l^{\min} \leq \Delta u_l(k+i-1|k) \leq \Delta u_l^{\max}, \quad i=1,2, \dots, N_c \quad (2.29)$$

trong đó  $l=1,2, \dots, m$  là chỉ số các phần tử của vector  $\mathbf{y}(k+i|k)$ ,  $\mathbf{u}(k+i|k)$  và  $\Delta\mathbf{u}(k+i|k)$  tại thời điểm  $k+i$  trong tương lai thuộc cửa sổ dự báo thứ  $k$ .

Nhưng vì ở bài toán tối ưu:

$$\widehat{\mathcal{U}}^* = \arg \min_{\mathcal{U} \in U} J(\widehat{\mathcal{U}}) \quad (2.30)$$

của điều khiển dự báo, ta chỉ quan tâm tới tập  $U$  gồm các điều kiện ràng buộc về tín hiệu điều khiển:

$$\widehat{\mathcal{U}} = \text{col}(\Delta\mathbf{u}(k|k), \Delta\mathbf{u}(k+1|k), \dots, \Delta\mathbf{u}(k+N_p-1|k))$$

nên cần thiết phải chuyển đổi những điều kiện ràng buộc trạng thái (2.27), (2.28) thành ràng buộc về tín hiệu điều khiển.

Công thức dự báo trạng thái (2.23) sẽ giúp ta chuyển được điều kiện ràng buộc trạng thái thành điều kiện ràng buộc tín hiệu điều khiển  $\widehat{\mathcal{U}}$ . Từ (2.23) ta có:

$$\begin{aligned} \widehat{\mathbf{x}}(k+i|k) &= \widehat{\mathcal{A}}_k^{k+i-1} \widehat{\mathbf{x}}(k|k) + \widehat{\mathcal{A}}_{k+1}^{k+i-1} \widehat{\mathcal{B}}(k) \Delta\mathbf{u}(k|k) + \dots + \widehat{\mathcal{B}}(k+i-1) \Delta\mathbf{u}(k+i-1|k) \\ &= \widehat{\mathcal{A}}_k^{k+i-1} \widehat{\mathbf{x}}(k|k) + \left( \widehat{\mathcal{A}}_{k+1}^{k+i-1} \widehat{\mathcal{B}}(k), \dots, \widehat{\mathcal{B}}(k+i-1), 0, \dots, 0 \right) \widehat{\mathcal{U}} \\ &= \widehat{\mathcal{A}}_k^{k+i-1} \widehat{\mathbf{x}}(k|k) + \mathcal{F}_i \widehat{\mathcal{U}} \end{aligned} \quad (2.31)$$

trong đó:

$$\mathcal{F}_i = \left( \widehat{\mathcal{A}}_{k+1}^{k+i-1} \widehat{\mathcal{B}}(k), \dots, \widehat{\mathcal{B}}(k+i-1), 0, \dots, 0 \right)$$

Ký hiệu  $\widehat{\mathbf{a}}_j^T(i)$ ,  $\widehat{\mathbf{f}}_j^T(i)$ ,  $j=1,2, \dots, n$  là các vector hàng của  $\widehat{\mathcal{A}}_{k+1}^{k+i-1}$  và  $\mathcal{F}_i$  thì từ (2.31) ta có:

$$\widehat{\mathbf{x}}_j(k+i|k) = \widehat{\mathbf{a}}_j^T(i) \widehat{\mathbf{x}}_k + \widehat{\mathbf{f}}_j^T(i) \widehat{\mathcal{U}} \quad (2.32)$$

trong đó  $\widehat{\mathbf{x}}(k|k)$  ở công thức (2.31) nay được viết lại thành  $\widehat{\mathbf{x}}_k$  trong công thức (2.32) để nhấn mạnh rằng đó là giá trị đã có (đo được bằng cảm biến hoặc quan sát được).

Vậy điều kiện ràng buộc (2.27) và (2.28) là tương đương với:

$$y_l^{\min} \leq \widehat{\mathbf{a}}_l^T(i) \widehat{\mathbf{x}}_k + \widehat{\mathbf{f}}_l^T(i) \widehat{\mathcal{U}} \leq y_l^{\max}, \quad l=1,2, \dots, m \quad \text{và} \quad i=1,2, \dots, N_p$$

và:

$$u_l^{\min} \leq \widehat{\mathbf{a}}_l^T(i)\widehat{\mathbf{x}}_k + \widehat{\mathbf{f}}_l^T(i)\widehat{\mathcal{U}} \leq u_l^{\max}, \quad l=1,2 \dots, m \quad \text{và} \quad i=1,2, \dots, N_c$$

Suy ra bài toán tối ưu (2.30) có tập ràng buộc  $U$  là:

$$U = \left\{ \widehat{\mathcal{U}} \in \mathbb{R}^{2N_p} \mid \begin{aligned} &\Delta u_l^{\min} \leq \Delta u_l(k+i-1|k) \leq \Delta u_l^{\max}, \quad l=1, \dots, m \quad \text{và} \quad i=1, \dots, N_p \\ &y_l^{\min} \leq \widehat{\mathbf{a}}_l^T(i)\widehat{\mathbf{x}}_k + \widehat{\mathbf{f}}_l^T(i)\widehat{\mathcal{U}} \leq y_l^{\max}, \quad l=1, \dots, m \quad \text{và} \quad i=1, \dots, N_p \\ &u_l^{\min} \leq \widehat{\mathbf{a}}_l^T(i)\widehat{\mathbf{x}}_k + \widehat{\mathbf{f}}_l^T(i)\widehat{\mathcal{U}} \leq u_l^{\max}, \quad l=1, \dots, m \quad \text{và} \quad i=1, \dots, N_c \end{aligned} \right\} \quad (2.33)$$

*Thuật toán điều khiển hệ song tuyến*

Với mô hình hệ song tuyến gồm các ma trận  $A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x})$  như đã đưa ra trong công thức (2.12), thì khi chuyển sang mô hình không liên tục bằng cách sử dụng phép xấp xỉ (2.13) với  $T$  là chu kỳ trích mẫu và cũng là khoảng dịch chuyển cửa cửa sổ dự báo, cũng như  $\mathbf{x}_k = \mathbf{x}(kT)$ ,  $\mathbf{u}_k = \mathbf{u}(kT)$ ,  $\mathbf{y}_k = \mathbf{y}(kT)$ , ta sẽ được mô hình không liên tục tương ứng (2.14) với quan hệ giữa các ma trận  $A(\mathbf{x}), B(\mathbf{x}), C(\mathbf{x})$  và  $\mathcal{A}(\mathbf{x}_k), \mathcal{B}(\mathbf{x}_k), \mathcal{C}(\mathbf{x}_k)$  cho bởi (2.15).

Vậy, phương pháp điều khiển hệ song tuyến trên nền quy hoạch phi tuyến có các bước được trình bày ở thuật toán sau:

**Thuật toán 2.1:** *Điều khiển dự báo phản hồi trạng thái hệ song tuyến bám theo tín hiệu đầu ra mẫu với cửa sổ dự báo hữu hạn.*

1. Chọn hàm phạt  $s(\widehat{\mathbf{x}}(k+N_p|k))$ , cửa sổ dự báo  $N_p$ , cửa sổ điều khiển  $N_c$  và hai ma trận trọng số  $Q, R$  đối xứng xác định dương. Chọn chu kỳ trích mẫu  $T$ . Gán  $k=0$  và  $\mathbf{u}_{-1} = (0,0)^T$ .
2. Đo  $\mathbf{x}_k = \mathbf{x}(k|k)$ . Xác định  $\widehat{\mathbf{x}}_k = \text{col}(\mathbf{x}_k, \mathbf{u}_{k-1})$ , các ma trận  $\widehat{\mathcal{A}}(\widehat{\mathbf{x}}_k)$ ,  $\widehat{\mathcal{B}}(\widehat{\mathbf{x}}_k)$ ,  $\widehat{\mathcal{C}}(\widehat{\mathbf{x}}_k)$  từ mô hình không liên tục (2.14) của hệ song tuyến theo [51],  $\widehat{M}(\widehat{\mathcal{U}})$ ,  $\widehat{N}(\widehat{\mathcal{U}})$  theo (2.24) và  $s(\widehat{\mathcal{U}})$ ,  $\mathcal{Y}_{ref}$  theo (2.26).
3. Xây dựng hàm mục tiêu  $J(\widehat{\mathcal{U}})$  theo (2.25) và tập ràng buộc  $U$  theo (2.33).

4. Tìm nghiệm  $\widehat{U}^*$  của bài toán tối ưu (2.30) nhờ các phương pháp quy hoạch phi tuyến, chẳng hạn như phương pháp SQP hoặc interior point [5].
5. Đưa  $\mathbf{u}_k = \mathbf{u}_{k-1} + (I, 0, \dots, 0)\widehat{U}^*$  vào điều khiển hệ song tuyến trong khoảng thời gian  $kT \leq t < (k+1)T$ , trong đó  $I$  là ma trận đơn vị. Gán  $k := k+1$  và quay về 2.

Sẽ có khá nhiều phương án khác nhau phục vụ việc cài đặt thuật toán trên và chúng được phân biệt với nhau ở việc lựa chọn phương pháp quy hoạch phi tuyến cụ thể để tìm nghiệm tối ưu  $\widehat{U}^*$  cho bài toán tối ưu có ràng buộc  $U$  (2.25), tức là thực hiện bước thứ 4 của thuật toán trên. Đây là bài toán tối ưu phi tuyến có ràng buộc, nên những phương pháp tương thích sẽ là SQP, hình chiếu gradient, kỹ thuật hàm chặn, hàm phạt, giải thuật di truyền... Tuy nhiên, sau này luận án sẽ chỉ nhất quán chỉ sử dụng SQP.

### 2.3. Kết luận chương 2

Trong chương 2 luận án đã trình bày phương pháp điều khiển dự báo cho hệ phi tuyến cũng như những kỹ thuật cài đặt bộ điều khiển này. Chi tiết hơn nữa, luận án cũng đã thể hiện cụ thể phương pháp điều khiển dự báo cho hệ song tuyến dưới dạng **thuật toán 2.1**. Ngoài ra, trong chương 2 luận án cũng đã xây dựng được mô hình dự báo (2.22) có khả năng tích hợp được điều kiện ràng buộc về số gia (vận tốc) của tín hiệu điều khiển.

Bộ điều khiển dự báo được trình bày ở chương 2 có những ưu nhược điểm sau:

- Bộ điều khiển được xây dựng từ mô hình không liên tục của đối tượng điều khiển.
- Chỉ sử dụng được với cửa sổ dự báo hữu hạn, do đó để đảm bảo được chất lượng ổn định hoặc bám ổn định theo giá trị đặt trước cần phải chọn được một hàm phạt thích hợp.
- Dễ dàng xử lý được các điều kiện ràng buộc kèm theo, kể cả các điều kiện ràng buộc về giá trị tín hiệu điều khiển, về số gia của tín hiệu điều khiển và về biên trạng thái của hệ thống.



### CHƯƠNG 3

## ĐỀ XUẤT MỘT PHƯƠNG PHÁP MỚI ĐỂ ĐIỀU KHIỂN DỰ BÁO HỆ PHI TUYẾN LIÊN TỤC TRÊN NỀN BIẾN PHÂN

Kết luận của chương 2 đã cho thấy ưu nhược điểm của phương pháp điều khiển dự báo áp dụng cho hệ phi tuyến. Bên cạnh khả năng dễ xử lý các điều kiện ràng buộc nhờ việc áp dụng phương pháp quy hoạch phi tuyến thì một số nhược điểm cố hữu của nó là:

- Chỉ xây dựng được trên mô hình không liên tục của hệ, trong khi đại đa số đối tượng điều khiển đều mô tả bởi mô hình liên tục. Tất nhiên từ mô hình liên tục ta cũng có thể có được mô hình không liên tục tương ứng bằng cách sử dụng phép xấp xỉ:

$$\dot{\mathbf{x}} \approx \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{T_a} \quad \text{Equation Section (Next)(3.1)}$$

cho phép tính vi phân, với  $\mathbf{x}_k = \mathbf{x}(kT_a)$  và  $T_a$  là chu kỳ trích mẫu tín hiệu, song việc xấp xỉ này tất nhiên sẽ kéo theo một sai lệch nhỏ trong mô hình không liên tục thu được, do đó phần nào cũng sẽ ảnh hưởng tới chất lượng điều khiển.

- Cửa sổ dự báo phải luôn là một số  $N$  hữu hạn. Điều này làm cho nghiệm  $\mathcal{U}^*$  của bài toán tối ưu (2.8) tìm được nhờ các phương pháp quy hoạch phi tuyến chưa chắc đã là nghiệm toàn cục và bộ điều khiển dự báo cũng chưa chắc đã làm hệ ổn định, hay bám ổn định theo quỹ đạo mẫu cho trước, đặc biệt là với hệ phi tuyến. Giải pháp khắc phục thường sử dụng ở đây là sử dụng hàm phạt  $s(\mathcal{U})$ . Song việc chọn hàm phạt như thế nào cho bài toán điều khiển dự báo phi tuyến thì cho tới tận bây giờ vẫn chưa có câu trả lời.

Từ đây, để khắc phục hai nhược điểm cố hữu trên của điều khiển dự báo xây dựng trên nền các phương pháp quy hoạch phi tuyến, trong chương này luận án sẽ đề xuất một phương pháp điều khiển dự báo mới, không sử dụng phương pháp quy hoạch phi tuyến cho việc tối ưu hóa mà thay vào đó là phương pháp biến phân (variation) của điều khiển tối ưu, và hơn thế nữa

phương pháp đề xuất này của luận án còn áp dụng được trực tiếp cho mô hình liên tục của đối tượng điều khiển:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}) \end{cases} \quad (3.2)$$

với cửa sổ dự báo là vô hạn, thay vì hữu hạn. Điều này luôn đảm bảo được tính ổn định, hay tính bám ổn định theo quỹ đạo mẫu cho trước, mà không cần sử dụng hàm phạt.

### 3.1. Nội dung cơ bản của phương pháp biến phân

Bài toán điều khiển tối ưu cho đối tượng điều khiển mô tả bởi mô hình liên tục (3.2) được hiểu là phải xác định được tín hiệu điều khiển tối ưu  $\mathbf{u}^*(t)$ ,  $0 \leq t \leq T$  thỏa mãn điều kiện ràng buộc  $\mathbf{u} \in U$  để đưa hệ đi từ điểm trạng thái đầu  $\mathbf{x}_0 = \mathbf{x}(0)$  tới điểm trạng thái cuối  $\mathbf{x}_T = \mathbf{x}(T)$  trong khoảng thời gian  $T$ , gọi là khoảng thời gian xảy ra quá trình tối ưu, sao cho chi phí (cost) của quá trình chuyển đổi trạng thái đó, tính theo:

$$J(\mathbf{u}) = \int_0^T \mathbf{g}(\mathbf{x}, \mathbf{u}) dt \quad (3.3)$$

đạt giá trị nhỏ nhất. Hàm chi phí (3.3) thường được gọi là hàm mục tiêu của bài toán điều khiển tối ưu.

Bài toán điều khiển tối ưu trên còn được phân chia thành nhiều bài toán con. Nguyên tắc của sự phân loại này là:

- Điểm trạng thái đầu hoặc cuối  $\mathbf{x}_0$ ,  $\mathbf{x}_T$  là cho trước hoặc là những điểm trạng thái bất kỳ.
- Khoảng thời gian xảy ra quá trình tối ưu  $T$  là cố định cho trước hoặc không cho trước.
- Điều kiện ràng buộc  $U$  là tập hở hoặc tập đóng.

Với bài toán điều khiển tối ưu hệ liên tục (3.2) mà ở đó có  $\mathbf{x}_0$  cho trước,  $T$  cũng cho trước và  $U$  là tập hở (hoặc trùng với toàn bộ không gian điều khiển, tức là bài toán không bị ràng buộc) thì phương pháp thích hợp nhất là phương pháp biến phân. Hơn thế nữa, trong trường hợp điểm trạng thái cuối

$\mathbf{x}_T$  là bất kỳ thì nghiệm tìm được theo phương pháp biến phân sẽ có dạng phụ thuộc trạng thái  $\mathbf{u}^*(\mathbf{x})$ ,  $0 \leq t \leq T$ , hay nó cũng chính là một bộ điều khiển phản hồi trạng thái tối ưu. Bởi vậy, để phân biệt với nghiệm tối ưu phụ thuộc thời gian  $\mathbf{u}^*(t)$ , ở trường hợp này người ta thường gọi  $\mathbf{u}^*(\mathbf{x})$  là nghiệm tối ưu on-line (trực tuyến).

### 3.1.1. Nguyên lý biến phân

Nguyên lý biến phân được phát biểu như sau [5]: Nếu  $\mathbf{u}^*$  là nghiệm bài toán tối ưu có  $\mathbf{x}_0$  cho trước,  $T$  cũng cho trước và  $U$  là tập hở, thì nghiệm đó phải thỏa mãn:

$$\left. \frac{\partial H}{\partial \mathbf{u}} \right|_{\mathbf{u}^*} = \mathbf{0}^T \quad (3.4)$$

(đạo hàm tại điểm tối ưu) trong đó:

- $\partial/\partial \mathbf{u}$  là ký hiệu đạo hàm Jacobi của một hàm nhiều biến.
- $\mathbf{0}^T = (0, \dots, 0)$
- $H = \mathbf{p}^T \mathbf{f}(\mathbf{x}, \mathbf{u}) - g(\mathbf{x}, \mathbf{u})$ , có tên là hàm Hamilton, với  $\mathbf{p}$  là vector biên đồng trạng thái (costates), thỏa mãn quan hệ Euler - Lagrange:

$$\dot{\mathbf{p}} = - \left( \frac{\partial H}{\partial \mathbf{x}} \right)^T \quad (3.5)$$

và điều kiện biên  $\mathbf{p}(T) = \mathbf{0}$  khi điểm trạng thái cuối là bất kỳ.

Dựa vào nguyên lý biến phân trên, nghiệm  $\mathbf{u}^*$  của bài toán tối ưu liên tục sẽ được xác định qua các bước như sau [5]:

1. Từ điều kiện (3.4) xác định quan hệ phải có  $\mathbf{u}(\mathbf{x}, \mathbf{p})$  cho tín hiệu điều khiển tối ưu.
2. Thay quan hệ tìm được này vào hệ đã cho  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$  và phương trình Euler-Lagrange (3.5) sẽ được hệ phương trình vi phân bậc nhất của hai biến số  $\mathbf{x}$ ,  $\mathbf{p}$  và tìm nghiệm hệ phương trình vi phân đó ứng với các điều kiện biên  $\mathbf{x}_0 = \mathbf{x}(0)$ ,  $\mathbf{x}_T = \mathbf{x}(T)$  nếu điểm cuối là cho trước hoặc  $\mathbf{p}(T) = \mathbf{0}$  nếu điểm cuối là bất kỳ.

3. Thay ngược nghiệm  $x(t)$ ,  $p(t)$  tìm được vào quan hệ  $u(x, p)$  đã có từ bước 1 để được nghiệm tối ưu  $u^*(t)$  của bài toán.

### 3.1.2. Bộ điều khiển LQR (Linear Quadratic Regulator)

Có thể thấy việc áp dụng nguyên lý 3 bước nêu trên của phương pháp biến phân là hoàn toàn không đơn giản cho hệ phi tuyến vì cho tới nay ta vẫn chưa có được phương pháp tìm nghiệm tường minh của hệ phương trình vi phân phi tuyến (bước 2). Bởi vậy người ta thường chỉ áp dụng cho bài toán có hệ (3.2) ở dạng tuyến tính tham số hằng (linear):

$$\dot{x} = Ax + Bu \quad (3.6)$$

có  $T = \infty$  (khoảng thời gian xảy ra quá trình tối ưu cho trước là vô cùng), hàm mục tiêu (3.3) ở dạng toàn phương (quadratic):

$$J(u) = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (3.7)$$

và điểm cuối  $x_T$  là bất kỳ, trong đó  $Q$  là ma trận đối xứng bán xác định dương ( $Q = Q^T \geq 0$ ),  $R$  là ma trận đối xứng xác định dương ( $R = R^T > 0$ ) cho trước.

Ở trường hợp đặc biệt này, nghiệm tối ưu  $u^*$  tìm được theo phương pháp biến phân sẽ có dạng on - line [5]:

$$u^* = -R^{-1} B^T L x = -R_{LQR} x \quad \text{với } R_{LQR} = R^{-1} B^T L \quad (3.8)$$

trong đó  $L$  là nghiệm đối xứng bán xác định dương của phương trình đại số Riccati:

$$L B R^{-1} B^T L - A^T L - L A = Q \quad (3.9)$$

Lúc này  $R_{LQR}$  cho bởi công thức (3.8) sẽ được gọi là bộ điều khiển (regulator) phản hồi trạng thái tối ưu.

### 3.1.3. Điều kiện đủ cho tính ổn định của hệ LQR

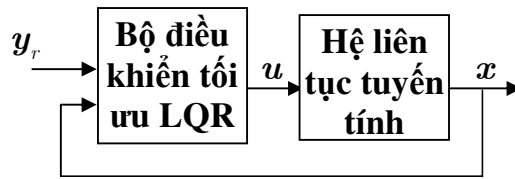
Hệ LQR được hiểu là hệ kín gồm đối tượng tuyến tính tham số hằng (3.6) và bộ điều khiển phản hồi trạng thái (3.8), như được mô tả ở hình 3.1. Do bài toán tối ưu này có điểm cuối  $x_T$  là bất kỳ nên chưa thể khẳng định

được hệ LQR đó là đã ổn định, tức là chưa thể khẳng định được là ở hệ LQR cũng sẽ có:

$$\mathbf{x}_T = \lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{0}$$

Tuy nhiên, nếu một trong các điều kiện nêu sau đây được thỏa mãn (điều kiện đủ), thì ta luôn khẳng định được hệ LQR là ổn định [5]:

- Bài toán có  $Q = Q^T > 0$ , tức là ma trận  $Q$  là xác định dương chứ không chỉ là bán xác định dương.
- Nghiệm  $L$  tìm được của phương trình Riccati (3.9) là xác định dương (chứ không chỉ là bán xác định dương)
- Cặp ma trận  $(A, Q)$  là quan sát được.



Hình 3.1. Hệ kín với bộ điều khiển phản hồi trạng thái tối ưu LQR

### 3.1.4. Áp dụng nguyên tắc điều khiển LQR để điều khiển tối ưu hệ tuyến tính bám ổn định theo giá trị đầu ra cho trước

Để tạo được khả năng sử dụng bộ điều khiển LQR như trên vào bài toán điều khiển dự báo hệ song tuyến bám theo được giá trị đầu ra cho trước, sẽ được trình bày ngay ở mục 3.2 sau đây, trong mục nhỏ này luận án sẽ tiến hành biến đổi một chút ít bộ điều khiển LQR (3.8) để có thể áp dụng được vào bài toán điều khiển tối ưu hệ tuyến tính tham số hằng:

$$\begin{cases} \dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \\ \mathbf{y} = C\mathbf{x} + D\mathbf{u} \end{cases} \quad (3.10)$$

sao cho đầu ra  $\mathbf{y}$  của nó bám theo được giá trị đầu ra mẫu  $\mathbf{y}_r$  cho trước. Sau đây bài toán này sẽ được luận án gọi là bài toán điều khiển bám tối ưu.

Trước tiên, do không phải bài toán điều khiển bám nào cũng có nghiệm, nên ta cần có các giả thiết sau cho bài toán điều khiển bám tối ưu:

- Bài toán bám tối ưu hệ tuyến tính tham số hằng (3.10) có nghiệm  $\mathbf{u}_e$  ở chế độ xác lập, trong đó ký hiệu chỉ số  $e$  để nói rằng đó là tín hiệu mà với nó có được  $\mathbf{y} \rightarrow \mathbf{y}_r$ ,
- Khi hệ đã bám theo được giá trị mẫu  $\mathbf{y}_r$ , tức là khi đã có  $\mathbf{y} = \mathbf{y}_r$ , thì hệ sẽ xác lập với trạng thái xác lập là  $\mathbf{x}_e$ .

Với hai giả thiết nêu trên, hiển nhiên phải có:

$$\begin{cases} \mathbf{0} = \dot{\mathbf{x}}_e = A\mathbf{x}_e + B\mathbf{u}_e \\ \mathbf{y}_r = C\mathbf{x}_e + D\mathbf{u}_e \end{cases} \quad (3.11)$$

và điều này tương đương với:

$$\begin{pmatrix} \mathbf{0} \\ \mathbf{y}_r \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \mathbf{x}_e \\ \mathbf{u}_e \end{pmatrix} \Leftrightarrow \begin{pmatrix} \mathbf{x}_e \\ \mathbf{u}_e \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{y}_r \end{pmatrix} \quad (3.12)$$

Tiếp theo ta đặt biến mới:

$$\boldsymbol{\delta} = \mathbf{x} - \mathbf{x}_e \quad \text{và} \quad \boldsymbol{\rho} = \mathbf{u} - \mathbf{u}_e$$

thì khi trừ từng vế của (3.10) và (3.11) cho nhau, sẽ được (gọi là hệ sai số):

$$\dot{\boldsymbol{\delta}} = A\boldsymbol{\delta} + B\boldsymbol{\rho} \quad (3.13)$$

và bài toán điều khiển bám theo giá trị đặt  $\mathbf{y}_r$  ở đầu ra cho hệ tuyến tính tham số hằng ban đầu là (3.10) nay đã trở thành bài toán điều khiển ổn định cho hệ sai số (3.13).

Áp dụng phương pháp điều khiển LQR cho hệ sai số ứng với hàm mục tiêu:

$$J(\boldsymbol{\rho}) = \int_0^{\infty} (\boldsymbol{\delta}^T Q \boldsymbol{\delta} + \boldsymbol{\rho}^T R \boldsymbol{\rho}) dt \quad (3.14)$$

có  $Q, R$  đều là hai ma trận đối xứng xác định dương, ta sẽ được:

$$\boldsymbol{\rho}^* = -R^{-1} B^T L \boldsymbol{\delta} \quad \text{với} \quad R_{LQR} = R^{-1} B^T L \quad (3.15)$$

trong đó  $L = L^T > 0$  là nghiệm đối xứng xác định dương của phương trình Riccati (3.9). Tất nhiên bộ điều khiển LQR (3.15) này sẽ làm ổn định hệ sai số (3.13), vì ở đây có  $Q$  là ma trận xác định dương.

Từ bộ điều khiển LQR (3.15) của hệ sai số (3.13) ta cũng suy ra được bộ điều khiển bám tối ưu theo giá trị đầu ra đặt trước  $\mathbf{y}_r$  cho hệ tuyến tính tham số hằng ban đầu (3.10) như sau:

$$\mathbf{u}^* = \mathbf{u}_e - R^{-1}B^T L(\mathbf{x} - \mathbf{x}_e) \quad (3.16)$$

### 3.2. Phương pháp đề xuất để điều khiển dự báo với cửa sổ dự báo vô hạn cho hệ song tuyến liên tục không dừng, bám theo được giá trị đầu ra cho trước

Dựa trên phương pháp điều khiển xây dựng trên nền biến phân đã được luận án giới thiệu ở mục 3.1.4, để điều khiển bám ổn định hệ tuyến tính tham số hằng mô tả bởi (3.10), sau đây luận án sẽ mở rộng tiếp thành một đề xuất phương pháp điều khiển dự báo cho bài toán điều khiển bám ổn định theo quỹ đạo tín hiệu đầu ra mẫu cho hệ song tuyến liên tục, thay vì chỉ điều khiển ổn định, và sử dụng trực tiếp mô hình liên tục đó trong mô hình dự báo, thay vì sử dụng mô hình xấp xỉ không liên tục của nó.

#### 3.2.1. Tư tưởng chính của phương pháp

Xét hệ song tuyến MIMO, không dừng, có số tín hiệu đầu vào bằng số các tín hiệu đầu ra, mô tả bởi mô hình liên tục:

$$\begin{cases} \dot{\mathbf{x}} = A(\mathbf{x}, t)\mathbf{x} + B(\mathbf{x}, t)\mathbf{u} \\ \mathbf{y} = C(\mathbf{x}, t)\mathbf{x} + D(\mathbf{x}, t)\mathbf{u} \end{cases} \quad (3.17)$$

trong đó  $\mathbf{u} \in \mathbb{R}^m$  là vector của  $m$  tín hiệu đầu vào,  $\mathbf{y} \in \mathbb{R}^m$  là vector của  $m$  các tín hiệu đầu ra và  $\mathbf{x} \in \mathbb{R}^n$  là vector của  $n$  biến trạng thái trong hệ. Các ma trận  $A(\mathbf{x}, t)$ ,  $B(\mathbf{x}, t)$ ,  $C(\mathbf{x}, t)$  và  $D(\mathbf{x}, t)$  đều có những phần tử là hàm số phụ thuộc biến trạng thái  $\mathbf{x}$  cũng như thời gian  $t$ . Bài toán điều khiển đặt ra ở đây là xác định bộ điều khiển phản hồi trạng thái sao cho tín hiệu đầu ra  $\mathbf{y}$  của hệ bám theo được giá trị đầu ra mong muốn  $\mathbf{y}_r$  cho trước.

Giả sử tất cả các ma trận  $A(\mathbf{x}, t)$ ,  $B(\mathbf{x}, t)$ ,  $C(\mathbf{x}, t)$ ,  $D(\mathbf{x}, t)$  là liên tục theo  $\mathbf{x}$  và  $t$ . Khi đó, ở thời điểm  $t_k$  hiện tại và trong khoảng thời gian  $t_k \leq t < t_k + T_k$  đủ nhỏ, hệ song tuyến (3.17) sẽ xấp xỉ được bởi mô hình tuyến tính tham số hằng:

$$H_k : \begin{cases} \dot{\mathbf{x}} = A_k \mathbf{x} + B_k \mathbf{u} \\ \mathbf{y} = C_k \mathbf{x} + D_k \mathbf{u} \end{cases} \quad (3.18)$$

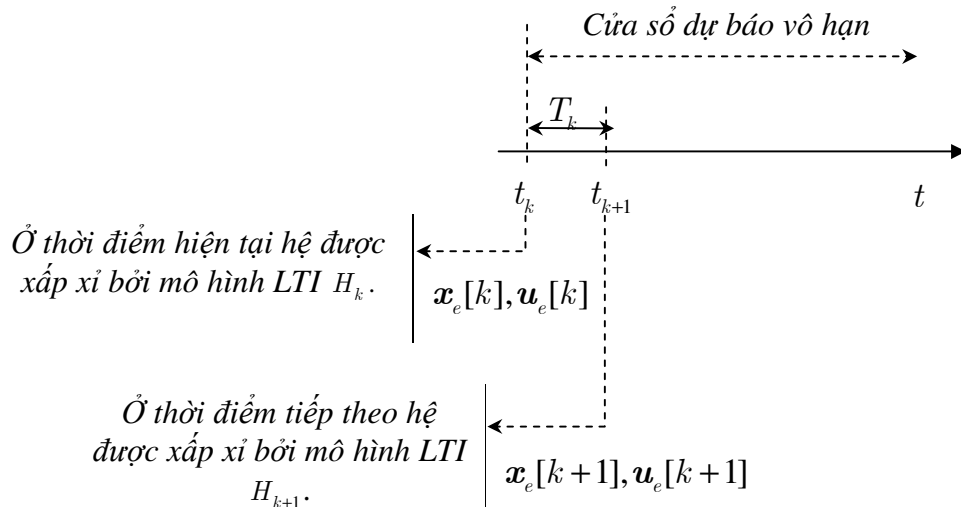
trong đó:

$$A(\mathbf{x}, t) \approx A_k, \quad B(\mathbf{x}, t) \approx B_k, \quad C(\mathbf{x}, t) \approx C_k, \quad D(\mathbf{x}, t) \approx D_k \quad \text{khi } t_k \leq t < t_k + T_k \quad (3.19)$$

Việc xấp xỉ trên là hoàn toàn chấp nhận được do từ giả thiết về tính liên tục của các ma trận tham số mô hình (3.17) luôn có:

$$\lim_{T_k \rightarrow 0} A(\mathbf{x}, t) = A_k, \quad \lim_{T_k \rightarrow 0} B(\mathbf{x}, t) = B_k, \quad \lim_{T_k \rightarrow 0} C(\mathbf{x}, t) = C_k, \quad \lim_{T_k \rightarrow 0} D(\mathbf{x}, t) = D_k$$

và  $T_k$  là thời gian tính toán cần thiết cho một vòng lặp của bộ điều khiển dự báo, nên rất nhỏ đối với card DSP1103 mà tác giả sử dụng khi tiến hành thực nghiệm thì  $T_k = 70ns$ . Nó cũng sẽ chính là khoảng dịch chuyển của cửa sổ dự báo.



Hình 3.2. Mô tả tư tưởng của phương pháp

Nhờ việc xấp xỉ mô hình song tuyến liên tục không dừng (3.17) thành vô số mô hình LTI (3.18) ứng với  $k = 0, 1, \dots$  nên bài toán điều khiển bám giá trị đầu ra mẫu  $\mathbf{y}_r$  của hệ song tuyến (3.17) trở thành bài toán bám tín hiệu ra mẫu  $\mathbf{y}_r$  cho tất cả vô số hệ LTI (3.18). Và như vậy, các bước điều khiển trong một vòng lặp sẽ là:



### Tư tưởng của phương pháp được đề xuất:

1. Tại thời điểm hiện tại  $t_k$  đo giá trị  $\mathbf{x}(t_k) = \mathbf{x}_k$  và xác định các ma trận hằng của mô hình LTI (3.18) gồm  $A_k, B_k, C_k, D_k$  theo các công thức:

$$A_k = A(\mathbf{x}_k, t_k), B_k = B(\mathbf{x}_k, t_k), C_k = C(\mathbf{x}_k, t_k), D_k = D(\mathbf{x}_k, t_k) \quad (3.20)$$

2. Xác định tín hiệu điều khiển  $\mathbf{u}(t)$  để hệ LTI (3.18) bám theo được giá trị tín hiệu đầu ra mẫu  $\mathbf{y}_r$ .
3. Đưa  $\mathbf{u}(t)$  vừa tìm được vào điều khiển hệ song tuyến (3.17) rồi quay về bước 1 để thực hiện vòng lặp mới tại thời điểm tiếp theo là  $t_{k+1}$

### 3.2.2. Xây dựng thuật toán điều khiển

Như vậy, ở thời điểm hiện tại  $t_k$  hệ song tuyến liên tục không dừng (3.17) đã được xấp xỉ bởi mô hình LTI (3.18) và tất nhiên mô hình xấp xỉ này cũng sẽ chỉ được sử dụng trong khoảng thời gian  $t_k \leq t < t_{k+1}$  của một vòng lặp, như mô tả ở hình 3.2, trong đó  $t_{k+1} = t_k + T_k$  với  $T_k$  là thời gian tính toán cần thiết cho một vòng lặp. Như vậy  $T_k$  sẽ là vô cùng nhỏ và việc xấp xỉ mô hình song tuyến (3.17) với tính liên tục của các ma trận  $A(\mathbf{x}, t), B(\mathbf{x}, t), C(\mathbf{x}, t), D(\mathbf{x}, t)$  thành mô hình LTI (3.18) trong khoảng thời gian  $t_k \leq t < t_{k+1}$  là hoàn toàn chấp nhận được.

Gọi  $\mathbf{y}_r$  là tín hiệu đầu ra mẫu mà hệ LTI (3.18) cần phải bám theo. Đây cũng chính là bài toán đã được trình bày ở mục 3.1.4 trước đây. Do đó nếu ký hiệu  $\mathbf{x}_e[k], \mathbf{u}_e[k]$  là giá trị trạng thái và điều khiển của hệ LTI (3.18) ở chế độ xác lập, tức là khi hệ đã bám theo được giá trị đầu ra mẫu  $\mathbf{y}_r$ , thì những giá trị xác lập này phải thỏa mãn:

$$\begin{cases} \mathbf{0} = A_k \mathbf{x}_e[k] + B_k \mathbf{u}_e[k] \\ \mathbf{y}_r = C_k \mathbf{x}_e[k] + D_k \mathbf{u}_e[k] \end{cases} \quad (3.21)$$

Từ đây ta suy ra được hệ gồm  $n + m$  phương trình cho  $n + m$  ẩn số  $(\mathbf{x}_e[k], \mathbf{u}_e[k])$  như sau:

$$\begin{pmatrix} A_k \mathbf{x}_e[k] + B_k \mathbf{u}_e[k] \\ C_k \mathbf{x}_e[k] + D_k \mathbf{u}_e[k] \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{y}_r \end{pmatrix} \Leftrightarrow \begin{pmatrix} A_k & B_k \\ C_k & D_k \end{pmatrix} \begin{pmatrix} \mathbf{x}_e[k] \\ \mathbf{u}_e[k] \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{y}_r \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} \mathbf{x}_e[k] \\ \mathbf{u}_e[k] \end{pmatrix} = \begin{pmatrix} A_k & B_k \\ C_k & D_k \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{y}_r \end{pmatrix} \quad (3.22)$$

và đó cũng là hệ phương trình cho phép xác định các giá trị xác lập  $(\mathbf{x}_e[k], \mathbf{u}_e[k])$  của hệ từ tín hiệu đầu ra mẫu  $\mathbf{y}_r$ .

Lại đặt các biến mới giống như đã làm ở mục 3.1.4:

$$\boldsymbol{\delta} = \mathbf{x} - \mathbf{x}_e[k] \text{ và } \boldsymbol{\rho} = \mathbf{u} - \mathbf{u}_e[k]$$

thì do  $(\mathbf{x}_e[k], \mathbf{u}_e[k])$  là những vector hằng số, ta sẽ có từ (3.18) và (3.21) mô hình hệ sai số như sau:

$$\dot{\boldsymbol{\delta}} = A_k \boldsymbol{\delta} + B_k \boldsymbol{\rho} \quad (3.23)$$

Như vậy, bài toán điều khiển hệ (3.18) bám theo giá trị đầu ra mẫu  $\mathbf{y}_r$  đã được chuyển thành bài toán điều khiển ổn định hệ sai số (3.23). Để thiết kế bộ điều khiển dự báo cho hệ tuyến tính liên tục (3.23) có được chất lượng ổn định tối ưu, ta sẽ áp dụng phương pháp điều khiển tối ưu LQR cho bước dịch chuyển cửa sổ dự báo thứ  $k$  với độ rộng của cửa sổ dự báo là vô hạn, được đánh giá bởi hàm mục tiêu mô tả sai số dự báo ở thời tương lai như sau:

$$J_k(\boldsymbol{\rho}) = \frac{1}{2} \int_{t_k}^{\infty} (\boldsymbol{\delta}^T Q_k \boldsymbol{\delta} + \boldsymbol{\rho}^T R_k \boldsymbol{\rho}) dt \rightarrow \min \quad (3.24)$$

trong đó  $Q_k, R_k$  là hai ma trận đối xứng xác định dương tùy chọn, thay đổi được theo từng bước dịch chuyển của cửa sổ dự báo. Việc tùy chọn hai ma trận  $Q_k, R_k$  này sẽ là một cơ hội để sau này thuật toán còn có khả năng thỏa mãn thêm điều kiện ràng buộc.

Áp dụng phương pháp biến phân để tìm nghiệm  $\boldsymbol{\rho}^*$  của bài toán tối ưu ở quá trình quá độ, mô tả bởi (3.23), (3.24), ta sẽ có (xem lại mục 3.1.2):

$$L_k B_k R_k^{-1} B_k^T L_k - L_k A_k - A_k^T L_k = Q_k \quad (3.25)$$

$$\text{và } \boldsymbol{\rho}^* = -R_k^{-1} B_k^T L_k \boldsymbol{\delta} = -R_k^{-1} B_k^T L_k (\mathbf{x} - \mathbf{x}_e[k])$$

Từ đây ta cũng đến được tín hiệu điều khiển cho hệ song tuyến liên tục và không dừng (3.17) ban đầu tại cửa sổ dự báo hiện tại:

$$\begin{aligned} \mathbf{u}^*(t) &= \mathbf{u}_e[k] + \boldsymbol{\rho}^*(t) \\ &= \mathbf{u}_e[k] - R_k^{-1} B_k^T L_k (\mathbf{x} - \mathbf{x}_e[k]) \end{aligned} \quad (3.26)$$

và tất nhiên tín hiệu điều khiển này chỉ sử dụng được trong đúng khoảng thời gian hiện tại là  $t_k \leq t < t_{k+1}$ .

Tổng kết lại thì bộ điều khiển dự báo với cửa sổ dự báo vô hạn, áp dụng cho hệ song tuyến liên tục không dừng (3.17), làm việc theo thuật toán đề xuất ở trên sẽ gồm các bước lặp như sau:

**Thuật toán 3.1:** *Điều khiển dự báo phản hồi trạng thái để tín hiệu ra bám theo tín hiệu đầu ra mẫu cho hệ song tuyến liên tục với cửa sổ dự báo vô hạn.*

1. Chọn quy luật thay đổi các ma trận trọng số  $Q_k, R_k$  đối xứng xác định dương. Gán  $t_0 = 0$  và  $k = 0$ .
2. Đo  $\mathbf{x}_k = \mathbf{x}(t_k)$  và tính xấp xỉ các ma trận hằng  $A_k, B_k, C_k, D_k$  của mô hình LTI (3.18) từ  $A(\mathbf{x}, t), B(\mathbf{x}, t), C(\mathbf{x}, t), D(\mathbf{x}, t)$  theo công thức (3.20).
3. Xác định  $(\mathbf{x}_e[k], \mathbf{u}_e[k])$  từ  $\mathbf{y}_r$  theo (3.22).
4. Tìm  $L_k$  là nghiệm đối xứng, bán xác định dương của phương trình đại số Riccati (3.25). Tính  $\mathbf{u}^*$  theo (3.26).
5. Đưa  $\mathbf{u}^*$  vào điều khiển đối tượng song tuyến liên tục không dừng (3.17) rồi gán  $k := k + 1$  và quay về 2.

Với thuật toán chi tiết như trình bày trên đây thì khoảng thời gian  $T_k$  cho việc xấp xỉ (3.17) thành (3.18) chính là khoảng thời gian cần thiết để thực hiện các phép tính từ bước 2 đến bước 5.

### 3.2.3. Khả năng xử lý điều kiện ràng buộc

Trong thuật toán trên có hai ma trận tham số  $Q_k, R_k$  là tùy chọn. Ta có thể chọn chúng thay đổi theo sự dịch chuyển của cửa sổ dự báo, tức là theo  $t_k, k = 0, 1, \dots$  sao cho nghiệm bài toán còn có thể thỏa mãn thêm được điều kiện ràng buộc:

$$|\boldsymbol{\rho}| \leq \Omega \in \mathbb{R} \quad (3.27)$$

với  $\Omega$  là giá trị chặn trên cho trước, trong khi chất lượng  $\delta = x - x_e[k] \rightarrow \mathbf{0}$  vẫn được đảm bảo.

Theo nguyên lý thay đổi  $Q_k, R_k$  giới thiệu ở tài liệu [1] thì:

- Khi  $R_k$  càng lớn, điều kiện ràng buộc (3.27) càng dễ được thỏa mãn.
- Khi  $Q_k$  càng lớn, càng dễ đạt được chất lượng  $\delta = x - x_e[k] \rightarrow \mathbf{0}$ .

Không bao giờ cả hai ma trận tham số  $Q_k, R_k$  cùng được tăng hoặc giảm tại một bước dịch chuyển của cửa sổ dự báo.

Như vậy, theo nguyên lý này ta có thể xác định quy luật thay đổi  $Q_k, R_k$  là giảm dần  $R_k$  (sau khi điều kiện ràng buộc (3.27) đã được thỏa mãn), hoặc tăng dần  $Q_k$  cùng với  $k$ , tức là cùng với sự dịch chuyển của cửa sổ dự báo.

Ngoài ra, để cài đặt thuật toán, ta cần đến phương pháp giải phương trình đại số Riccati (3.25). Một số phương pháp tìm nghiệm  $L_k$  hữu ích cho phương trình Riccati này đã được giới thiệu ở tài liệu [5].

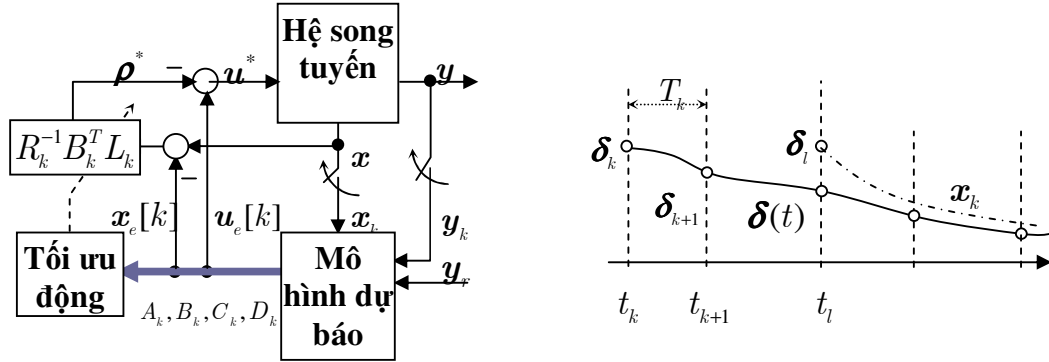
### 3.2.4. Chứng minh tính bám ổn định của phương pháp được đề xuất

Hình 3.3 minh họa hệ điều khiển dự báo với bộ điều khiển làm việc theo thuật toán 3.1 vừa đề xuất.

Hệ kín này làm việc theo nguyên lý phản hồi trạng thái và hiển nhiên không phải là hệ rời rạc, vì ở đó có  $x(t), u^*(t)$  là vector các hàm liên tục. Trong mỗi một cửa sổ điều khiển với độ rộng vô hạn  $[k, \infty]$ , tín hiệu  $u^*(t)$  được đưa vào điều khiển đối tượng trong đúng một khoảng thời gian tính toán của một vòng lặp  $t_k \leq t < t_{k+1}$ . Bởi vậy ta có thể ký hiệu  $u_k^*(t)$  thay vì  $u^*(t)$ . Trong suốt quá trình điều khiển, tín hiệu điều khiển  $u(t)$  sẽ chính là dãy ghép nối các tín hiệu liên tục  $u_k^*(t)$ ,  $k = 0, 1, \dots$  này. Vì vậy hệ kín đó là thuộc nhóm hệ thống dữ liệu mẫu (*sampled data systems*) [4].

Khi áp dụng phương pháp biến phân của điều khiển tối ưu, trong hình 3.3, khi dùng cửa sổ dự báo vô hạn chúng ta cảm thấy mất đi mô hình dự báo nhưng phương pháp này vẫn được gọi là điều khiển dự báo vì nó vẫn thực hiện được việc cực tiểu hóa sai lệch dự báo từ thời điểm hiện tại cho đến vô

cùng ở tương lai và nguyên tắc dịch cửa sổ dự báo cũ vẫn được giữ nguyên ở phương pháp đề xuất này theo nguyên lý Receding Horizon Control (chiến lược điều khiển dịch dần về tương lai) nên thực chất vẫn là điều khiển dự báo.



Hình 3.3. Điều khiển dự báo hệ song tuyến liên tục với cửa sổ dự báo vô hạn

Về tính ổn định của hệ kín trong hình 3.3, ta có phát biểu sau:

**Định lý 3.1:** Nếu các ma trận  $Q_k, R_k, k=0,1, \dots$  là đối xứng xác định dương và:

$$\begin{pmatrix} A_k & B_k \\ C_k & D_k \end{pmatrix} \quad (3.28)$$

không suy biến thì ở hệ điều khiển dự báo cho trong hình 3.3 sẽ có  $y \rightarrow y_r$ .

Chứng minh:

Hiển nhiên, với giả thiết (3.28) thì sẽ đủ nếu ta chứng minh được  $|\delta(t)|$  là bị chặn và có  $\delta(t) \rightarrow \mathbf{0}$ , trong đó  $\delta(t)$  là quỹ đạo sai lệch trạng thái trên toàn bộ trục thời gian, có được nhờ ghép tất cả các quỹ đạo thành phần trong từng cửa sổ dự báo (hình 3.3).

Xét cửa sổ dự báo thứ  $t_k$  bất kỳ. Do  $Q_k, R_k$  là đối xứng xác định dương, nên hệ kín với mô hình sai lệch:

$$\dot{\delta} = (A_k - B_k R_k^{-1} B_k^T L_k) \delta = \widehat{A}_k \delta \quad \text{ứng với } t_k \leq t < t_{k+1} \text{ có:}$$

$$\widehat{A}_k = A_k - B_k R_k^{-1} B_k^T L_k$$

là ổn định. Bởi vậy  $\widehat{A}_k$  là ma trận Hurwitz [4].

Hơn nữa, hệ trên có quỹ đạo trạng thái trong toàn bộ khoảng thời gian là:

$$\boldsymbol{\delta}(t) = e^{\widehat{A}_k(t-t_k)} \boldsymbol{\delta}_k, \text{ trong đó } \boldsymbol{\delta}_k = \boldsymbol{\delta}(t_k)$$

Bởi vậy ta suy ra được với mọi  $t \geq 0$ :

$$\begin{aligned} \boldsymbol{\delta}(t) &= e^{\widehat{A}_k(t-t_k)} \boldsymbol{\delta}_k \\ &= e^{\widehat{A}_k(t-t_k)} e^{\widehat{A}_{k-1}(t_k-t_{k-1})} \boldsymbol{\delta}_{k-1} = \exp\left[\widehat{A}_k(t-t_k) + \widehat{A}_{k-1}(t_k-t_{k-1})\right] \boldsymbol{\delta}_{k-1} \\ &\vdots \\ &= \exp\left[\widehat{A}_k(t-t_k) + \sum_{i=0}^{k-1} \widehat{A}_i(t_{i+1}-t_i)\right] \boldsymbol{\delta}_0 \end{aligned}$$

Ngoài ra, vì  $t_{i+1} - t_i > 0$  và tất cả các ma trận:

$$\widehat{A}_k(t-t_k) \text{ và } \widehat{A}_i(t_{i+1}-t_i) \text{ với } i=0,1, \dots, k-1$$

là Hurwitz, tức là:

$$\lim_{t \rightarrow \infty} \exp \widehat{A}_k(t-t_k) = \mathbf{\Theta}, \forall k \text{ (ma trận có tất cả các phần tử bằng 0)}$$

cũng như:

$$\exp \sum_{i=0}^{k-1} \widehat{A}_i(t_{i+1}-t_i) \text{ là hữu hạn}$$

nên:

$$\lim_{t \rightarrow \infty} |\boldsymbol{\delta}(t)| = 0$$

và đó chính là điều phải chứng minh. ■

### 3.2.5. Khả năng áp dụng cho hệ phi tuyến affine không dừng

Phương pháp điều khiển dự báo đề xuất ở trên cùng với thuật toán 3.1 cho hệ song tuyến liên tục, không dừng cũng hoàn toàn áp dụng được cho cả những hệ phi tuyến affine không dừng, mô tả bởi:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t) + B(\boldsymbol{x}, t)\boldsymbol{u}$$

nếu như ở đó có vector hàm  $\boldsymbol{f}(\boldsymbol{x}, t)$  liên tục theo  $t$  và khi  $|\boldsymbol{x}| \rightarrow 0$  thì  $|\boldsymbol{f}(\boldsymbol{x})|$  cũng tiến về 0 với tốc độ nhanh hơn  $|\boldsymbol{x}| \rightarrow 0$ . Thật vậy, với giả thiết nêu trên thì:

$$\boldsymbol{f}(\boldsymbol{x}, t) = \frac{\boldsymbol{f}(\boldsymbol{x}, t)\boldsymbol{x}^T}{|\boldsymbol{x}|^2} \boldsymbol{x} = A(\boldsymbol{x}, t)\boldsymbol{x} \text{ trong đó } A(\boldsymbol{x}, t) = \frac{\boldsymbol{f}(\boldsymbol{x}, t)\boldsymbol{x}^T}{|\boldsymbol{x}|^2}$$

tức là hệ song tuyến tương đương:

$$\dot{x} = A(x, t)x + B(x, t)u$$

cũng sẽ có ma trận  $A(x, t)$  là liên tục theo  $x$ .

### 3.3. Kết luận chương 3

Toàn bộ nội dung chương 3 là đề xuất mới của luận án về phương pháp điều khiển dự báo cho hệ phi tuyến liên tục không dừng nói chung và cho hệ song tuyến liên tục không dừng nói riêng. Phương pháp đề xuất này được xây dựng trên nền phương pháp biến phân của điều khiển tối ưu và áp dụng được cho cả bài toán điều khiển ổn định cũng như bài toán điều khiển bám ổn định theo giá trị mẫu cho trước ở đầu ra. Nó cũng đã được luận án thể hiện chi tiết dưới dạng **thuật toán 3.1** cũng như ở **định lý 3.1** cho sự khẳng định về chất lượng bám ổn định của hệ kín thu được.

Phương pháp đề xuất này của luận án có những ưu nhược điểm sau:

- Dễ dàng áp dụng được cho hệ phi tuyến liên tục và không dừng chứ không chỉ riêng hệ song tuyến.
- Phương pháp đề xuất sử dụng cửa sổ dự báo vô hạn nên không cần có thêm hàm phạt, vốn rất khó, thậm chí chưa có gợi ý hữu ích nào cho việc xác định chúng.
- Khó xử lý các điều kiện ràng buộc phức tạp. Hiện nay phương pháp đề xuất này của luận án mới chỉ có khả năng xử lý được các điều kiện ràng buộc về giá trị của tín hiệu điều khiển thông qua chỉnh định hai ma trận tham số  $Q_k, R_k$  trong các hàm mục tiêu mô tả sai lệch bám cho bởi công thức (3.24).

## CHƯƠNG 4

### THỰC NGHIỆM KIỂM CHỨNG CHẤT LƯỢNG PHƯƠNG PHÁP ĐÃ ĐỀ XUẤT TRÊN ĐỐI TƯỢNG TRMS

Để kiểm chứng chất lượng bộ điều khiển dự báo do luận án đề xuất, trình bày ở **thuật toán 3.1**, sau đây luận án sẽ áp dụng để điều khiển hệ TRMS và so sánh kết quả khi hệ được điều khiển với bộ điều khiển dự báo làm việc trên nền quy hoạch phi tuyến, tức là khi hệ được điều khiển theo **thuật toán 2.1**. Công việc so sánh kết quả này sẽ được thực hiện bằng cả 2 cách:

- nhờ mô phỏng trên MatLab, và
- thực nghiệm trên bàn thí nghiệm vật lý hệ TRMS có tại Phòng thí nghiệm Điện - Điện tử, trường Đại học Kỹ thuật Công nghiệp.

#### 4.1. Mô hình toán của hệ TRMS

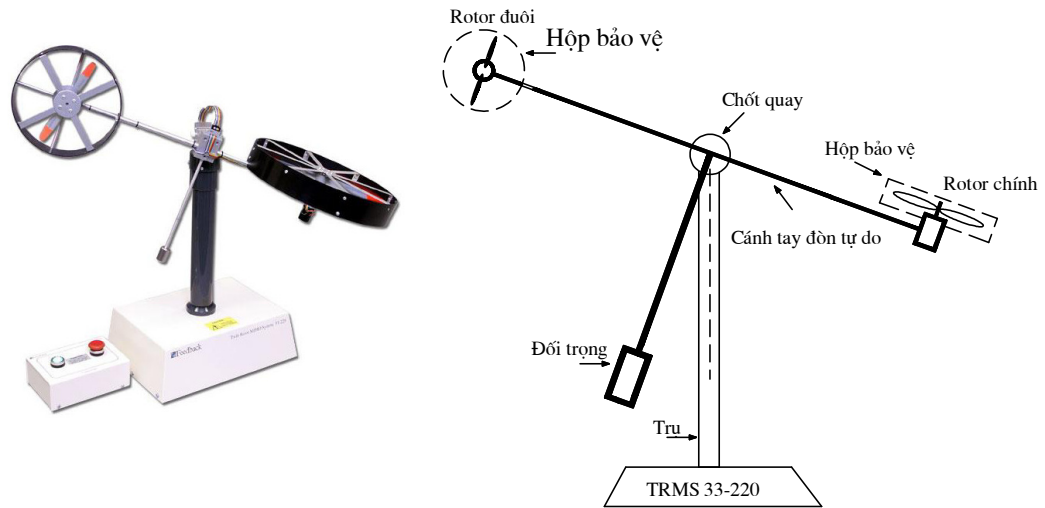
Mô hình toán hệ TRMS đã được rất nhiều tài liệu cung cấp, chẳng hạn như [9], [10], [12], [41], [51], [52], [53], [54], [60], và cũng có nhiều cách khác nhau để xây dựng các mô hình đó. Chúng chủ yếu dựa theo hai hướng chính:

- Thứ nhất là sử dụng các định luật cân bằng của Newton (khi đó mô hình thu được có tên gọi là mô hình tựa Newton).
- Thứ hai là sử dụng công thức cân bằng năng lượng của Lagrange.

##### 4.1.1. Mô tả vật lý hệ TRMS

Hệ thống hai cánh quạt nhiều vào nhiều ra (TRMS) là một hệ thống có 2 bậc tự do để điều chỉnh tầm (góc chao dọc) và hướng (góc đảo lái) của cánh tay đòn, như mô tả ở hình 4.1 a, b.





Hình 4.1. a, b. Cấu hình vật lý hệ TRMS

Hệ TRMS bao gồm 2 cánh quạt hướng vuông góc và được nối với nhau bằng một cánh tay đòn (gọi là cánh tay đòn tự do), cánh tay đòn tự do được nối với một trục thông qua khớp quay, cánh tay đòn có thể quay tự do theo mặt phẳng đứng (tầm, góc chao dọc) và mặt phẳng nằm ngang (hướng, góc đảo lái). Việc điều chỉnh góc chao dọc được thực hiện bởi cánh quạt 1, điều chỉnh hướng được thực hiện bởi cánh quạt 2. Hai cánh quạt này được điều khiển bởi hai động cơ một chiều kích từ độc lập. Các động cơ điện được cấp điện thông qua bộ biến đổi điện tử công suất, tốc độ các cánh quạt được điều chỉnh bằng cách thay đổi điện áp đặt vào động cơ. Ngoài ra hệ thống còn được trang bị một đối trọng gắn vào cánh tay đòn, đối trọng dùng để cân bằng mô men động học ở trạng thái ổn định. Việc điều chỉnh góc tầm và góc hướng của hệ TRMS có tác động xen kẽ lẫn nhau. Xét trên một số phương diện, hoạt động của TRMS giống như máy bay.

#### 4.1.2. Mô hình tựa Newton

Tác giả Akbar Rahideh (2009) của luận án [10] đã xây dựng mô hình toán hệ TRMS theo cả hai phương pháp: sử dụng các định luật cân bằng của Newton và sử dụng công thức cân bằng năng lượng của Lagrange. Trong

luận án này, tôi sử dụng mô hình được xây dựng theo các định luật cân bằng của Newton. Mô hình toán đối tượng TRMS gồm 6 biến trạng thái được đưa ra như sau:

$$\frac{d}{dt} \begin{pmatrix} \omega_h \\ S_h \\ \alpha_h \\ \omega_v \\ S_v \\ \alpha_v \end{pmatrix} = \begin{pmatrix} -\frac{(k_{ah}\varphi_h)^2}{J_{tr}R_{ah}}\omega_h - \frac{B_{tr}}{J_{tr}}\omega_h - \frac{f_1(\omega_h)}{J_{tr}} + \frac{k_{ah}\varphi_h}{J_{tr}R_{ah}}f_6(U_h) \\ \frac{l_t\gamma_t f_2(\omega_h)\cos\alpha_v - f_7(\Omega_h) - f_3(\alpha_h)}{D\cos^2\alpha_v + E\sin^2\alpha_v + F} \\ S_h + \frac{k_m\omega_v\cos\alpha_v}{D\cos^2\alpha_v + E\sin^2\alpha_v + F} \\ -\frac{(k_{av}\varphi_v)^2}{J_{mr}R_{mr}}\omega_v - \frac{B_{mr}}{J_{mr}}\omega_v - \frac{f_4(\omega_v)}{J_{mr}} + \frac{k_{av}\varphi_v}{J_{mr}R_{av}}f_8(U_v) \\ \frac{f_5(\omega_v)(l_m\gamma_m + k_g\Omega_h\cos\alpha_v) - f_9(\Omega_v) + g[(A-B)\cos\alpha_v - C\sin\alpha_v] - 0.5\Omega_h^2 H \sin 2\alpha_v}{J_v} \\ S_v + \frac{k_t}{J_v}\omega_h \end{pmatrix}$$

Equation Section (Next) (4.1)

Trong đó:

$U_v / U_h$ : Điện áp ĐCMC cánh quạt chính/đuôi (V)

$B_{mr} / B_{tr}$ : Hệ số ma sát nhớt của ĐCMC chính và ĐCMC đuôi ( $kgm^2/s$ )

$R_{av} / R_{ah}$ : Điện trở phần ứng của ĐCMC cánh quạt chính/đuôi ( $\Omega$ )

$k_{av}, k_{ah}$ : Các hệ số dương ( $Nm/AWb$ )

$\varphi_v / \varphi_h$ : Từ thông của ĐCMC cánh quạt chính/đuôi (Wb)

$\omega_v / \omega_h$ : Vận tốc góc của ĐCMC cánh quạt chính/đuôi ( $rad/s$ )

$J_{mr} / J_{tr}$ : Mômen quán tính của ĐCMC chính/đuôi ( $kgm^2/s$ )

$g$ : Gia tốc trọng trường ( $m/s^2$ )

$J_v$ : Mômen quán tính của trục ngang (trục hoành) ( $kgm^2$ )

$k_g$ : Hệ số con quay

$l_m$ : Chiều dài của phần chính của cánh tay đòn ( $m$ )

$l_t$ : Chiều dài của phần đuôi của cánh tay đòn ( $m$ )

$\Omega_v$ : Vận tốc góc của cánh tay đòn TRMS trong mặt phẳng thẳng đứng ( $rad/s$ )

$\alpha_v$ : Vị trí theo phương thẳng đứng của cánh tay đòn TRMS ( $rad$ )

$\gamma_m, \gamma_t$ : là các hệ số biến dạng của chiều dài cánh tay đòn chính và đuôi (hai hệ số này có giá trị biến thiên từ 0.9 đến 1). Trường hợp cả hai hệ số này bằng 1 thì mô hình sẽ tương đương với giả thiết là vật liệu tuyệt đối cứng.

$S_h$ : Vận tốc góc của cánh tay đòn TRMS trong mặt phẳng ngang mà không bị ảnh hưởng bởi cánh quạt chính ( $rad/s$ )

$S_v$ : Vận tốc góc của cánh tay đòn TRMS trong mặt phẳng thẳng đứng mà không bị ảnh hưởng bởi cánh quạt đuôi ( $rad/s$ )

$\alpha_h$ : Vị trí theo phương ngang của cánh tay đòn TRMS ( $rad$ )

$\Omega_h$ : Vận tốc góc của cánh tay đòn TRMS trong mặt phẳng ngang ( $rad/s$ )

Các giá trị:

$R_{ah}, L_{ah}, k_{ah}, \varphi_h, J_{tr}, B_{tr}, l_t, D, E, F, k_m, R_{av}, L_{av}, k_{av}, \varphi_v, J_{mr}, B_{mr}, l_m,$

$k_g, g, A, B, C, H, J_v, k_t$  là những hằng số dương và các hàm số:

$f_1(\omega_h), f_2(\omega_h), f_7(\Omega_h), f_3(\alpha_h), f_5(\omega_v), f_9(\Omega_v)$  là các hàm phi tuyến phụ thuộc vào các biến  $\omega_h, \Omega_h, \alpha_h, \omega_v,$  và  $\Omega_v$ .

Bảng 4.1 đưa ra các tham số của đối tượng như sau

Bảng 4.1. Các tham số hệ thống

| Tham số      | Giá trị | Tham số           | Giá trị               |
|--------------|---------|-------------------|-----------------------|
| $l_t(m)$     | 0.282   | $R_a(\Omega)$     | 8                     |
| $l_m(m)$     | 0.246   | $L_a(mH)$         | 0.86                  |
| $l_b(m)$     | 0.290   | $k_a\phi(Nm/A)$   | 0.0202                |
| $l_{cb}(m)$  | 0.276   | $J_{mr}(gcm^2)$   | 1272                  |
| $r_{ms}(m)$  | 0.155   | $J_{tr}(gcm^2)$   | 248                   |
| $r_{ts}(m)$  | 0.100   | $B_{tr}(kgm^2/s)$ | $2.3 \times 10^{-5}$  |
| $m_{tr}(kg)$ | 0.221   | $B_{mr}(kgm^2/s)$ | $4.5 \times 10^{-5}$  |
| $m_{mr}(kg)$ | 0.236   | $k_{th}$          | $3.6 \times 10^{-7}$  |
| $m_{cb}(kg)$ | 0.068   | $k_{tv}$          | $8.7 \times 10^{-7}$  |
| $m_t(kg)$    | 0.015   | $k_{fhp}$         | $1.84 \times 10^{-6}$ |
| $m_m(kg)$    | 0.014   | $k_{fhn}$         | $2.20 \times 10^{-7}$ |
| $m_b(kg)$    | 0.022   | $k_{fvp}$         | $1.62 \times 10^{-5}$ |
| $m_{ts}(kg)$ | 0.119   | $k_{fvm}$         | $1.08 \times 10^{-5}$ |
| $m_{ms}(kg)$ | 0.219   | $k_t$             | $2.6 \times 10^{-5}$  |
| $kg$         | 0.2     | $k_m$             | $2 \times 10^{-4}$    |

Phương trình không gian trạng thái phi tuyến trên đây của hệ TRMS có thể biểu diễn lại được dưới dạng phi tuyến chuẩn như đã cho ở công thức (2.12) là:

$$\begin{cases} \dot{\mathbf{x}} = A(\mathbf{x})\mathbf{x} + B(\mathbf{x})\mathbf{u} \\ \mathbf{y} = C(\mathbf{x})\mathbf{x} \end{cases} \quad (4.2)$$

trong đó:

$$\mathbf{x} = (\omega_h \quad S_h \quad \alpha_h \quad \omega_v \quad S_v \quad \alpha_v)^T$$

$$A(\mathbf{x}) = \begin{pmatrix} \frac{-(k_{ah}\varphi_h)^2}{J_{tr}R_{ah}} - \frac{B_{tr}}{J_{tr}} - \frac{k_{tvp/n}}{J_{tr}} & 0 & 0 & 0 & 0 & 0 \\ \frac{l_t\gamma_t k_{fp/n} |\omega_h| \cos \alpha_v}{f_{10}(\alpha_v)} & \frac{-k_{vfh}}{f_{10}(\alpha_v)} & \frac{-k_{chp/n}}{f_{10}(\alpha_v)} & \frac{-k_{vfh}k_m \cos \alpha_v}{(f_{10}(\alpha_v))^2} & 0 & 0 \\ 0 & 1 & 0 & \frac{k_m \cos \alpha_v}{f_{10}(\alpha_v)} & 0 & 0 \\ 0 & 0 & 0 & \frac{-(k_{av}\varphi_v)^2}{J_{mr}R_{av}} - \frac{B_{mr}}{J_{mr}} - \frac{k_{tvp/n}}{J_{mr}} |\omega_v| & 0 & 0 \\ \frac{-k_{vfv}k_t}{J_v^2} & 0 & 0 & \frac{k_{fp/n} |\omega_v| (l_m\gamma_m + k_g\Omega_h \cos \alpha_v)}{J_v} & \frac{-k_{vfv}}{J_v} & f_{11}(\alpha_v) \\ \frac{k_t}{J_v} & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (4.3)$$

$$B(\mathbf{x}) = \begin{pmatrix} \frac{k_1 k_{ah} \varphi_h}{J_{tr} R_{ah}} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{k_2 k_{av} \varphi_v}{J_{mr} R_{av}} \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad C(\mathbf{x}) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.4)$$

$$\text{và } f_{10}(\alpha_v) = D \cos^2 \alpha_v + E \sin^2 \alpha_v + F$$

$$f_{11}(\alpha_v) = \begin{cases} \frac{g[(A-B)\cos \alpha_v - C \sin \alpha_v]}{\alpha_v} & \text{nếu } \alpha_v \neq 0 \\ 0 & \text{nếu } \alpha_v = 0 \end{cases}$$

Tác giả sử dụng mô hình toán học ở trên để cài đặt thuật toán điều khiển dự báo trên nền quy hoạch phi tuyến và thuật toán điều khiển dự báo bám ổn định để điều khiển hệ thống TRMS nhằm kiểm nghiệm các thuật toán đã xây dựng ở chương 2 và chương 3.

## 4.2. Thiết kế bộ điều khiển dự báo trên nền quy hoạch phi tuyến

### 4.2.1. Thiết kế và cài đặt bộ điều khiển dự báo cho hệ TRMS

Đối tượng hai cánh quạt nhiều đầu vào nhiều đầu ra (TRMS) là hệ thống thí nghiệm về khí động lực học có đặc tính phi tuyến cao, gồm hai đầu vào, hai đầu ra và 6 tham số trạng thái. Trên thế giới hệ thống này đã và đang được nghiên cứu, ứng dụng thử nghiệm để đánh giá và thực hiện các kỹ thuật điều khiển tiên tiến. Tuy nhiên, ở Việt Nam thì TRMS mới được lắp đặt tại một số phòng thí nghiệm của các trường Đại học nhưng hầu như chưa được sử dụng để kiểm nghiệm các thuật toán điều khiển mới.

TRMS là một hệ phi tuyến khá điển hình nên luận án sử dụng hệ thống này để kiểm chứng các kết quả nghiên cứu lý thuyết trong chương 2 và chương 3. Dựa trên mô hình toán học của TRMS trong mục 4.2 luận án sẽ trình bày một số bộ điều khiển dự báo, áp dụng cho hệ TRMS, mô phỏng và đánh giá chất lượng của các bộ điều khiển dự báo đó. Sau đây luận án sẽ trình bày kết quả mô phỏng của phương án cài đặt cụ thể là sử dụng SQP, công bố ở công trình 2.

Để cài đặt thuật toán điều khiển dự báo 2.1 đã trình bày ở chương 2 trong trường hợp có các điều kiện ràng buộc:

$$y_j^{\min} \leq y_j(k+i|k) \leq y_j^{\max}, \quad i=1,2, \dots, N_p \quad (4.5)$$

$$u_j^{\min} \leq u_j(k+i-1|k) \leq u_j^{\max}, \quad i=1,2, \dots, N_c \quad (4.6)$$

$$\Delta u_j^{\min} \leq \Delta u_j(k+i-1|k) \leq \Delta u_j^{\max}, \quad i=1,2, \dots, N_c \quad (4.7)$$

trong đó  $j=1,2$  là chỉ số các phần tử của vector  $\mathbf{y}(k+i|k)$ ,  $\mathbf{u}(k+i|k)$  và  $\Delta \mathbf{u}(k+i|k)$  tại thời điểm  $k+i$  trong tương lai thuộc cửa sổ dự báo thứ  $k$ , mà cụ thể là:

$$\begin{aligned} \mathbf{y}(k+i|k) &= (y_1(k+i|k), y_2(k+i|k))^T = (\alpha_h((k+i)T), \alpha_v((k+i)T))^T \\ \mathbf{u}(k+i|k) &= (u_1(k+i|k), u_2(k+i|k))^T = (U_h((k+i)T), U_v((k+i)T))^T \\ \Delta \mathbf{u}(k+i|k) &= (\Delta u_1(k+i|k), \Delta u_2(k+i|k))^T = (\Delta U_h((k+i)T), \Delta U_v((k+i)T))^T \end{aligned}$$

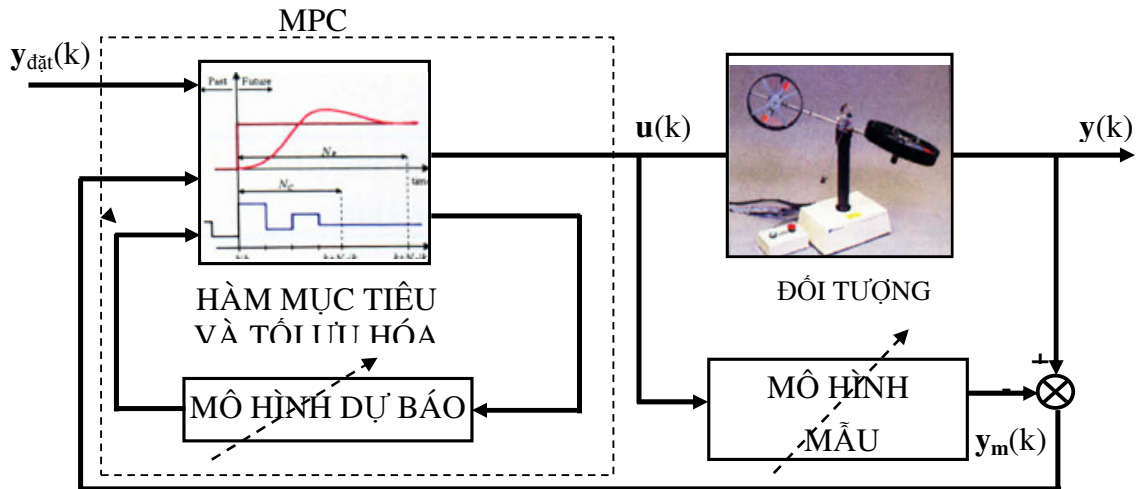
ta cần biến đổi các điều kiện ràng buộc này về dạng chung như đã trình bày tại công thức (2.33).

Do  $y_1 = \alpha_h$  là phần tử thứ 3,  $y_2 = \alpha_v$  là phần tử thứ 6 và hai phần tử của  $\mathbf{u}$  chính là các tín hiệu điều khiển trạng thái  $\hat{\mathbf{x}}$  của mô hình (2.22), nên hai điều kiện ràng buộc (2.27), (2.28) liên quan đến ràng buộc vector trạng thái  $\hat{\mathbf{x}}$ . Điều kiện (2.29) còn lại liên quan đến vector điều khiển  $\Delta \mathbf{u}$ . Bởi vậy công thức (2.33) mô tả điều kiện ràng buộc của hệ TRMS sẽ là:

$$\begin{aligned} U = \left\{ \hat{\mathbf{U}} \in \mathbb{R}^{2N_p} \mid \Delta u_j^{\min} \leq \Delta u_j(k+i-1|k) \leq \Delta u_j^{\max}, \quad i=1,2, \dots, N_c \right. \\ \left. y_j^{\min} \leq \hat{\mathbf{a}}_j^T(i)\hat{\mathbf{x}}_k + \hat{\mathbf{f}}_j^T(i)\hat{\mathbf{U}} \leq y_j^{\max}, \quad j=3,6 \text{ và } i=1,2, \dots, N_p \right. \\ \left. u_j^{\min} \leq \hat{\mathbf{a}}_j^T(i)\hat{\mathbf{x}}_k + \hat{\mathbf{f}}_j^T(i)\hat{\mathbf{U}} \leq u_j^{\max}, \quad j=7,8 \text{ và } i=1,2, \dots, N_c \right\} \end{aligned} \quad (4.8)$$

#### 4.2.2. Mô phỏng trên MatLab

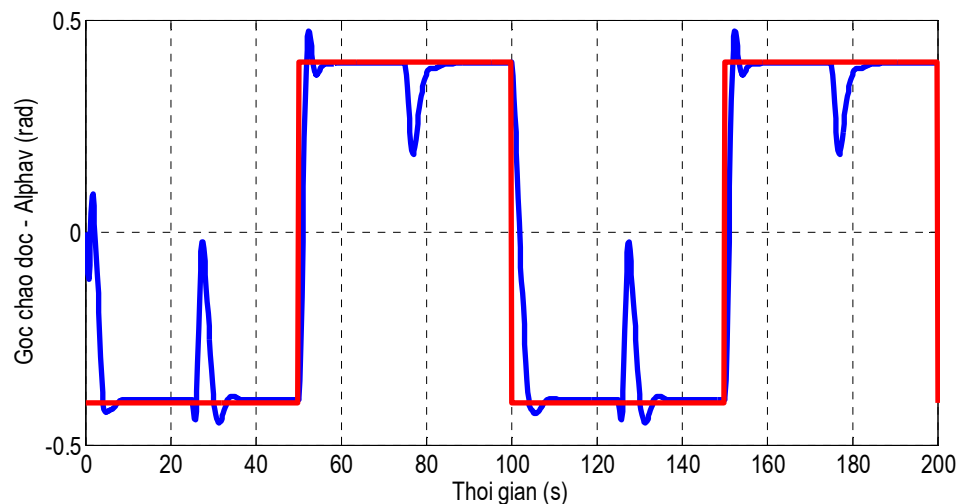
Hình 4.2 biểu diễn cấu trúc khi cài đặt thuật toán xây dựng trên nền quy hoạch phi tuyến đã trình bày ở mục 2.2. Để tìm nghiệm tối ưu cho bài toán (2.30), tác giả sử dụng phương pháp SQP. Chi tiết code chương trình sẽ được đưa ra trong phần phụ lục của luận án.



Hình 4.2. Cấu trúc bộ ĐKDB áp dụng cho thuật toán SQP

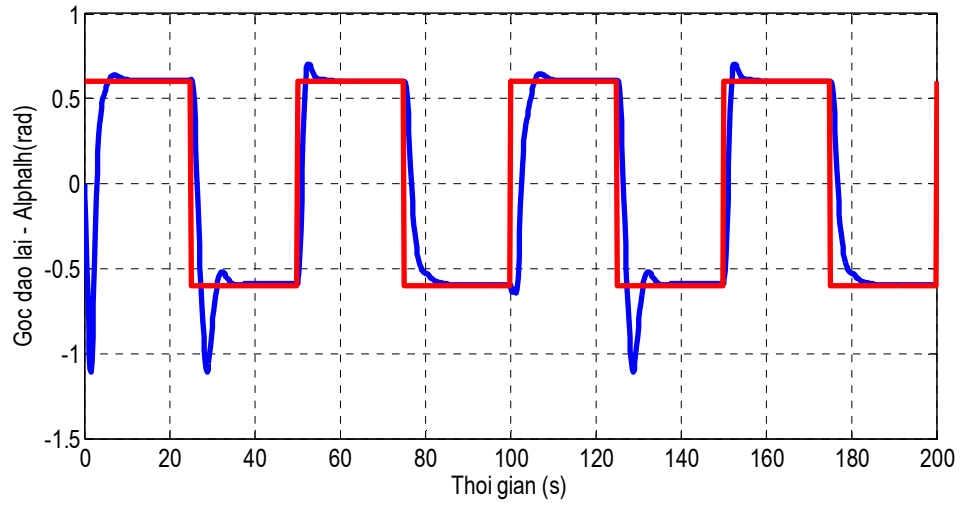
a. Cài đặt bộ điều khiển với thuật toán SQP

Khi cài đặt bộ điều khiển dự báo với thuật toán SQP cho đối tượng TRMS ứng với hàm mục tiêu  $J(\hat{U})$  cho bởi (2.25) cùng các điều kiện ràng buộc  $U$  đưa ra trong các công thức từ (4.5) đến (4.7), nay được viết chung lại thành (4.8), theo sơ đồ cấu trúc trong hình 4.2 sử dụng thuật toán 2.1, ta thu được các kết quả mô phỏng ứng với những dạng tín hiệu đặt khác nhau như trong hình vẽ từ 4.3 đến 4.6. Code chương trình đưa ra trong phụ lục 1 của luận án.

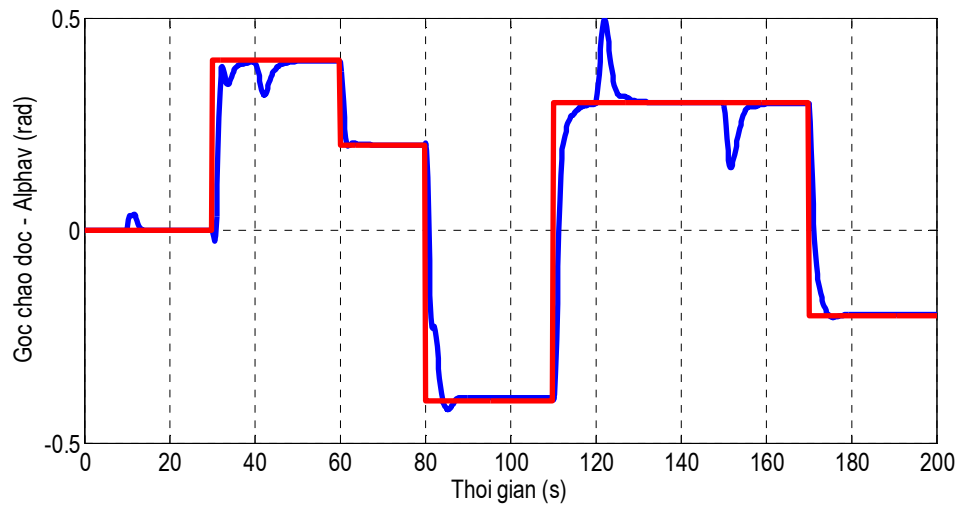


Hình 4.3. Đáp ứng của góc chao dọc khi tín hiệu đặt là xung vuông

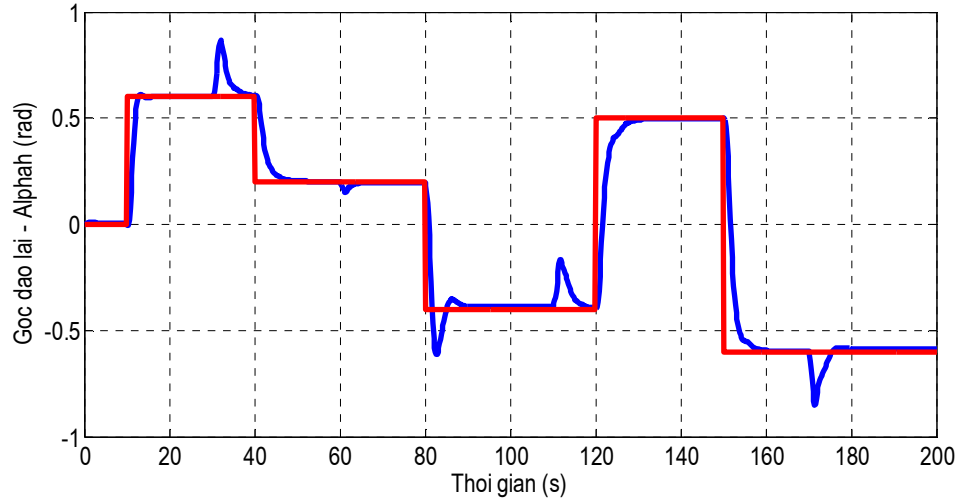




Hình 4.4. Đáp ứng của góc đảo lái khi tín hiệu đặt là xung vuông



Hình 4.5. Đáp ứng của góc chao dọc khi tín hiệu đặt là substep



Hình 4.6. Đáp ứng của góc đảo lái khi tín hiệu đặt là substep

**Nhận xét, đánh giá các kết quả mô phỏng:** Dựa trên các kết quả mô phỏng với thời gian mô phỏng là 200 giây cho thấy khi dùng bộ điều khiển dự báo để điều khiển đối tượng TRMS sử dụng thuật toán SQP thì đáp ứng đầu ra của góc đảo lái và góc chao dọc bám theo quỹ đạo đặt. Ngay khi thay đổi tín hiệu đặt của góc đảo lái  $\alpha_h$  ở giây thứ 25, giây thứ 50... trong hình 4.4 thì góc chao dọc  $\alpha_v$  trong hình 4.3 cũng bị thay đổi theo và ngược lại. Và sau đó các đầu ra lại bám theo các tín hiệu đặt đầu vào. Tuy nhiên, khi thay đổi biên độ tín hiệu đặt càng lớn (cụ thể từ 0.6 rad xuống -0.6 rad hình 4.4) thì độ quá điều chỉnh của cả góc đảo lái và góc chao dọc càng cao, còn khi thay đổi biên độ tín hiệu đặt ở giây thứ 60 của góc chao dọc trong hình 4.5 nhỏ (từ 0.4 rad xuống 0.2 rad) thì độ quá điều chỉnh của góc đảo lái cũng rất nhỏ. Kết quả đó cho ta thấy rất rõ tác động xen kênh của cánh quạt chính và cánh quạt đuôi thông qua góc chao dọc và góc đảo lái (điều này hoàn toàn phù hợp với thực tế là khi thay đổi tốc độ của cánh quạt chính càng lớn thì mức độ ảnh hưởng đến cánh quạt đuôi càng nhiều và ngược lại). Tính ổn định của hệ thống được xét theo tiêu chuẩn tích phân bình phương sai lệch, giá trị đó trong khoảng từ 0.08 đến 0.561 tùy theo giá trị tín hiệu đặt.

Tính ổn định của hệ thống được xét theo tiêu chuẩn tích phân bình phương sai lệch, giá trị đó trong khoảng từ 0.15 đến 0.806 tùy theo giá trị tín hiệu đặt.

### 4.3. Thiết kế bộ điều khiển dự báo trên nền biến phân (phương pháp điều khiển được luận án đề xuất)

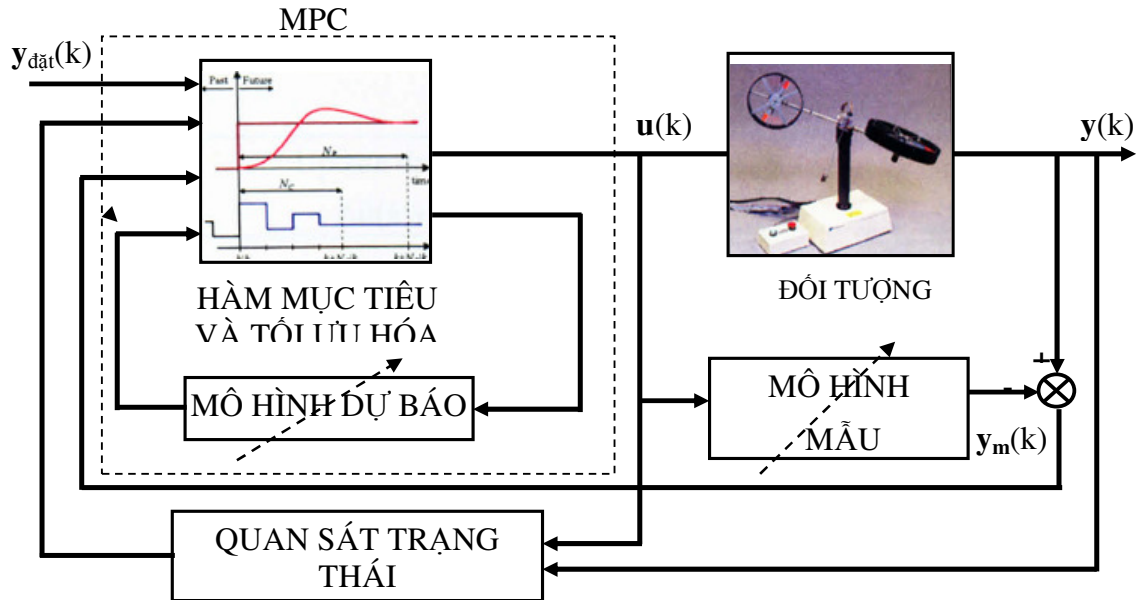
#### 4.3.1. Thiết kế và cài đặt bộ điều khiển

Phương pháp điều khiển dự báo hệ TRMS trình bày ở mục 4.2.1, bên cạnh việc đã giải quyết được trọn vẹn bài toán tối ưu với tất cả những điều kiện ràng buộc nêu trong (4.8), song vẫn bộc lộ một số nhược điểm, bao gồm:

- Bắt buộc phải sử dụng dạng mô hình xấp xỉ không liên tục của đối tượng điều khiển;
- Thời gian tính toán rất lâu;
- Chất lượng ổn định của hệ phụ thuộc vào hàm phạt  $s(x(k + N_c | k))$  được chọn, trong khi vẫn chưa có được một quy tắc chọn hàm phạt tối ưu;
- Vẫn tồn tại sai lệch bám khi các tín hiệu đặt  $\mathbf{y}_{ref}$  biến đổi nhanh.

Để khắc phục các nhược điểm trên, trong mục này luận án sẽ đề xuất một phương pháp điều khiển dự báo khác. Phương pháp mở rộng này sử dụng trực tiếp mô hình liên tục của hệ TRMS, và khoảng dự báo có độ rộng vô hạn nên luôn đảm bảo được tính bám ổn định cho hệ mà không cần sử dụng tới hàm phạt. Hơn thế nữa, phương pháp đề xuất này được xây dựng trên nền lý thuyết điều khiển tối ưu, thay vì tối ưu hóa, nên thời gian tính được rút ngắn đáng kể.

Với thuật toán 3.1 đề xuất trong mục 3.2.2, ta có sơ đồ cấu trúc bộ điều khiển dự báo phản hồi trạng thái để tín hiệu ra bám theo tín hiệu đầu ra mẫu cho hệ TRMS như hình 4.7. Chi tiết code chương trình sẽ được đưa ra trong phần phụ lục của luận án.



Hình 4.7. Sơ đồ cấu trúc bộ ĐKDB phản hồi trạng thái để tín hiệu ra bám theo tín hiệu đầu ra mẫu cho hệ TRMS

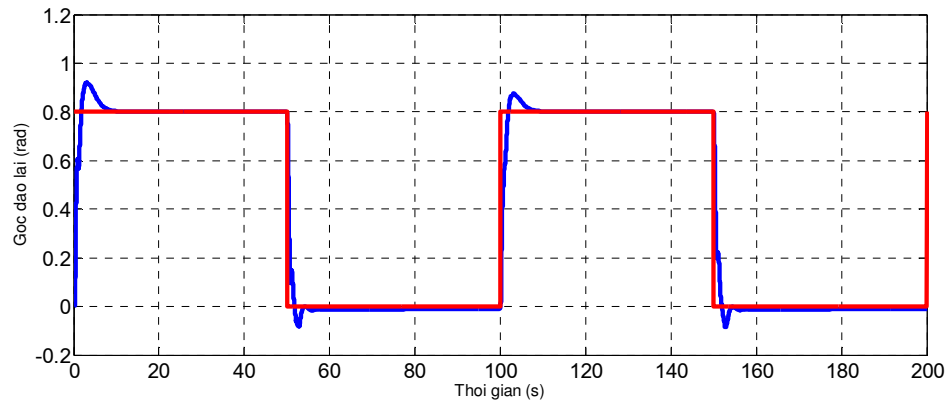
#### 4.3.2. Mô phỏng trên MatLab và so sánh, đánh giá chất lượng

##### Mô phỏng trên matlab

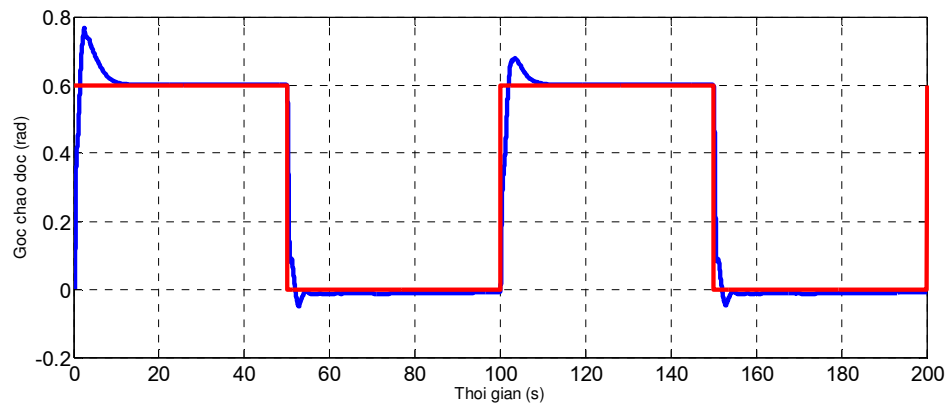
Sử dụng thuật toán 3.1 đưa ra trong mục 3.2.2, cài đặt thuật toán cho đối tượng TRMS theo sơ đồ cấu trúc trong hình 4.7 với các thông số  $Q_k, R_k$  và chu kì trích mẫu  $T$  như sau:

$$T = 0.1, R_k = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}, Q_k = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000000 \end{pmatrix}$$

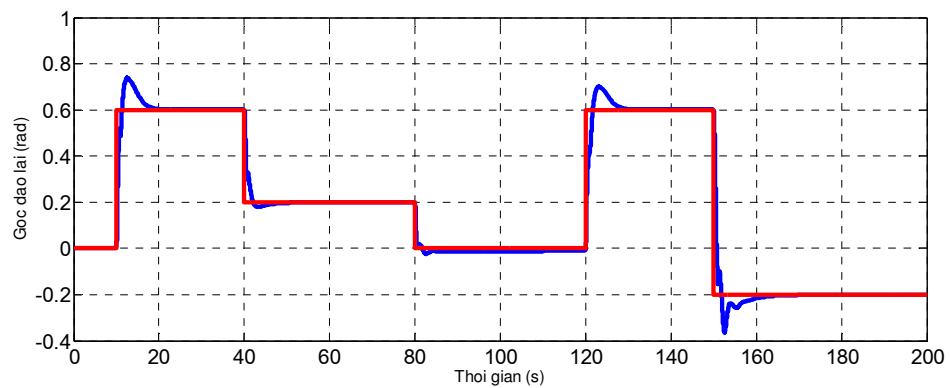
Tác giả thu được các kết quả mô phỏng đưa ra trong các hình từ 4.8 đến 4.11



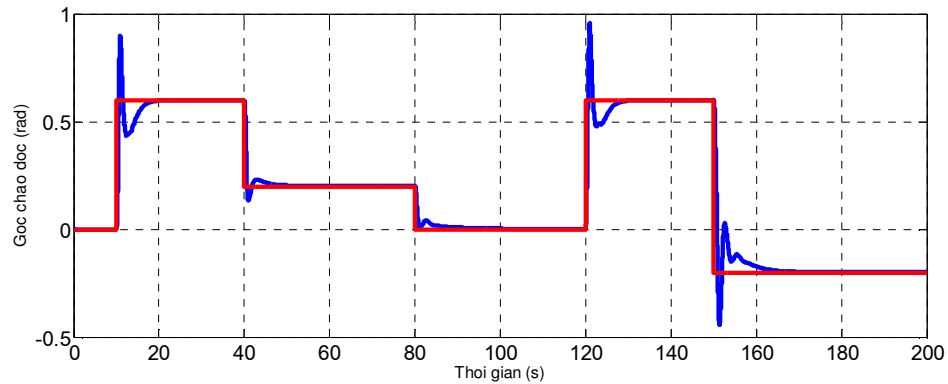
Hình 4.8. Đáp ứng đầu ra góc đảo lái khi tín hiệu đặt là xung vuông



Hình 4.9. Đáp ứng đầu ra góc chao dọc khi tín hiệu đặt là xung vuông



Hình 4.10. Đáp ứng đầu ra góc đảo lái khi tín hiệu đặt là substep



Hình 4.11. Đáp ứng đầu ra góc chao dọc khi tín hiệu đặt là substep

### Nhận xét, đánh giá các kết quả mô phỏng:

Với các kết quả mô phỏng đưa ra trong hình 4.8 và 4.9 lần lượt là đáp ứng của góc đảo lái và đáp ứng của góc chao dọc khi tín hiệu đặt là xung vuông ta thấy sau khoảng 8 giây đáp ứng đầu ra của cả 2 góc bám ổn định theo tín hiệu đặt cho trước. Ở đây khi thay đổi giá trị tín hiệu đặt của cả góc đảo lái và góc chao dọc thì đáp ứng đầu ra bị biến đổi, chứng tỏ hệ TRMS có tác động xen kênh giữa hai kênh dọc và ngang (điều này hoàn toàn phù hợp với thực tế là khi tốc độ của cánh quạt chính thay đổi đột ngột thì tốc độ của cánh quạt đuôi cũng bị thay đổi theo và ngược lại). Trong trường hợp cả hai cánh quạt cùng thay đổi tốc độ thì mức độ ảnh hưởng lẫn nhau càng lớn thể hiện rất rõ trên hình vẽ ở các thời điểm mô phỏng tại giây thứ 0, giây thứ 50, giây thứ 100 và giây thứ 150.

Tương tự như vậy trong hình 4.10 và 4.11 cũng cho ta thấy khi tín hiệu đặt là substep thì đầu ra cũng luôn bám theo tín hiệu đặt đầu vào. Tại giây mô phỏng thứ 10, giây thứ 120 và giây thứ 150 khi tín hiệu đặt của cả góc đảo lái và góc chao dọc thay đổi từ 0 rad lên 0.6 rad và từ 0.6 rad xuống -0.2 rad thì đáp ứng đầu ra ảnh hưởng xen kênh lẫn nhau rất rõ rệt, còn tại các thời gian mô phỏng ở giây thứ 40 hoặc 80 khi thay đổi tín hiệu đặt ở cả 2 góc nhỏ (từ 0.6 rad xuống 0.2 rad hoặc từ 0.2 rad xuống 0 rad) thì tác động xen kênh cũng nhỏ. Từ các kết quả mô phỏng trên đây đã chứng minh tính đúng đắn mà luận

án đã đề xuất ở thuật toán 3.1 trong mục 3.2.2 khi áp dụng phương pháp biến phân để điều khiển bám ổn định cho hệ phi tuyến liên tục (cụ thể ở đây là đối tượng TRMS).

Tính ổn định của hệ thống được xét theo tiêu chuẩn tích phân bình phương sai lệch, giá trị đó trong khoảng từ 0.08 đến 0.561 tùy theo giá trị tín hiệu đặt.

#### *So sánh và đánh giá chất lượng*

Ưu, nhược điểm của các phương pháp tối ưu dùng quy hoạch phi tuyến

*Ưu điểm:* thoả mãn các điều kiện ràng buộc (kể cả ràng buộc trạng thái và ràng buộc đầu vào, ràng buộc đầu ra) trọn vẹn.

*Nhược điểm:* Thời gian tính toán lâu, khó đưa vào cài đặt, ứng dụng trong thực tế.

Để khắc phục những hạn chế của các phương pháp tối ưu quy hoạch phi tuyến luận án đề xuất sử dụng phương pháp biến phân.

Ưu, nhược điểm của phương pháp biến phân

*Ưu điểm:* Thời gian tính toán nhanh, dễ dàng cài đặt, ứng dụng vào thực tế, dùng được với cửa sổ dự báo vô hạn, tính ổn định gần như được đảm bảo chắc chắn.

*Nhược điểm:* Chưa xử lý được trực tiếp các điều kiện ràng buộc phức tạp.

Để khắc phục hạn chế của phương pháp biến phân luận án đề xuất quy luật thay đổi các ma trận trọng số  $Q_k, R_k$  trong hàm mục tiêu thì các ĐKRB được thảo luận.

Bảng 4.1 đánh giá thời gian tính toán khi sử dụng các phương pháp tối ưu khác nhau trong quy hoạch phi tuyến và phương pháp biến phân.

**Bảng 4.1.** So sánh thời gian tính toán của bộ điều khiển dự báo đề xuất trong luận án so với những bộ điều khiển dự báo khác được xây dựng trên nền quy hoạch phi tuyến.

| STT | Phương pháp (Thuật toán)                                   | Thời gian mô phỏng | Thời gian chạy thực của máy tính |
|-----|--|--------------------|----------------------------------|
| 1   | SQP (để giải bài toán tối ưu)                              | 200 giây           | 10 phút                          |
| 2   | Biến phân (giải bài toán tối ưu và điều khiển bám ổn định) | 200 giây           | 6 phút                           |

Ngoài ra, sử dụng thuật toán điều khiển dự báo mà luận án đề xuất (thuật toán 3.1), khi so sánh với những bộ điều khiển dự báo thực hiện trên nền quy hoạch phi tuyến đã trình bày ở mục 4.2.1, thì bên cạnh ưu điểm là bộ điều khiển dự báo đề xuất của luận án đã sử dụng trực tiếp mô hình liên tục của đối tượng, do đó sẽ mang tính sát với thực tế ứng dụng cao hơn, nó còn mang lại thời gian thực hiện cũng nhanh hơn. Bảng 4.1 trên thống kê thời gian tính toán của bộ điều khiển dự báo mà luận án đề xuất, so với những bộ điều khiển dự báo khác được xây dựng trên nền quy hoạch phi tuyến, đã xác nhận kết luận này.

#### 4.4. Thí nghiệm trên mô hình vật lý của hệ TRMS

Trong phần này sẽ đề cập việc xây dựng bộ lọc Kalman mở rộng để ước lượng các thông số trạng thái của hệ thống TRMS và sử dụng bộ ước lượng này để tiến hành thử nghiệm trên hệ thống thực nhằm kiểm tra thuật toán điều khiển đề xuất ở chương 2 và chương 3.

Hầu hết các phương pháp điều khiển hiện đại trong đó có điều khiển dự báo đều cần phải đo được hoặc quan sát được các vectơ trạng thái của mô hình đối tượng. Tuy nhiên, đối với hệ thống thực thường không đo được đầy



đủ các biến trạng thái. Khi đó, ta phải sử dụng bộ quan sát trạng thái để ước lượng vector trạng thái của mô hình đối tượng. Một trong những giải pháp hữu hiệu để ước lượng trạng thái của hệ thống là sử dụng bộ lọc Kalman (cho hệ thống tuyến tính) và lọc Kalman mở rộng (EKF) cho hệ thống phi tuyến.

Trong hệ thống TRMS có 6 biến trạng thái  $(\omega_h \ S_h \ \alpha_h \ \omega_v \ S_v \ \alpha_v)$ . Trong đó ta chỉ đo được 2 biến là góc đảo lái  $(\alpha_h)$  và góc chao dọc  $(\alpha_v)$  còn các biến trạng thái khác không đo được. Vì vậy, để thu được các biến trạng thái ở mỗi lần lặp của thuật toán điều khiển cần phải sử dụng một bộ quan sát trạng thái.

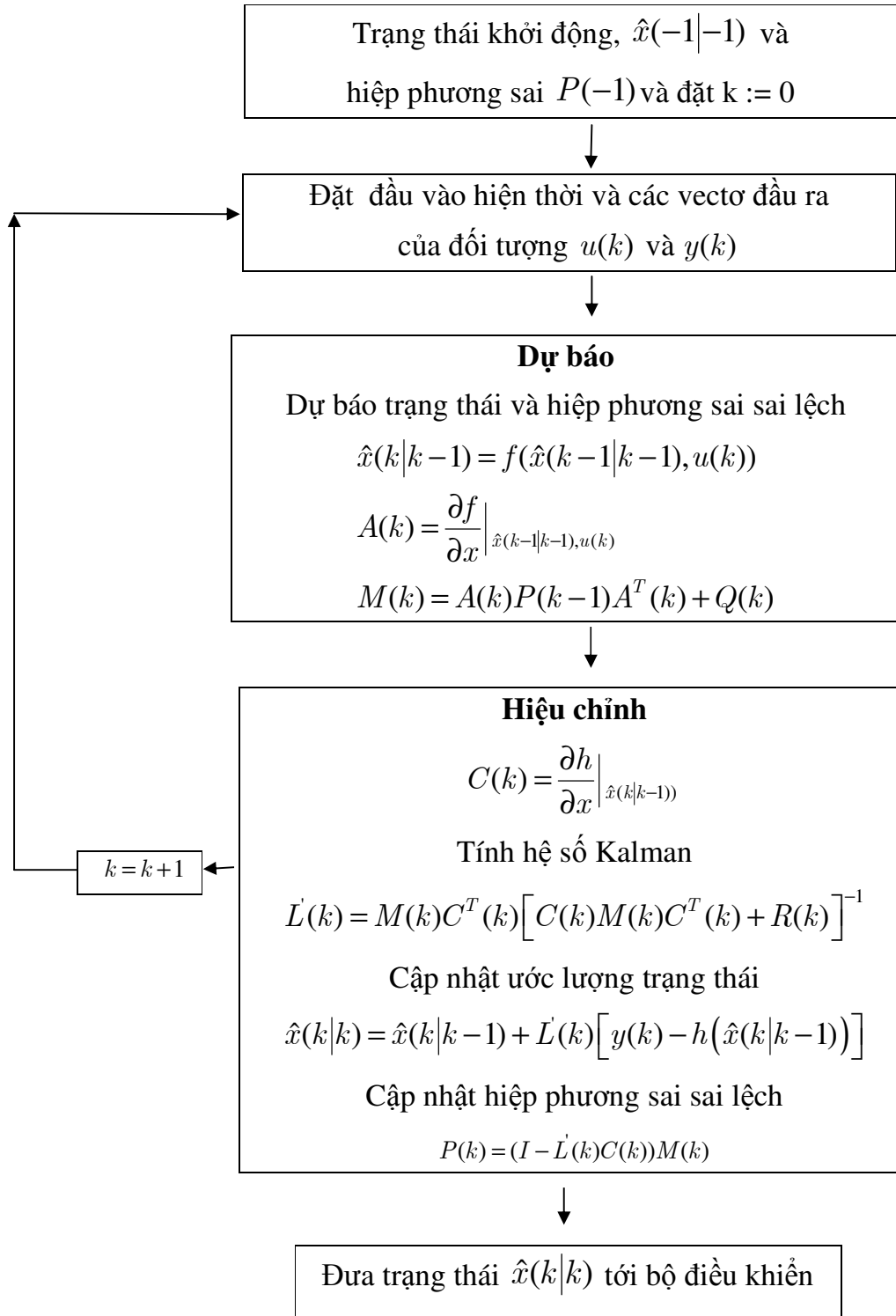
#### 4.4.1. Cài đặt bộ quan sát Kalman

Ước lượng các thông số trạng thái một đối tượng là dựa vào các thông số đo được ở đầu vào và đầu ra của chúng ở mỗi thời điểm mà suy diễn (ước lượng) các thông số trạng thái khác (không đo được) của chúng.

Giả sử một hệ phi tuyến có mô hình không gian trạng thái:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) + w(k) \\ y(k) &= h(x(k)) + v(k) \end{aligned} \tag{4.9}$$

Lưu đồ thuật toán lọc Kalman mở rộng để ước lượng trạng thái đối tượng (4.9) được biểu diễn ở Hình 4.12.



Hình 4.12. Lưu đồ của phương pháp quan sát Kalman mở rộng

trong đó  $w(k)$  và  $v(k)$  là nhiễu trắng với các ma trận hiệp phương sai  $Q(k)$  và  $R(k)$ . Phương trình không gian trạng thái phi tuyến có thể được xấp xỉ bởi phương trình không gian trạng thái phụ thuộc trạng thái dưới đây:

$$\begin{aligned}x(k+1) &= A(x(k))x(k) + B(x(k))u(k) + w(k) \\y(k) &= C(x(k))x(k) + v(k)\end{aligned}\quad (4.10)$$

Sự ước lượng các biến trạng thái tối ưu có thể đạt được khi sử dụng các phương trình lặp dưới đây.

#### **Giai đoạn dự báo**

$$\hat{x}(k|k-1) = f(\hat{x}(k-1|k-1), u(k)) \quad (4.11)$$

$$M(k) = A(\hat{x}(k|k-1))P(k-1)A^T(\hat{x}(k|k-1)) + Q(k) \quad (4.12)$$

trong đó,  $\hat{x}(k|k-1)$  là trạng thái dự báo và  $M(k)$  là hiệp phương sai của sai số dự báo.

#### **Giai đoạn hiệu chỉnh hoặc cập nhật**

$$L(k) = M(k)C^T(\hat{x}(k|k-1)) \left[ C(\hat{x}(k|k-1))M(k)C^T(\hat{x}(k|k-1)) + R(k) \right]^{-1} \quad (4.13)$$

$$\hat{x}(k|k) = \hat{x}(k|k-1) + L(k) \left[ y(k) - h(\hat{x}(k|k-1)) \right] \quad (4.14)$$

$$P(k) = \left( I - L(k)C(\hat{x}(k|k-1)) \right) M(k) \quad (4.15)$$

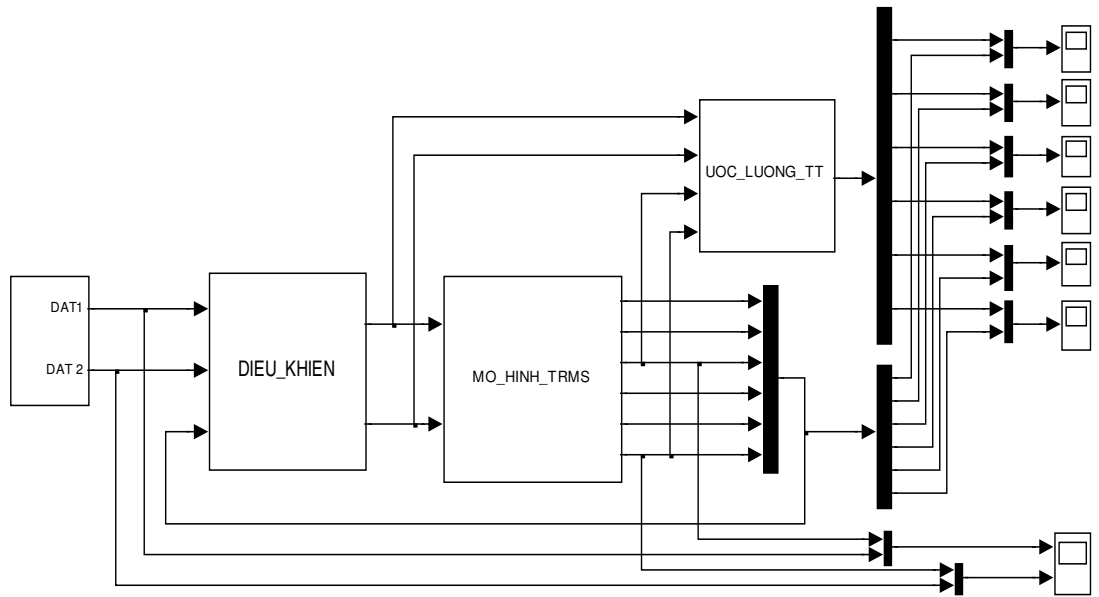
trong đó,  $L(k)$  là khuếch đại Kalman,  $\hat{x}(k|k)$  là ước lượng trạng thái cập nhật, và  $P(k)$  là hiệp phương sai của sai số cập nhật. Mã nguồn của các bộ lọc có thể xem trong phụ lục của luận án.

#### **Kiểm tra bộ quan sát (ước lượng) trạng thái**

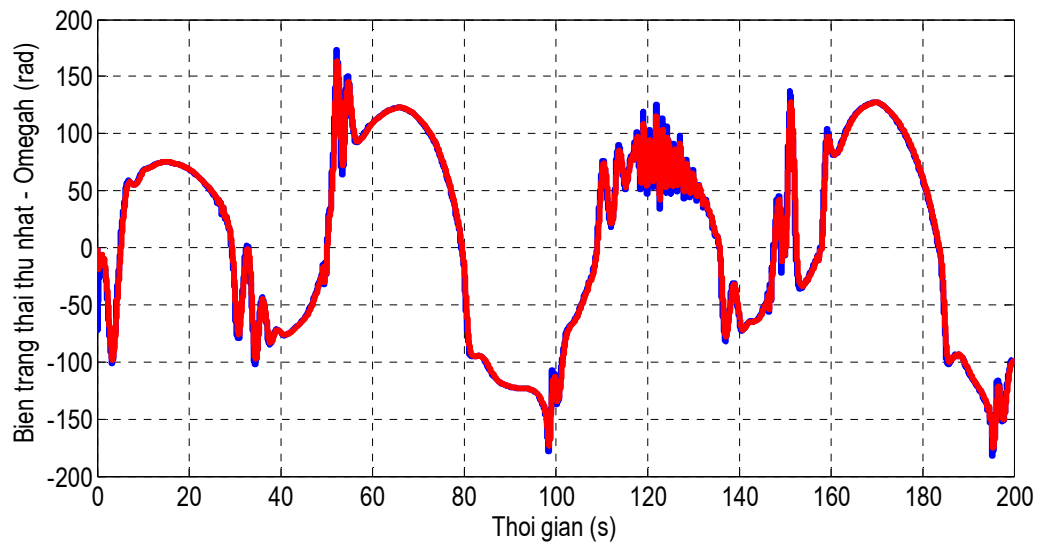
Trong mục này, tác giả sử dụng phương pháp mô phỏng trên phần mềm Matlab-Simulink để kiểm tra độ chính xác của thuật toán ước lượng trạng thái đã trình bày ở mục 4.4.1. Sơ đồ mô phỏng kiểm tra bộ quan sát trạng thái

được chỉ ra trong Hình 4.13. Trong đó khối mô hình TRMS được xây dựng từ biểu thức (4.1), các đầu vào và 2 đầu ra của mô hình (góc tầm và góc hướng) được đưa vào khối quan sát trạng thái, các đầu ra của khối quan sát trạng thái chính là các trạng thái cần ước lượng của TRMS. Các trạng thái ước lượng được so sánh với các trạng thái của mô hình. Các kết quả mô phỏng được chỉ ra trên các hình từ Hình 4.14 đến Hình 4.19. Trong đó:

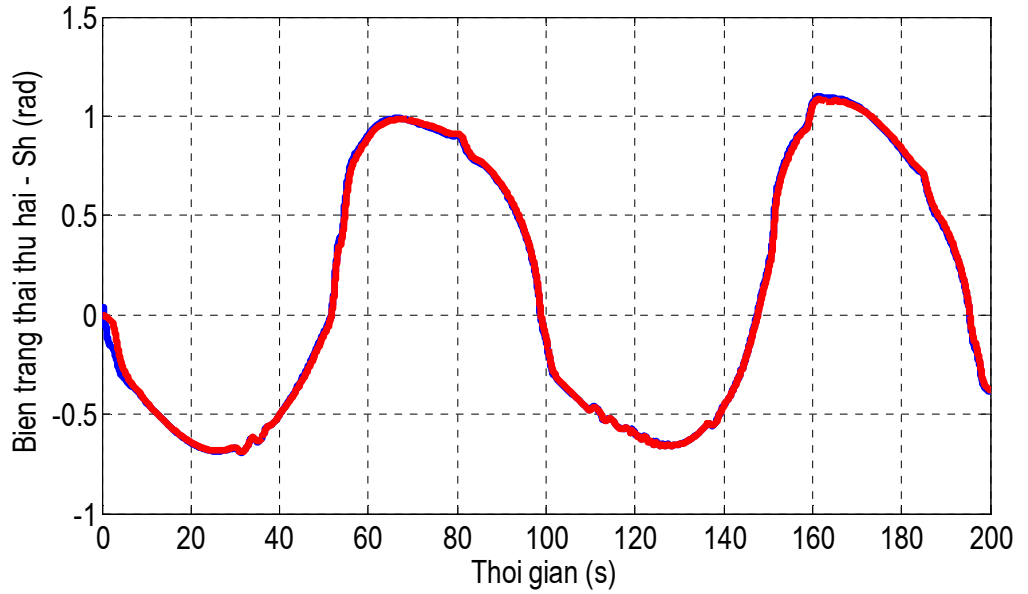
- Hình 4.14. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ nhất ( $\Omega_h$ )
- Hình 4. 15. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ hai ( $S_h$ )
- Hình 4.16. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ ba ( $\alpha_h$ )
- Hình 4.17. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ tư ( $\Omega_v$ )
- Hình 4.18. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ năm ( $S_v$ )
- Hình 4.19. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ sáu ( $\alpha_v$ )



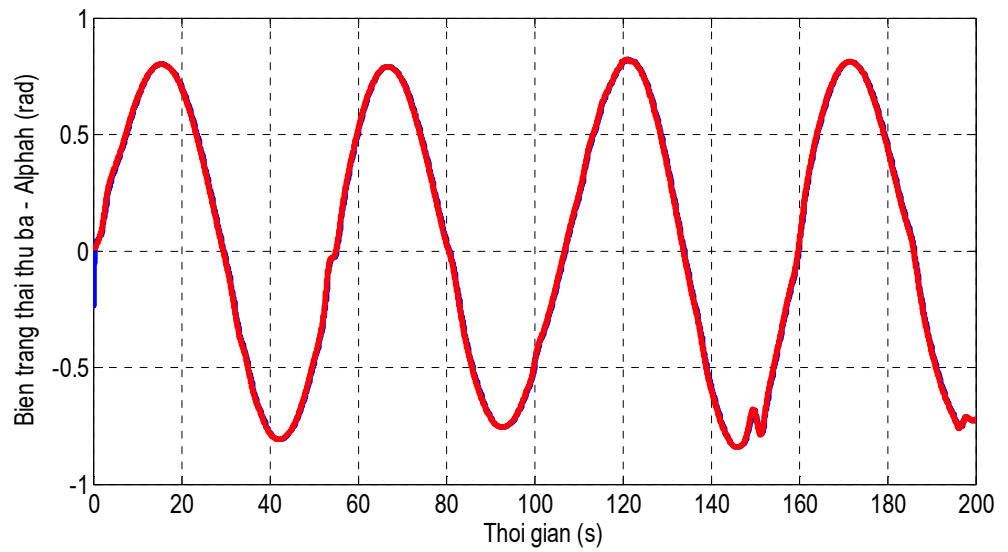
Hình 4.13. Sơ đồ mô phỏng kiểm tra bộ quan sát trạng thái



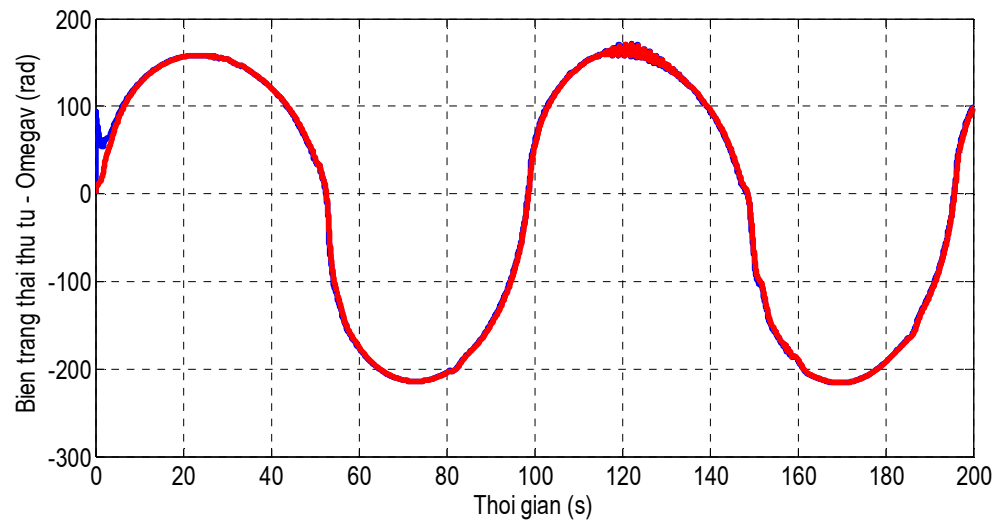
Hình 4.14. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ nhất ( $\Omega_h$ )



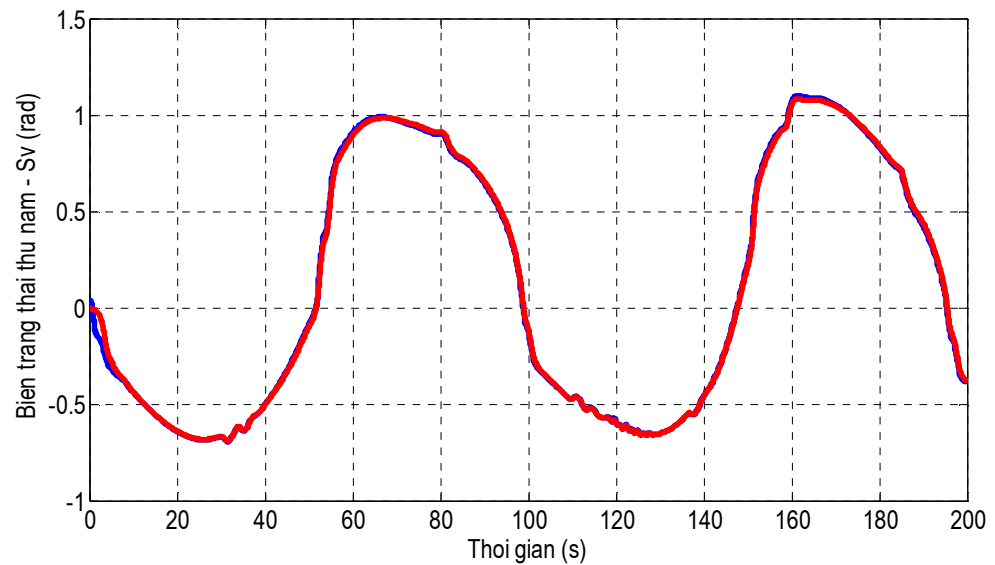
Hình 4.15. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ hai ( $S_h$ )



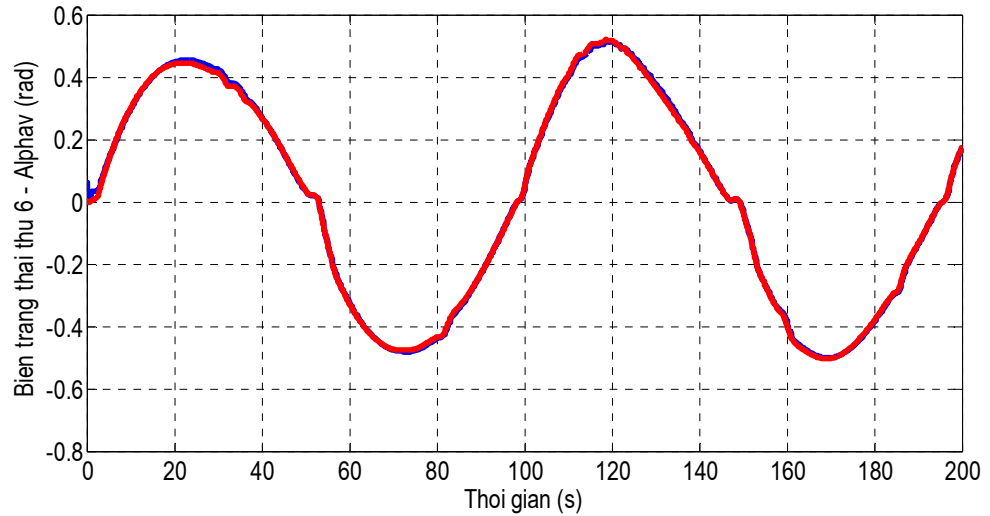
Hình 4.16. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ ba ( $\alpha_h$ )



Hình 4.17. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ tư ( $\Omega_v$ )



Hình 4.18. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ năm ( $S_v$ )



Hình 4.19. Đáp ứng đầu ra bộ quan sát trạng thái so với đáp ứng đầu ra của mô hình của biến trạng thái thứ sáu ( $\alpha_v$ )

#### Nhận xét:

Từ các kết quả mô phỏng ở trên ta thấy rằng các biến trạng thái của bộ ước lượng trạng thái gần trùng khít với các trạng thái của mô hình hệ thống TRMS (sai số lớn nhất chỉ cỡ 0,05% ở các bước ban đầu). Do đó ta có thể sử dụng bộ quan sát này tiến hành thực nghiệm điều khiển dự báo cho hệ thống TRMS.

#### 4.4.2. Các kết quả thực nghiệm

Để kiểm tra bằng thực nghiệm thuật toán điều khiển dự báo tối ưu hóa trên nền qui hoạch phi tuyến và thuật toán điều khiển dự báo bám ổn định theo tín hiệu mẫu ở đầu ra, tác giả đã tiến hành thí nghiệm cho hệ thống TRMS tại phòng thí nghiệm Điện - Điện tử trường Đại học Kỹ thuật Công nghiệp - ĐHTN (Hình 4.20). Việc ghép nối hệ thống với máy tính thông qua CARD ghép nối **DSP1103**, sử dụng công cụ Real-time Workshop của Matlab/Simulink. Tín hiệu đầu ra là góc đảo lái và góc chao dọc được đo bởi Encoder, các trạng thái khác thu được thông qua bộ quan sát trạng thái đề cập ở mục 4.4.1





*Hình 4.20. Hình ảnh thí nghiệm điều khiển hệ thống TRMS*

### **Giới thiệu CARD điều khiển dSPACE1103**

Bộ DSP – DS1103 là sản phẩm xử lý tín hiệu số do hãng dSPACE (Đức) sản xuất, được tích hợp thao tác trên phần cứng (card DS1103), phần mềm (ControlDesk và Simulink) tạo thành một bộ điều khiển, tính toán tốc độ cao trong mô phỏng thời gian thực và thao tác trực tiếp hệ thống. Trong thực tế, DSP1103 được ứng dụng trong các lĩnh vực:

- + Điều khiển robot.
- + Điều khiển động cơ truyền động
- + Điều khiển các cấu trúc mềm

+ ...

Các hệ thống điều khiển thường được thiết kế và mô phỏng trên phần mềm Matlab/Simulink trước khi được cài đặt vào các bộ vi xử lý, vi điều khiển hoặc các bộ điều khiển công nghiệp. Quá trình mô phỏng sẽ cho biết quy luật đáp ứng đầu ra hoặc quy luật của sai số khi thay đổi đầu vào và các tham số bộ điều khiển. Tuy nhiên, đối với nhiều hệ thống phức tạp thì việc lập trình cài đặt các thuật toán điều khiển không hề đơn giản và tốn nhiều thời gian. Vì thế, năm 1991 công ty dSPACE ở Đức (sau đó mở rộng sang Mỹ) đã bắt đầu nghiên cứu một bộ điều khiển cho phép người sử dụng có thể dịch và cài đặt trực tiếp các thuật toán từ Matlab/Simulink xuống bộ điều khiển chỉ thông qua một thao tác kích chuột, bộ điều khiển này có tên là dSPACE, với 2 phiên bản dSPACE1103 và dSPACE1104. Các bộ điều khiển dSPACE thực chất là sự tích hợp của các chip điều khiển tốc độ cao DSP, phần mềm và các giao diện vào ra đơn giản, giúp cho nó có thể dịch và chạy chương trình trực tiếp từ Matlab/Simulink. Bộ điều khiển dSPACE1103 là phiên bản mới nhất của dSPACE, có bộ vi xử lý lên đến 1 GB và tần số Bus là 133 MHz. Nó có thể cung cấp một lượng lớn các giao diện vào ra, giúp cho việc điều khiển các hệ thống phức tạp được thực hiện một cách đơn giản. Bộ điều khiển dSPACE1103 được đưa ra trong hình 4.21



*Hình 4.21. Bộ điều khiển dSPACE1103*

Các tài nguyên của bộ điều khiển dSPACE1103 được liệt kê như sau:

+ Có 20 đầu vào tương tự ADC với chân cắm trực tiếp trên vỏ hộp điều khiển và có thể mở rộng thêm 16 ADC khác. Những ADC này có thể đọc giá trị đo từ cảm biến trong hệ thống điều khiển.

+ Có 8 đầu ra tương tự DAC với chân cắm trực tiếp trên vỏ hộp điều khiển. Đầu ra của những DAC này là điện áp đưa tới bộ khuếch đại công suất (Driver) để điều khiển động cơ.

+ Có 7 encoder cắm trực tiếp trên vỏ hộp điều khiển để đo vị trí của trục động cơ.

+ Số lượng đầu ra số (Digital I/O) cố định là 32 và có thể mở rộng thêm 18 Digital I/O khác (những I/O này bị chia sẻ tài nguyên với các giao diện khác, nghĩa là nó có thể sử dụng làm I/O, ADC, PWM hoặc ngắt).

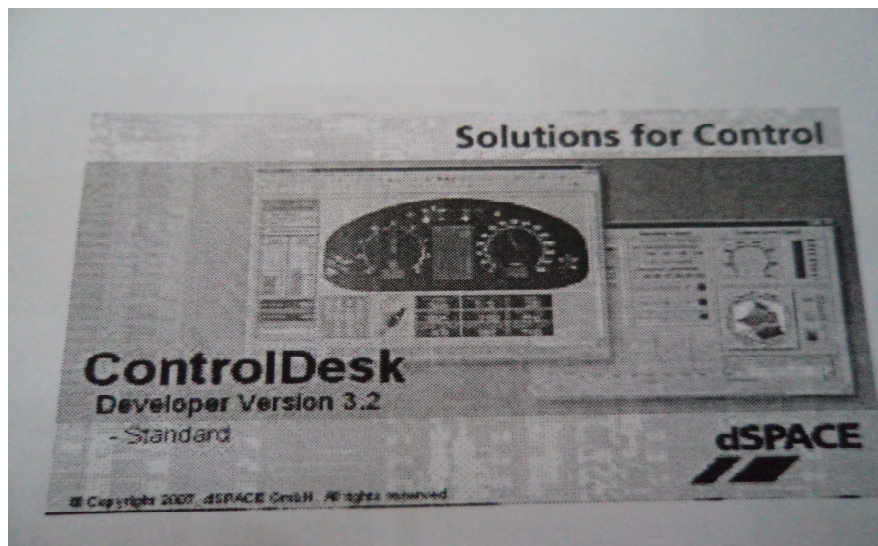
+ Có 4 PWM một pha và 1 PWM ba pha, những PWM cũng có thể sử dụng để điều khiển động cơ. Như vậy với 8 DAC và 4 PWM, dSPACE1103 có thể điều khiển tới 12 động cơ.

+ Với 4 ngắt ngoài (External interrupt) có thể được sử dụng để tạo thêm 4 encoder.

+ Với 3 bộ định thời gian (Timer) có thể kết hợp với các I/O để tạo thêm nhiều giao diện encoder khác.

+ Với 2 cổng RS232 và 2 cổng RS422, dSPACE1103 có thể giao tiếp với máy tính và các bộ điều khiển khác một cách dễ dàng.

Đi kèm với bộ điều khiển dSPACE là một phần mềm điều khiển và giám sát ControlDesk. Phần mềm này chạy trên máy tính bàn hoặc máy tính xách tay để giao tiếp với bộ điều khiển dSPACE thông qua card PCI. Đây là phần mềm rất đơn giản vì người sử dụng chỉ việc dùng chuột gắp các khối cần thiết để hiển thị và thay đổi các biến trong thuật toán điều khiển. Phần mềm ControlDesk khi dùng với phiên bản dSPACE1103 thì các biến của chương trình điều khiển được lưu vào một file trong cùng thư mục làm việc sau khi dịch chương trình từ Matlab/Simulink xuống dSPACE. File này có thể được mở bằng phần mềm ControlDesk để giao tiếp với bộ điều khiển dSPACE. Phần mềm ControlDesk được đưa ra trong hình 4.22.



Hình 4.22. Phần mềm giám sát và điều khiển ControlDesk

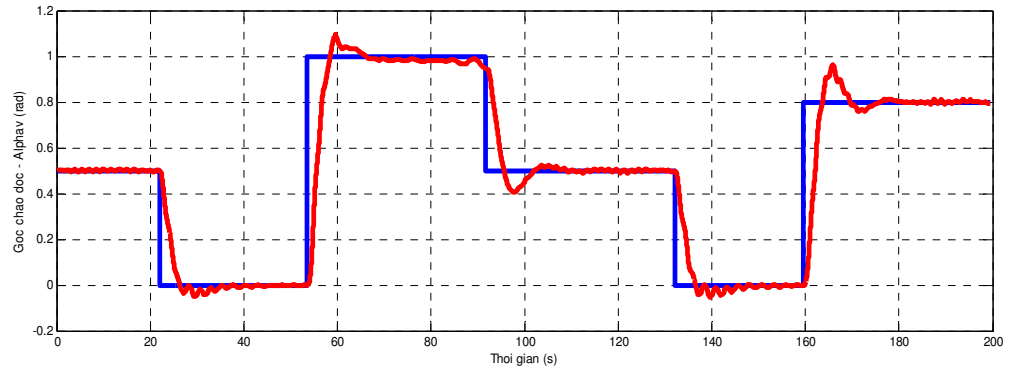
Phần mềm ControlDesk cung cấp các khối đồ họa để hiển thị và thay đổi giá trị của biến điều khiển dưới dSPACE. Các khối đồ họa này có thể là đồng hồ hiển thị, thanh trượt, đồ thị, ..., được gắn với các biến điều khiển thông qua một hành động nhấp và thả chuột. Nói chung, phần mềm ControlDesk đã làm đơn giản hóa mọi thao tác của người sử dụng, giúp cho việc tiếp cận và làm việc với nó được dễ dàng.

Như vậy, với bộ điều khiển dSPACE và phần mềm ControlDesk, người sử dụng không cần viết code, mà chủ yếu sử dụng chuột để thiết kế, điều khiển và giám sát hoạt động của hệ thống.

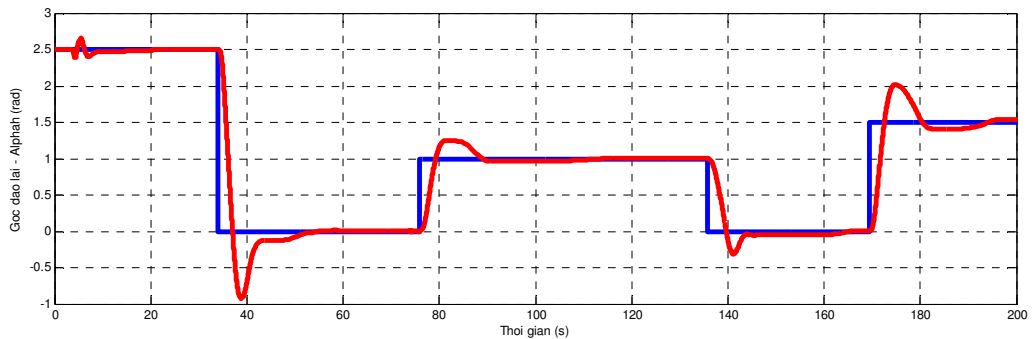
Các kết quả thực nghiệm được biểu diễn trên các hình từ Hình 4.23 đến Hình 4.26. Trong đó:

- Hình 4.23 là đáp ứng đầu ra của góc chao dọc ứng với lượng đặt khác nhau khi sử dụng bộ điều khiển dự báo tối ưu hóa trên nền qui hoạch phi tuyến;
- Hình 4.24 là đáp ứng đầu ra của góc đảo lái ứng với lượng đặt khác nhau khi sử dụng bộ điều khiển dự báo tối ưu hóa trên nền qui hoạch phi tuyến;
- Hình 4.25 là đáp ứng đầu ra của góc chao dọc ứng với lượng đặt khác nhau khi sử dụng bộ điều khiển dự báo bám ổn định theo tín hiệu mẫu ở đầu ra;
- Hình 4.26 là đáp ứng đầu ra của góc đảo lái ứng với lượng đặt khác nhau khi sử dụng bộ điều khiển dự báo bám ổn định theo tín hiệu mẫu ở đầu ra.





Hình 4.23. Đáp ứng đầu ra của góc chao dọc khi sử dụng bộ điều khiển dự báo tối ưu hóa trên nền qui hoạch phi tuyến

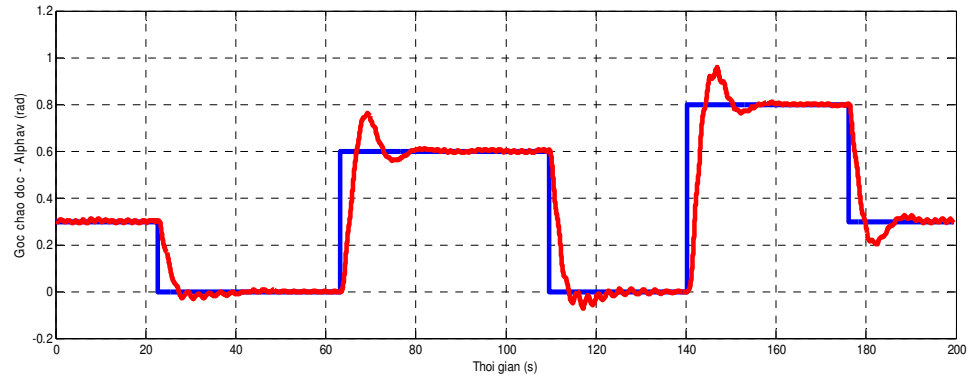


Hình 4.24. Đáp ứng đầu ra của góc đảo lái khi sử dụng bộ điều khiển dự báo tối ưu hóa trên nền qui hoạch phi tuyến

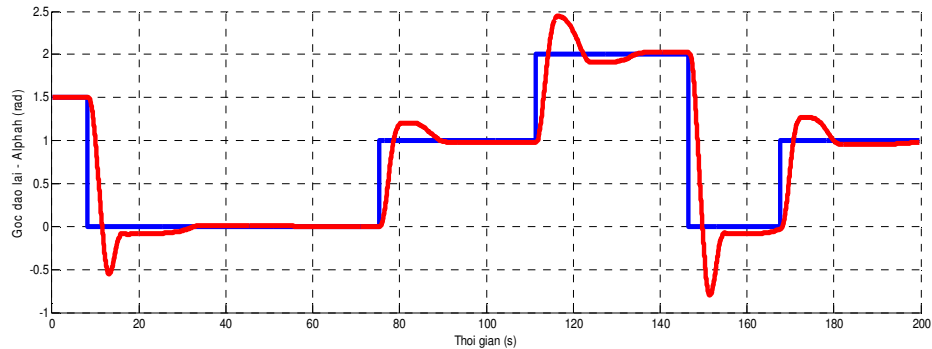
### Nhận xét, đánh giá các kết quả thực nghiệm khi sử dụng thuật toán SQP:

Kết quả thực nghiệm đưa ra trong hình 4.23 và 4.24 cho thấy tuy số lần dao động nhiều hơn khi mô phỏng, và độ quá điều chỉnh của góc đảo lái ở giây thứ 39 hơi lớn khi thay đổi biên độ tín hiệu đặt lớn (từ 2.5 rad xuống 0 rad) nhưng đáp ứng đầu ra của góc chao dọc và góc đảo lái vẫn bám theo tín hiệu đặt.

Tính ổn định của hệ thống được xét theo tiêu chuẩn tích phân bình phương sai lệch, giá trị đó khoảng từ 0.83.



Hình 4.25. Đáp ứng đầu ra của góc chao dọc khi sử dụng bộ điều khiển dự báo bám ổn định theo tín hiệu mẫu ở đầu ra



Hình 4.26. Đáp ứng đầu ra của góc đảo lái khi sử dụng bộ điều khiển dự báo bám ổn định theo tín hiệu mẫu ở đầu ra

### Nhận xét, đánh giá các kết quả thực nghiệm khi sử dụng phương pháp biến phân:

Các kết quả thu được thực nghiệm khi sử dụng phương pháp biến phân được đưa ra trong hình 4.25 và hình 4.26 cho thấy các đặc tuyến động của hệ thống bám sát các tín hiệu đặt. Tính ổn định của hệ thống được xét theo tiêu chuẩn tích phân bình phương sai lệch, giá trị đó khoảng từ 0.86.

Vậy, kết quả thực nghiệm cũng đã khẳng định tính đúng đắn của 2 thuật toán mà luận án đề xuất và tính khả thi của chúng khi áp dụng điều khiển các hệ thống thực.

#### 4.5. Kết luận chương 4

Chương 4 của luận án đã đề cập tới những vấn đề cơ bản sau:

Một là, sử dụng mô hình toán (4.1) và áp dụng thuật toán 2.1 để cài đặt bộ điều khiển dự báo cho hệ TRMS với cửa sổ dự báo hữu hạn thông qua phương pháp SQP để giải bài toán tối ưu trong chiến lược tối ưu hóa của điều khiển dự báo.

Hai là, sử dụng mô hình toán (4.1) và dùng thuật toán 3.1 để cài đặt bộ điều khiển dự báo cho hệ TRMS với cửa sổ dự báo vô hạn qua việc sử dụng phương pháp biến phân để giải bài toán tối ưu trong chiến lược tối ưu hóa của điều khiển dự báo.

Ba là, đã xây dựng thuật toán ước lượng trạng thái cho hệ thống TRMS và sử dụng chúng để kiểm chứng thuật toán điều khiển dự báo trên nền qui hoạch phi tuyến và thuật toán điều khiển dự báo bám ổn định theo tín hiệu mẫu ở đầu ra trên hệ thống thực tại phòng thí nghiệm điện, điện tử - trường Đại học KTCN - Đại học Thái Nguyên.

Các kết quả mô phỏng và thực nghiệm được đưa ra đã chứng minh tính ứng dụng và chính xác của các thuật toán mà luận án đã đề xuất. Đồng thời cũng khẳng định khả năng ứng dụng các các thuật toán này trong chiến lược tối ưu hóa của bộ điều khiển dự báo hệ phi tuyến nói chung và hệ thống TRMS nói riêng.



## KẾT LUẬN

Với mục tiêu nghiên cứu đề xuất thuật toán mới giải bài toán tối ưu trong hệ thống điều khiển dự báo phi tuyến MIMO và đưa ra các thuật toán điều khiển cho một lớp các đối tượng phi tuyến (hệ song tuyến), kết quả nghiên cứu của luận án đã có một số kết quả mới sau:

1. Bổ sung hoàn thiện thuật toán thiết kế bộ điều khiển dự báo sử dụng phương pháp quy hoạch phi tuyến để giải bài toán tối ưu trong chiến lược tối ưu hóa của điều khiển dự báo, mở rộng khả năng ứng dụng điều khiển dự báo vào điều khiển các đối tượng công nghiệp. Kết quả nghiên cứu được kiểm chứng bằng lập trình mô phỏng trên máy tính và thực nghiệm trên mô hình vật lý của hệ thống TRMS cụ thể.

2. Xây dựng phương pháp luận để thiết kế bộ điều khiển dự báo cho hệ phi tuyến và đề xuất một giải pháp mới trong chiến lược tối ưu hóa của điều khiển dự báo hệ phi tuyến, cụ thể là: Điều khiển dự báo phi tuyến trên nền phương pháp biến phân, luận án đã phát biểu và chứng minh định lý về tính bám ổn định theo tín hiệu ra mẫu cho hệ phi tuyến liên tục khi cửa sổ dự báo tiến đến vô cùng. Kết quả nghiên cứu này đã khắc phục được các nhược điểm của các phương pháp giải bài toán tối ưu trên nền quy hoạch phi tuyến và rút ngắn được thời gian tính toán, nâng cao chất lượng điều khiển, mở rộng khả năng ứng dụng và cài đặt bộ điều khiển dự báo vào điều khiển các đối tượng thực.

3. Cài đặt thuật toán mới mà luận án đưa ra thông qua mô phỏng trên máy tính và thực hiện điều khiển trên mô hình vật lý tại phòng thí nghiệm Điện - Điện tử trường Đại học Kỹ thuật Công nghiệp - Đại học Thái Nguyên, thông qua đó đã kiểm chứng và khẳng định tính khả thi của thuật toán đề xuất.

Để nâng cao chất lượng bộ điều khiển, mở rộng khả năng ứng dụng của điều khiển dự báo vào điều khiển các đối tượng thực, tác giả sẽ tiếp tục nghiên cứu hoàn thiện thêm các thuật toán đã đề xuất và triển khai ứng dụng điều khiển các hệ thống trong thực tế. Đồng thời nghiên cứu đề xuất thêm các thuật toán mới có thời gian tính toán nhanh hơn.

## **DANH MỤC CÔNG TRÌNH ĐÃ CÔNG BỐ LIÊN QUAN ĐẾN ĐỀ TÀI**

1. Nguyễn Thị Mai Hương, Mai Trung Thái, Nguyễn Hữu Chinh, Lại Khắc Lãi (2014), "Nghiên cứu ảnh hưởng của các tham số trạng thái trong hệ thống hai cánh quạt nhiều đầu vào nhiều đầu ra", *Tạp chí Khoa học Công nghệ Đại học Thái Nguyên*, Số 06, tập 120, tr. 87 - 92.
2. Nguyen Thi Mai Huong, Mai Trung Thai, Nguyen Huu Chinh, Tran Thien Dung, Lai Khac Lai (09-2014), "Model Predictive Control for Twin Rotor MIMO system", *The University of Da Nang Journal of science and Technology* , 12[85], pp. 39 - 42.
3. Nguyễn Thị Mai Hương, Mai Trung Thái, Lại Khắc Lãi, Đỗ Thị Tú Anh (2014), "Ổn định hóa hệ thống hai cánh quạt nhiều đầu vào nhiều đầu ra dựa trên phương pháp quy hoạch động của Bellman", *Tạp chí Khoa học và công nghệ Đại học Thái Nguyên*, Tập 128, số 14, tr. 161 - 165.
4. Nguyễn Thị Mai Hương, Mai Trung Thái, Lê Thị Huyền Linh, Lại Khắc Lãi (2013), "Nghiên cứu chiến lược tối ưu hóa trong điều khiển dự báo", *Tạp chí Khoa học Công nghệ Đại học Thái Nguyên*, Tập 113, số 13, tr. 115 - 121;
5. Huong Nguyen T. M., Thai Mai T., Lai Lai K. (2015), "Model Predictive Control to get Desired Output with Infinite Predictive Horizon for Bilinear Continuous Systems", *International Journal of Mechanical Engineering and Robotics Research*, Vol. 4, No. 4, pp. 299 - 303.

## TÀI LIỆU THAM KHẢO

### Tiếng Việt

1. Đỗ Thị Tú Anh (2015), *Điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ phi tuyến*, Luận án tiến sĩ Kỹ thuật, Trường Đại học Bách Khoa Hà Nội.
2. Nguyễn Phương Huy (2014), *Mô hình kết hợp logic mờ và giải thuật di truyền cho bài toán quản lý hàng đợi tích cực trên mạng TCP/IP*, Luận án tiến sĩ Kỹ thuật, Trường Đại học Bách Khoa Hà Nội.
3. Nguyễn Doãn Phước (2009), *Lý thuyết điều khiển nâng cao (in lần thứ 4)*, Nhà xuất bản Khoa học và Kỹ thuật.
4. Nguyễn Doãn Phước (2012), *Phân tích và điều khiển hệ phi tuyến*, Nhà xuất bản Bách khoa, Hà Nội.
5. Nguyễn Doãn Phước (2015), *Tối ưu hóa trong điều khiển và điều khiển tối ưu*, Nhà xuất bản Bách khoa, Hà Nội.
6. Trần Quang Tuấn (2012), *Về một phương pháp điều khiển dự báo thích nghi mờ cho đối tượng phi tuyến bất định*, Luận án tiến sĩ Kỹ thuật, Trường Đại học Bách Khoa Hà Nội.

### Tiếng Anh

7. Abbas M. A. (2011), “Survey paper: Optimization in Model Predictive Control”, *FEAS, OUIT*, Canada.  
<http://www.scribd.com/doc/37338823/Optimization-Methods-in-Model-Predictive-Control-Survey#scribd> (online).
8. Adetola V., Guay M. (2010), “Intergration of real-time optimization and Model Predictive Control”, *Journal of Process Control*, 20, pp. 125 - 133.
9. Ahmad S. M., Shaheed M. H., Chipperfield A. J., and Tokhi M. O. (2000), “Nonlinear modelling of a twin rotor MIMO system using radial

- basis function networks”, *IEEE National Aerospace and Electronics Conference*, pp. 313 - 320.
10. Akabar Rahideh (2009), *Model Identification and Robust nonlinear Model predictive control of a twin rotor MIMO system*, A thesis Doctor of philosophy in the school of Engineering and Materials Science Queen Mary, University of London.
  11. Alanisa A. Y., Sanchezb E. N., Loukianovb A. G., Hernandez E. A. (2010), “Discrete-time recurrent high order neural networks for nonlinear identification”, *Journal of the Franklin Institute*, 347 (2010), pp. 1253 -1265.
  12. Aldebrez F. M., Mat Darus I. Z., and Tokhi M. O (2004), “Dynamic modelling of a twin rotor system in Appendix hovering position”, *International Symposium on Control, communications and signal processing*, pp. 823 - 826.
  13. Aufderheide B., Bequette B. W. (2001) “A Variably Tuned Multiple Model Predictive Controller Based on Minimal Process Knowledge”, *American Control Conference, Arlington*, pp. 3490 - 3495.
  14. Aufderheide B., Prasad V., Bequette B. W. (2001), “A Comparison of Fundamental Model-Based and Multiple Model Predictive Control”, *IEEE Conference on Decision and Control, Orlando, Florida USA*, pp. 4863 - 4868.
  15. Bemporad, A., & Morari, M. (2004), “Robust model predictive control” *A survey. In Proc. of European Control Conference, Porto, Portugal*, pp. 939 - 944.
  16. Bitmead R., Gevers M., Wertz V. (1990), *Adaptive Optimal Control: The Thinking Man's General Predictive Control*, Prentice Hall.

17. Camacho, Bordons (2004), *Model predictive control*, Springer Verlag, London.
18. Chadlia M., Akhenak A., Ragot J., Maquin D. (2009), "State and unknown input estimation for discrete time multiple model", *Journal of the Franklin Institute*, 346, pp. 593 - 610.
19. Cuzzola F. A., Geromel J. C., Morari M. (2002), "An Improved Approach for Constrained Robust Model Predictive Control", *Automatica*, Vol. 38, pp. 1183 - 1189.
20. De Nicolao G., Magni L., Scattolini R. (2000), "Stability and robustness of nonlinear receding horizon control", in: F. Allgower, A. Zheng (Eds.), *Nonlinear Model Predictive Control, Progress in Systems and Control Theory*, Birkhauser Verlag, Vol. 26, pp. 3 - 22.
21. De Oliveira Kothare S. L., Morari M. (2000), "Contractive model predictive control for constrained nonlinear systems", *IEEE Transactions on Automatic Control*, 45 (6), pp. 1053 - 1071.
22. De Oliveira N. M. C., Biegler L. T. (1995), "An extension of Newton-type algorithms for nonlinear process control", *Automatica*, 31 (2), pp. 281 - 286.
23. Diehl M., Bock H. G., Schlöder J. P. (2005), "A Real-time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control", *SIAM Journal on Control and Optimization*, Vol. 43, No. 5, pp. 1714 - 1736.
24. Diehl M., Findeisen R., Allgöwer F., Bock H. G., Schlöder J. P. (2005), "Nominal Stability of Real-time Iteration Scheme for Nonlinear Model Predictive Control", *IEEE Proc. - Control Theory Application*, Vol. 152, No. 3, pp. 296 - 308.

25. Ding B., Xi Y., Cychowski M. T., Mahony T. O' (2007), *A Synthesis Approach for Output Feedback Robust Constrained Model Predictive Control*, Automalica, In Press.
26. Ding B., Xi Y., Li S. (2004), "A Synthesis Approach of On-line Constrained Robust Model Predictive Control", *Automatica*, Vol. 40, 2004, pp. 163 - 167.
27. Dougherty D., Cooper D. (2003), "A practical Multiple Model Adaptive Strategy for Multivariable Model Predictive Control", *Control Engineering Practice*, Vol. 11, pp. 649 - 664.
28. Findeisen R., Diehl M., Burner T., Allgower F., Bock H. G., Schloder J. P. (2002), "Efficient output feedback nonlinear model predictive control", in: *Proceedings of the American Control Conference*, Anchorage, AK, pp. 4752 - 4757.
29. Fukushima H., Kim T. H., Sugie T. (2007), "Adaptive model predictive control for a class of constrained linear systems based on the comparison model", *Automatica*, 43, pp. 301 - 308.
30. Gu D., Hu H. (2005), "A Stabilizing Receding Horizon Regulator for Nonholonomic Mobile Robots", *IEEE Transactions on Robotics*, Vol. 21, No. 5, pp. 1022 - 1028.
31. Hanbay D., Turkoglu I., Demir Y. (2010), "Modeling switched circuits based wavelet decomposition and neural networks", *Journal of the Franklin Institute*, 347, pp. 607 - 617.
32. Imsland L., Findeisen R., Bullinger E., Allgower F., Foss B. A. (2003), "A note on stability, robustness and performance of output feedback nonlinear model predictive control", *Journal of Process Control*, 13, pp. 633 - 644.

33. Jadbabaie A., Yu J. J., Hauser (2001), "Unconstrained receding-horizon control of nonlinear systems", *IEEE Transaction on Automatic Control*, 46 (5), pp. 776 - 783.
34. Jalali A. A., Nadimi V. (2006), "A Survey on Robust Model Predictive Control from 1999 - 2006", *International Conference on Computational Intelligence for Modelling Control and Automation, Sydney, Australia*, pp. 207 - 212.
35. Kawati H., Kawati Y., Fujita M. (2001), "Visual Feedback Control of Planar Manipulators Based on Nonlinear Receding Horizon Control Approach", *IEEE American Control Conference*, pp. 763 - 768.
36. Kim H. J., Shim D. H., Sastry S. (2002), "Flying Robots Modelling, Control and Decision Making", *IEEE International Conference on Robotics and Automation*, pp. 66 - 71.
37. Kim T. H., Sugie T. (2005), "A new adaptive parameter estimation algorithm for robust constrained predictive control", in: *Proceedings of the IEEE Conference on Decision and Control, and European Control Conference, Seville, Spain*, pp. 284 - 289.
38. Li W. C., Biegler L. T., Multistep (1989), "Newton-type control strategies for constrained nonlinear processes", *Chemical Engineering Research and Design*, 67, pp. 562 - 577.
39. Lirraalde F., Nunes E. V. L., Hsu L., Wen J. T. (2003), "Mobile Robots Navigation using Sensor Fusion", *IEEE International Conference on Robotics and Automation*, pp. 458 - 463.
40. Mao W. J. (2003), "Robust Stabilization of Uncertain Time-varying Discrete Systems and Comments on An Improved Approach for Constrained Robust Model Predictive Control", *Automatica*, Vol. 39, pp. 1109 - 1112.

41. Mat Darus I. Z., Aldebrez F. M., Tokhi M. O. (2004), "Parametric Modelling of a Twin Rotor system Using Genetic Algorithms", *International Symposium on Control, Communications and Signal Processing*, pp. 115 - 118.
42. Mayne D. Q., Rakovic S. V., Findeisen R., Allgöwer F. (2006), "Robust Output Feedback Model Predictive Control of Constrained Linear Systems", *Automatica*, Vol. 42, pp. 1217 - 1222.
43. Mayne D. Q., Rawlings J. B., Rao C. V., Scokaert P. O. M. (2000), "Constrained model predictive control: stability and optimality", *Automatica*, 36, pp. 789 - 814.
44. Michalska H., Mayne D. Q. (1993), "Robust receding horizon control of constrained nonlinear systems", *IEEE Transaction on Automatic Control*, 38 (11), pp. 1623 - 1633.
45. Nocedal J., and Wright S. J. (1996), *Numerical Optimization*, Springer-New York.
46. O' Brien M., Ordys A. W. (2006), "Nonlinear Predictive Control of a Wastewater Model", *UKAAC Control, Glasgow, UK*, pp. 63 - 68.
47. Poignet P., Gautier M. (2000), "Nonlinear Model Predictive Control of a Robot Manipulator", *IEEE AMC Conference*, pp. 401 - 406.
48. Qin, S., & Badgwell, T. (2000), "An overview of nonlinear model predictive control applications", in *F. Allgöwer and A. Zheng (eds). Nonlinear Predictive Control, Birkhäuser*, pp. 369 - 393.
49. Qin, S., & Badgwell, T. (2003), "A Survey of industrial model prediction control technology", in *Control Engineering Practice*, Vol. 11, pp. 733 - 764.



50. Rahideh A., Shaheed M. H. (2011), "State model predictive control for a nonlinear system", *Journal of the Franklin Institute*, 348, pp. 1983 - 2004.
51. Rahideh A., Shaheed M. H. (2007), "Mathematical dynamic modelling of a twin rotor multiple input - multiple output system", *Proceedings of the IMechE, Part I. Journal of Systems and Control Engineering*, 221 (2007), pp. 89 - 101.
52. Rahideh A., Shaheed M. H. (2008), "Dynamic Modelling of a Twin rotor MIMO system using grey box approach", *International Symposium on Mechatronics & its applications, symposium Proceedings*, pp. 237 - 242.
53. Rahideh A., Shaheed M. H. (2008), "Neural network-based modelling of a two-degrees-of-freedom twin rotor multiple input, multiple output system using conjugate gradient learning algorithms", *P I Mech Eng G - J Aer*, vol. 222, (G6), pp. 757 - 771.
54. Rahideh A., Shaheed M. H. (2009), "Real Time Hybrid Fuzzy-PID Control of a Twin Rotor System", *IEEE International Conference on mechatronics*, Vols 1 and 2, pp. 327 - 332.
55. Rahideh A., Shaheed M. H. (2012), "Constrained output feedback model predictive control for nonlinear systems", *Control Engineering Practice*, 20, pp. 431 - 443.
56. Rao R. R., Aufderheide B., Bequette B. W. (2003), "Experimental studies on multiple-model predictive control for automated regulation of hemodynamic variables", *IEEE Transaction on Biomedical Engineering*, 50 (3), pp. 277 - 288.
57. Rawlings J. B., Muske K. R. (1993), "The stability of constrained receding horizon control", *IEEE Transaction on Automatic Control*, 38 (10), pp. 1512 - 1516.

58. Scokaert P. O. M., Mayne D. Q. (1998), "Min-Max Feedback Model Predictive Control for Constrained Linear Systems", *IEEE Transactions on Automatic Control*, Vol. 43, No. 8, pp. 1136 - 1142.
59. Scokaert P. O. M., Mayne D. Q., Rawlings J. B. (1999), "Suboptimal model predictive control (feasibility implies stability)", *IEEE Transaction on Automatic Control*, 44 (3), pp. 648 - 654.
60. Shaheed M. H. (2004), "Performance analysis of 4 types of conjugate gradient algorithm in the nonlinear dynamic modelling of a TRMS using feedforward neural networks", *IEEE International Conference on Systems, man and cybernetics*, pp. 5985 - 5990.
61. Shim D. H., Kim H. J., Sastry S. (2003), "Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots", *IEEE Conference on Decision and Control*, pp. 3621 - 3626.
62. Tellez F. O., Loukianov A. G., Sanchez E. N., Corrochano E. J. B. (2010), "Decentralized neural identification and control for uncertain nonlinear systems: application to planar robot", *Journal of the Franklin Institute*, 347, pp. 1015 - 1034.
63. VandenBoom T. J. J., De Schutter B., Schullerus G., Krebs V. (2004), "Adaptive model predictive control for max-plus-linear discrete event input-output systems", *IEE Proc .- Control Theory Applications*, Vol. 151, No. 3, pp. 339 - 346.
64. Veronica Adetola, Darryl DeHaan, Martin Guay (2009), "Adaptive Model Predictive Control for constrained nonlinear systems", *Systems & Control Letters*, 58, pp. 320 - 326.
65. Vivas A., Poignet P., Pierrot F. (2003), "Predictive Functional Control for a Parallel Robot", *IEEE International Conference on Intelligent Robotic and systems*, pp. 2785 - 2790.

66. Wan Z., Kothare M. V. (2002), "Robust Output Feedback Model Predictive Control using Off-line Linear Matrix Inequality", *Journal of Process Control*, Vol. 12, pp. 763 - 774.
67. Wan Z., Kothare M. V. (2003), "An Efficient Off-line Formulation of Robust Model Predictive Control using Linear Matrix Inequalities", *Automatica*, Vol. 39, pp. 837 - 846.
68. Wan Z., Kothare M. V. (2004), "Efficient Scheduled Stabilizing Output Feedback Model Predictive Control for Constrained Nonlinear Systems", *IEEE Transactions on Automatic Control*, Vol. 49, No. 7, pp. 1172 - 1177.
69. Wang Y. J., Rawlings J. B. (2004), "A New Robust Model Predictive Control Method I: Theory and Computation", *Journal of Process Control*, Vol. 14, pp. 231 - 247.
70. Wang Y., Boyd S., (2010), "Fast model Predictive Control using online optimization", *IEEE Transactions on Control Systems Technology*, 18(2), pp. 267 - 278.
71. Wroblewski W. (2004), "Implementation of a Model Predictive Control Algorithms for 6 DOF Manipulator-Simulation Results", *IEEE 4th International Workshop on Robotic and Control*, pp. 209 - 212.
72. Zhang B., Zhang W. (2006), "Adaptive predictive functional control of a class of nonlinear systems", *ISA Transactions*, 45 (2), pp. 175 - 183.

## PHỤ LỤC

1. Code chương trình cài đặt bộ điều khiển MPC sử dụng thuật toán SQP.

### TRMS\_Controller

```
function [sys,x0,str,ts,simStateCompliance] = TRMS_Controller(t,x,u,flag)
global Up0;
global dU0;
switch flag,
    case 0,
        [sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes;
    case 2,
sys=[];
    case 3,
        sys = mdlOutputs(t,x,u);
    case 9,
        sys = [];
    otherwise
        DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes
global Up0;
global dU0;
Up0=[0;0];
dU0=zeros(30,1);
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 2;
sizes.NumInputs = 8;
sizes.DirFeedthrough = 1;
```

```

sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = [];
str = [];
ts = [0.2 0];      % Sample period of 0.1 seconds (10Hz)
simStateCompliance = 'DefaultSimState';

function sys = mdlOutputs(t,x,u)
global Up0;
global dU0;
lt=2.82e-1;      % Chieu dai cua phan canh tay don duoi
lm=2.46e-1;      % Chieu dai cua phan canh tay don chinh
Jv=5.26e-2;      % Mo men quan tinh cua truc ngang
D=5.52877e-2;
E=5.85764e-3;
F=5.85077e-3;
g=9.81;         % Gia toc trong trung
A=9.82347e-2;
B=1.135659e-1;
C=2.21788e-2;
H=4.94734e-2;
Jtr=3.1432e-5;  % Mo men quan tinh cua dong co duoi/tai
Rah=8;
kthp=5e-8;      % Th=kthp*sign(wh)*(wh)^2
kthn=4.41e-8;   % Th=kthn*sign(wh)*(wh)^2
Btr=2.3e-5;     % He so ma sat nhot cua rotor duoi
Jmr=2.0098e-4; % Mo men quan tinh cua dong co chinh/tai
Rav=8;
ktvp=5.6e-7;    % Th=ktvp*sign(wv)*(wv)^2

```

```

ktvn=5.1e-7;      % Th=ktvn*sign(wv)*(wv)^2
Bmr=2.97e-5;     % He so ma sat nhot cua rotor chinh
av0=-6.048e-1;   % Goc chao doc ban dau
kfhp=2.1377e-6; % Fh(wh)=kfhp*sign(wh)*(wh)^2
kfhn=1.9056e-6; % Fh(wh)=kfhn*sign(wh)*(wh)^2
kfvp=1.9506e-5; % Fv(wv)=kfvp*sign(wv)*(wv)^2
kfvn=1.1012e-5; % Fv(wv)=kfvn*sign(wv)*(wv)^2
kvfh=4.91e-3;    % Ma sat nhot theo phuong ngang
kvfv=5.47e-3;    % Ma sat nhot theo phuong thang dung
kcfh=3.96e-5;
kcfv=1.5e-4;
kchp=5.60e-3;    % He so ma sat (cable force coefficient)
kchn=5.60e-3;    % He so ma sat (cable force coefficient)
km=0.00023;      % Anh huong cua rotor chinh len goc theo phuong ngang
kt=0.000026;     % Anh huong cua rotor duoi len goc theo phuong thang dung
TC=0.0202;       % He so mo men xoan (mo men xoan khong doi)
k1=8.5;          % He so giao dien theo phuong thang dung (doc)
k2=6.5;          % He so giao dien ngang
kg=0.2;
Ts=0.2;
Np=20;
Nc=15;
nx=6;
nu=2;
ny=2;
Q=[0.09 0;0 1.5]; %He so trong so sai so
R=[0.0125 0;0 0.0125]; %He so trong so no luc dieu khien (control effort)
Cu=[-2.5 2.5;-2.5 2.5]; %Rang buoc dau vao doi tuong
Cdu=[-2 2;-2 2]; %Rang buoc dau vao thay doi

```

```

Cy=[-1.5 1.5;-1.5 1.5]; %Rang buoc dau ra doi tuong
al=[0 0;0 0]; %Cac he so quy dao tham chieu (dat)
Up=Up0;
Xp=u(1:6);
ref=u(7:8);
d=zeros(2,1);
dx=zeros(6,1);
tsim=100;
%Cac tham so mo hinh khong gian trang thai tuyen tinh roi rac
AA=zeros(nx*Np,nx);
BB=zeros(nx*Np,nu);
CC=zeros(ny*Np,nx);
Upf=Up;
Xpf=Xp;
Xint1(:,1)=Xp;
Xint2(:,1)=Xp;
for i=1:Np
    [Ad,Bd,Cd,Dd]=TRMSlin(Xp,Up);
    AA(nx*(i-1)+1:nx*i,:)=Ad;
    BB(nx*(i-1)+1:nx*i,:)=Bd;
    CC(ny*(i-1)+1:ny*i,:)=Cd;
    Xint1(:,i+1)=Ad(:,:)*...
    Xint1(:,i)+Bd(:,:)*Up;
    Uhp=Up(1);
    Uvp=Up(2);
    %Cap nhap bien trang thai o moi gia doan
    whp=Xp(1);
    Shp=Xp(2);
    ahp=Xp(3);

```

```

wvp=Xp(4);
Svp=Xp(5);
avp=Xp(6);
if whp>=0
    whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp-kthp*whp^2)/Jtr;
else
    whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp+kthn*whp^2)/Jtr;
end
if wvp>=0
    wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-Bmr*wvp-
    ktvp*wvp^2)/Jmr;
else
    wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-
    Bmr*wvp+ktvn*wvp^2)/Jmr;
end
if whp>=0
    Shn=Shp+Ts*(lt*kfhp*whp^2*cos(avp+av0)-kvfh*Shp-...
    kchp*ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
    Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+E*...
    (sin(avp+av0))^2+F)^2;
else
    Shn=Shp+Ts*(-lt*kfhn*whp^2*cos(avp+av0)-kvfh*Shp-...
    kchp*ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
    Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+E*...
    (sin(avp+av0))^2+F)^2;
end
if wvp>=0
    Svn=Svp+Ts*(lm*kfvp*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
    g*((A-B)*cos(avp+av0)-C*sin(avp+av0))/Jv;
else

```



```

Svn=Svp+Ts*(-lm*kfvn*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
    g*((A-B)*cos(avp+av0)-C*sin(avp+av0))/Jv;
end
ahn=ahp+Ts*Shp+Ts*km*wvp*cos(avp+av0)/...
    (D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F);
ahint(i)=ahn;
avn=avp+Ts*Svp+Ts*kt*whp/Jv;
avint(i)=avn;

Xp=[whn;Shn;ahn;wvn;Svn;avn];
Xint2(:,i+1)=Xp;
% Cap nhat cac dau vao o moi giai doan
if i<=Nc
    Up=Up+dU0((i-1)*nu+1:i*nu,1);
end
end
Cz=Cd(1:2,:);
dU1= MyMPC9(AA,BB,Cz,Dd,Np,Nc,Q,R,ref,Cu,...
    Cdu,Cy,Xpf,Upf,d,al,dx,dU0,t);
Out1=Xint1;
Out2=Xint2;
sys=Up0;
%end mdlOutputs

HuongTRMS
function [sys,x0,str,ts,simStateCompliance]=HuongTRMS(t,x,u,flag)
switch flag
case 0
    [sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes();

```

```

case 1
    sys = mdlDerivatives(t,x,u);
case {2,9}
    sys = [];
case 3
    sys = mdlOutputs(t,x,u);
otherwise
    DAStudio.error('Simulink:blocks:unhandledFla9.81', num2str(flag));
end
function [sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes()
sizes = simsizes;
sizes.NumContStates = 6;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 6;
sizes.NumInputs = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 2;
sys = simsizes(sizes);
str = [];
x0=zeros(6,1);
ts = [0 0;0.2 0]; % sample time: [period, offset]
simStateCompliance = 'DefaultSimState';
function sys = mdlDerivatives(t,x,u)
lt=2.82e-1; % Chieu dai cua phan canh tay don duoi
lm=2.46e-1; % Chieu dai cua phan canh tay don chinh
Jv=5.26e-2; % Mo men quan tinh cua truc ngang
D=5.52877e-2;
E=5.85764e-3;
F=5.85077e-3;

```

```

g=9.81;           % Gia toc trong trung
A=9.82347e-2;
B=1.135659e-1;
C=2.21788e-2;
H=4.94734e-2;
Jtr=3.1432e-5;   % Mo men quan tinh cua dong co duoi/tai
Rah=8;
kthp=5e-8;       % Th=kthp*sign(wh)*(wh)^2
kthn=4.41e-8;   % Th=kthn*sign(wh)*(wh)^2
Btr=2.3e-5;     % He so ma sat nhot cua rotor duoi
Jmr=2.0098e-4;  % Mo men quan tinh cua dong co chinh/tai
Rav=8;
ktvp=5.6e-7;    % Th=ktvp*sign(wv)*(wv)^2
ktvn=5.1e-7;    % Th=ktvn*sign(wv)*(wv)^2
Bmr=2.97e-5;    % He so ma sat nhot cua rotor chinh
av0=-6.048e-1;  % Goc chao doc ban dau
kfhp=2.1377e-6; % Fh(wh)=kfhp*sign(wh)*(wh)^2
kfhn=1.9056e-6; % Fh(wh)=kfhn*sign(wh)*(wh)^2
kfvp=1.9506e-5; % Fv(wv)=kfvp*sign(wv)*(wv)^2
kfvn=1.1012e-5; % Fv(wv)=kfvn*sign(wv)*(wv)^2
kvfh=4.91e-3;   % Ma sat nhot theo phuong ngang
kvfv=5.47e-3;   % Ma sat nhot theo phuong thang dung
kcfh=3.96e-5;
kcfv=1.5e-4;
kchp=5.60e-3;   % He so ma sat (cable force coefficient)
kchn=5.60e-3;   % He so ma sat (cable force coefficient)
km=0.00023;     % Anh huong cua rotor chinh len goc theo phuong ngang
kt=0.000026;    % Anh huong cua rotor duoi len goc theo phuong thang dung
TC=0.0202;      % He so mo men xoan (mo men xoan khong doi)

```

```

k1=8.5;           % He so giao dien theo phuong thang dung (doc)
k2=6.5;           % He so giao dien ngang
kg=0.2;
Ts=0.2;
Uh=u(1);
Uv=u(2);
wh=x(1);
Sh=x(2);
ah=x(3);
wv=x(4);
Sv=x(5);
av=x(6);
Jh=D*(cos(av+av0))^2+E*(sin(av+av0))^2+F;
% Tinh dah , dav
dah=Sh+km*wv*cos(av+av0)/Jh;   % dah=Omegah
dav=Sv+kt*wh/Jv;               % dav=Omegav
% Tinh dwh, dSh
if wh>=0
    dwh=(TC*(k2*Uh-TC*wh)/Rah-Btr*wh-kthp*wh^2)/Jtr;%%%%%%%%
    dSh=(lt*kfhp*wh^2*cos(av+av0)-(kvfh*dah+kcfh*sign(dah))-
kchp*ah)/Jh;
else
    dwh=(TC*(k2*Uh-TC*wh)/Rah-Btr*wh+kthn*wh^2)/Jtr;%%%%%%%%
    dSh=(-lt*kfhn*wh^2*cos(av+av0)-(kvfh*dah+kcfh*sign(dah))-
kchp*ah)/Jh;
end
% Tinh dwv, dSv
if wv>=0
    dwv=(TC*(k1*Uv-TC*wv)/Rav-Bmr*wv-ktvp*wv^2)/Jmr;%%%%%%%%

```

```

    dSv= (kfv*ww^2*(lm+kg*dah*cos(av+av0))-
(kvfv*dav+kcfv*sign(dav))+...
        g*((A-B)*cos(av+av0)-C*sin(av+av0))-
0.5*dah^2*H*sin(2*(av+av0)))/Jv;
else
    dwv=(TC*(k1*Uv-TC*ww)/Rav-Bmr*ww+ktvn*ww^2)/Jmr;%%%%%%%%%
    dSv= (-kfv*ww^2*(lm+kg*dah*cos(av+av0))-
(kvfv*dav+kcfv*sign(dav))+...
        g*((A-B)*cos(av+av0)-C*sin(av+av0))-
0.5*dah^2*H*sin(2*(av+av0)))/Jv;
end

```

### TRMSlin.m

```

function [As,Bs,Cs,Ds]=TRMSlin(Xp,Up)
whp=Xp(1);
Shp=Xp(2);
ahp=Xp(3);
wvp=Xp(4);
Svp=Xp(5);
avp=Xp(6);
Uhp=Up(1);
Uvp=Up(2);

lt=2.82e-1;           % Chieu dai cua phan canh tay don duoi
lm=2.46e-1;           % Chieu dai cua phan canh tay don chinh
Jv=5.26e-2;           % Mo men quan tinh cua truc ngang
D=5.52877e-2;
E=5.85764e-3;
F=5.85077e-3;
g=9.81;               % Gia toc trong trung
A=9.82347e-2;

```

```

B=1.135659e-1;
C=2.21788e-2;
H=4.94734e-2;
Jtr=3.1432e-5;    % Mo men quan tinh cua dong co duoi/tai
Rah=8;
kthp=5e-8;       % Th=kthp*sign(wh)*(wh)^2
kthn=4.41e-8;   % Th=kthn*sign(wh)*(wh)^2
Btr=2.3e-5;     % He so ma sat nhot cua rotor duoi
Jmr=2.0098e-4;  % Mo men quan tinh cua dong co chinh/tai
Rav=8;
ktvp=5.6e-7;    % Th=ktvp*sign(wv)*(wv)^2
ktvn=5.1e-7;    % Th=ktvn*sign(wv)*(wv)^2
Bmr=2.97e-5;    % He so ma sat nhot cua rotor chinh
av0=-6.048e-1;  % Goc chao doc ban dau
kfhp=2.1377e-6; % Fh(wh)=kfhp*sign(wh)*(wh)^2
kfhn=1.9056e-6; % Fh(wh)=kfhn*sign(wh)*(wh)^2
kfvp=1.9506e-5; % Fv(wv)=kfvp*sign(wv)*(wv)^2
kfvn=1.1012e-5; % Fv(wv)=kfvn*sign(wv)*(wv)^2
kvfh=4.91e-3;   % Ma sat nhot theo phuong ngang
kvfv=5.47e-3;   % Ma sat nhot theo phuong thang dung
kcfh=3.96e-5;
kcfv=1.5e-4;
kchp=5.60e-3;   % He so ma sat (cable force coefficient)
kchn=5.60e-3;   % He so ma sat (cable force coefficient)
km=0.00023;     % Anh huong cua rotor chinh len goc theo phuong ngang
kt=0.000026;    % Anh huong cua rotor duoi len goc theo phuong thang dung
TC=0.0202;      % He so mo men xoan (mo men xoan khong doi)
k1=8.5;         % He so giao dien theo phuong thang dung (doc)
k2=6.5;         % He so giao dien ngang

```

```

kg=0.2;
Ts=0.2;

As=zeros(6,6);
Bs=zeros(6,2);
Cs=zeros(2,6);
Ds=zeros(2,2);
dx=zeros(6,1);
if whp>=0
    As(1,1)=1-Ts*(Btr+kthp*whp+TC^2/Rah)/Jtr;
    As(2,1)=Ts*lt*kfhp*whp*cos(avp+av0)/DEN;
else
    As(1,1)=1-Ts*(Btr-kthn*whp+TC^2/Rah)/Jtr;
    As(2,1)=-Ts*lt*kfhn*whp*cos(avp+av0)/DEN;
end
As(2,2)=1-Ts*kvfh/DEN;

if ahp>=0
    As(2,3)=-Ts*kchp/DEN;
else
    As(2,3)=-Ts*kchn/DEN;
end
As(2,4)=-Ts*kvfh*km*cos(avp+av0)/DEN^2;
As(3,2)=Ts;
As(3,3)=1;
As(3,4)=Ts*km*cos(avp+av0)/DEN;
if wvp>=0
    As(4,4)=1-Ts*(Bmr+ktvp*wvp+TC^2/Rav)/Jmr;
    As(5,4)=Ts*lm*kfvp*wvp/Jv;

```

```

else
    As(4,4)=1-Ts*(Btr-ktvn*wvp+TC^2/Rav)/Jmr;
    As(5,4)=-Ts*lm*kfvn*wvp/Jv;
end
As(5,1)=-Ts*kfv*kt/(Jv^2);
As(5,5)=1-Ts*kfv/Jv;
if avp==0
    As(5,6)=Ts*g*((A-B)*cos(avp+av0)-C*sin(avp+av0))/Jv;
else
    As(5,6)=Ts*g*((A-B)*cos(avp+av0)-C*sin(avp+av0))/(Jv*avp);
end
As(6,1)=Ts*kt/Jv;
As(6,5)=Ts;
As(6,6)=1;
Bs(1,1)=Ts*TC*k2/(Jtr*Rah);
Bs(4,2)=Ts*TC*k1/(Jmr*Rav);
Cs(1,3)=1;
Cs(2,6)=1;

```

2. Code chương trình cài đặt bộ điều khiển dự báo trên nền phương pháp biến phân.

### TRMS\_1

```

function [sys,x0,str,ts,simStateCompliance] = TRMS_1(t,x,u,flag)
global out;
global K_lqr;
global xeq;
global ueq;
switch flag,
    case 0,
        [sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes;

```



```

case 2,
    sys=[];
case 3,
    sys = mdlOutputs(t,u);
case 9,
    sys = [];
otherwise
    DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes
global out;
global K_lqr;
global xeq;
global ueq;
xeq=[0;0;0;0;0;0];
ueq=[0;0];
K_lqr=[0 0 0 0 0 0;0 0 0 0 0 0];
out=[0;0];
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 2;
sizes.NumInputs = 8;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = [];
str = [];
ts = [0 0];

```

```

% specify that the simState for this s-function is same as the default
simStateCompliance = 'DefaultSimState';
function sys = mdlOutputs(t,u)
global out;
global K_lqr;
global xeq;
global ueq;
ref=[0 0 0 0 0 1 0;0 0 0 0 0 0 1]*u;
    x=[1 0 0 0 0 0 0;0 1 0 0 0 0 0 ;0 0 1 0 0 0 0;0 0 0 1 0 0 0;0 0 0 0 1 0
0 0;0 0 0 0 0 1 0 0]*u;
    [As Bs Cs Ds]=TRMS_tuyentinh(x,out);
% ref=[-1;-0.5];
    lamtron=(fix(t/0.1))*0.1;
if ((t-lamtron)==0)&(t>0.1)
kchp=5.60e-3;    % He so ma sat (cable force coefficient)
kchn=5.60e-3;    % He so ma sat (cable force coefficient)
lt=2.82e-1;      % Chieu dai cua phan canh tay don duoi
av0=-6.048e-1;   % Goc chao doc ban dau
kfhp=2.1377e-6;  % Fh(wh)=kfhp*sign(wh)*(wh)^2
kfhn=1.9056e-6;  % Fh(wh)=kfhn*sign(wh)*(wh)^2
kt=0.000026;    % Anh huong cua rotor duoi len goc theo phuong thang dung
Jv=5.26e-2;     % Mo men quan tinh cua truc ngang
gi=9.81;        % Gia toc trong trung
Ai=9.82347e-2;
Bi=1.135659e-1;
Ci=2.21788e-2;
lm=2.46e-1;     % Chieu dai cua phan canh tay don chinh
kfvp=1.9506e-5; % Fv(wv)=kfvp*sign(wv)*(wv)^2
kfvn=1.1012e-5; % Fv(wv)=kfvn*sign(wv)*(wv)^2
km=0.00023;    % Anh huong cua rotor chinh len goc theo phuong ngang

```

```

Di=5.52877e-2;
Ei=5.85764e-3;
Fi=5.85077e-3;
% Tinh xeq, ueq
if ref(1)>=0
    FhNp=kchp*ref(1)/(lt*cos(ref(2)+av0));
    whNp=sqrt(abs(FhNp/kfhp));
else
    FhNp=kchn*ref(1)/(lt*cos(ref(2)+av0));
    whNp=-sqrt(abs(FhNp/kfhn));
end
SvNp=-kt*whNp/Jv;
FvNp=-gi*((Ai-Bi)*cos(ref(2)+av0)-Ci*sin(ref(2)+av0))/lm;
if FvNp>=0
    wvNp=sqrt(abs(FvNp/kfvp));
else
    wvNp=-sqrt(abs(FvNp/kfvn));
end
ShNp=-km*wvNp*cos(ref(2)+av0)/(Ei*(sin(ref(2)+av0))^2+...
    Di*(cos(ref(2)+av0))^2+Fi);
xeq=[whNp;ShNp;ref(1);wvNp;SvNp;ref(2)];
% Tinh ueq
k1=8.5;           % He so giao dien theo phuong thang dung (doc)
k2=6.5;           % He so giao dien ngang
TC=0.0202;        % He so mo men xoan (mo men xoan khong doi)
Rah=8;
Btr=2.3e-5;        % He so ma sat nhot cua rotor duoi
Bmr=2.97e-5;      % He so ma sat nhot cua rotor chinh
ktvp=5.6e-7;      % Th=ktvp*sign(wv)*(wv)^2

```

```

ktvn=5.1e-7;      % Th=ktvn*sign(wv)*(wv)^2
kthp=5e-8;       % Th=kthp*sign(wh)*(wh)^2
kthn=4.41e-8;    % Th=kthn*sign(wh)*(wh)^2
Rav=8;
if whNp>=0
    ueq(1,1)=(Btr+kthp*whNp+TC^2/Rah)*whNp*Rah/(k2*TC);
else
    ueq(1,1)=(Btr-kthn*whNp+TC^2/Rah)*whNp*Rah/(k2*TC);
end
if wvNp>=0
    ueq(2,1)=(Bmr+ktvp*wvNp+TC^2/Rav)*wvNp*Rav/(k1*TC);
else
    ueq(2,1)=(Bmr-ktvn*wvNp+TC^2/Rav)*wvNp*Rav/(k1*TC);
end
delta=x-xeq;
out=ueq-K_lqr*delta;
if out(1,1)>=10
    out(1,1)=10;
else if out(1,1)<-10
    out(1,1)=-10;
end
end
if out(2,1)>=10
    out(2,1)=10;
else if out(2,1)<-10
    out(2,1)=-10;
end
end
sys=[out];

```

HuongTRMS2

function [sys,x0,str,ts,simStateCompliance]=HuongTRMS2(t,x,u,flag)

switch flag

case 0

[sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes();

case 1

sys = mdlDerivatives(t,x,u);

case {2,9}

sys = [];

case 3

sys = mdlOutputs(t,x,u);

otherwise

DAStudio.error('Simulink:blocks:unhandledFla9.81', num2str(flag));

end

function [sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes()

sizes = simsizes;

sizes.NumContStates = 6;

sizes.NumDiscStates = 0;

sizes.NumOutputs = 6;

sizes.NumInputs = 2;

sizes.DirFeedthrough = 0;

sizes.NumSampleTimes = 1;

sys = simsizes(sizes);

str = [];

x0=zeros(6,1);

% x0(6,1)= -0.6048;

ts = [0 0]; % sample time: [period, offset]

simStateCompliance = 'DefaultSimState';

function sys = mdlDerivatives(t,x,u)

```

lt=2.82e-1;      % Chieu dai cua phan canh tay don duoi
lm=2.46e-1;      % Chieu dai cua phan canh tay don chinh
Jv=5.26e-2;      % Mo men quan tinh cua truc ngang
D=5.52877e-2;
E=5.85764e-3;
F=5.85077e-3;
g=9.81;          % Gia toc trong trung
A=9.82347e-2;
B=1.135659e-1;
C=2.21788e-2;
H=4.94734e-2;
Jtr=3.1432e-5;   % Mo men quan tinh cua dong co duoi/tai
Rah=8;
kthp=5e-8;        % Th=kthp*sign(wh)*(wh)^2
kthn=4.41e-8;     % Th=kthn*sign(wh)*(wh)^2
Btr=2.3e-5;       % He so ma sat nhot cua rotor duoi
Jmr=2.0098e-4;   % Mo men quan tinh cua dong co chinh/tai
Rav=8;
ktvp=5.6e-7;      % Th=ktvp*sign(wv)*(wv)^2
ktvn=5.1e-7;      % Th=ktvn*sign(wv)*(wv)^2
Bmr=2.97e-5;      % He so ma sat nhot cua rotor chinh
av0=-6.048e-1;   % Goc chao doc ban dau
kfhp=2.1377e-6;   % Fh(wh)=kfhp*sign(wh)*(wh)^2
kfhn=1.9056e-6;   % Fh(wh)=kfhn*sign(wh)*(wh)^2
kfvp=1.9506e-5;   % Fv(wv)=kfvp*sign(wv)*(wv)^2
kfvn=1.1012e-5;   % Fv(wv)=kfvn*sign(wv)*(wv)^2
kvfh=4.91e-3;     % Ma sat nhot theo phuong ngang
kvfv=5.47e-3;     % Ma sat nhot theo phuong thang dung
kcfh=3.96e-5;

```

```

kcfv=1.5e-4;
kchp=5.60e-3;    % He so ma sat (cable force coefficient)
kchn=5.60e-3;    % He so ma sat (cable force coefficient)
km=0.00023;      % Anh huong cua rotor chinh len goc theo phuong ngang
kt=0.000026;     % Anh huong cua rotor duoi len goc theo phuong thang dung
TC=0.0202;       % He so mo men xoan (mo men xoan khong doi)
k1=8.5;          % He so giao dien theo phuong thang dung (doc)
k2=6.5;          % He so giao dien ngang
kg=0.2;
Ts=0.2;

Uh=u(1);
Uv=u(2);
wh=x(1);
Sh=x(2);
ah=x(3);
wv=x(4);
Sv=x(5);
av=x(6);
Jh=D*(cos(av+av0))^2+E*(sin(av+av0))^2+F;
% Tinh dah , dav
dah=Sh+km*wv*cos(av+av0)/Jh;    % dah=Omegah
dav=Sv+kt*wh/Jv;                % dav=Omegav
% Tinh dwh, dSh
if wh>=0
    dwh=(TC*(k2*Uh-TC*wh)/Rah-Btr*wh-kthp*wh^2)/Jtr;%%%%%%%%%
    dSh=(lt*kfhp*wh^2*cos(av+av0)-(kvfh*dah+kcfh*sign(dah))-
kchp*ah)/Jh;
else
    dwh=(TC*(k2*Uh-TC*wh)/Rah-Btr*wh+kthn*wh^2)/Jtr;%%%%%%%%%

```

```

    dSh=(-lt*kfhn*wh^2*cos(av+av0)-(kvfh*dah+kcfh*sign(dah))-
kchp*ah)/Jh;
end
% Tinh dwv, dSv
if wv>=0
    dwv=(TC*(k1*Uv-TC*wv)/Rav-Bmr*wv-ktvp*wv^2)/Jmr;%%%%%%%%%
    dSv= (kfvp*wv^2*(lm+kg*dah*cos(av+av0))-
(kfvv*dav+kcfv*sign(dav))+...
    g*((A-B)*cos(av+av0)-C*sin(av+av0))-
0.5*dah^2*H*sin(2*(av+av0)))/Jv;
else
    dwv=(TC*(k1*Uv-TC*wv)/Rav-Bmr*wv+ktvn*wv^2)/Jmr;%%%%%%%%%
    dSv= (-kfvn*wv^2*(lm+kg*dah*cos(av+av0))-
(kfvv*dav+kcfv*sign(dav))+...
    g*((A-B)*cos(av+av0)-C*sin(av+av0))-
0.5*dah^2*H*sin(2*(av+av0)))/Jv;
end
sys=[dwh;dSh;dah;dwv;dSv;dav];
function sys = mdlOutputs(t,x,u)
sys = x;

```

3. Code chương trình EKF cài đặt bộ điều khiển dự báo cho đối tượng TRMS chạy thực nghiệm.

```

function Xe=mdlKalman(Uhp,Uvp,ah,av)
%===== Cac hang so =====
lt=2.82e-1;      % Chieu dai cua phan canh tay don duoi
lm=2.46e-1;      % Chieu dai cua phan canh tay don chinh
Jv=5.26e-2;      % Mo men quan tinh cua truc ngang
D=5.52877e-2;
E=5.85764e-3;
F=5.85077e-3;

```



```

g=9.81;           % Gia toc trong trung
A=9.82347e-2;
B=1.135659e-1;
C=2.21788e-2;
H=4.94734e-2;
Jtr=3.1432e-5;   % Mo men quan tinh cua dong co duoi/tai
Rah=8;
kthp=5e-8;       % Th=kthp*sign(wh)*(wh)^2
kthn=4.41e-8;    % Th=kthn*sign(wh)*(wh)^2
Btr=2.3e-5;      % He so ma sat nhot cua rotor duoi
Jmr=2.0098e-4;  % Mo men quan tinh cua dong co chinh/tai
Rav=8;
ktvp=5.6e-7;     % Th=ktvp*sign(wv)*(wv)^2
ktvn=5.1e-7;     % Th=ktvn*sign(wv)*(wv)^2
Bmr=2.97e-5;    % He so ma sat nhot cua rotor chinh
av0=-6.048e-1;  % Goc chao doc ban dau
kfhp=2.1377e-6; % Fh(wh)=kfhp*sign(wh)*(wh)^2
kfhn=1.9056e-6; % Fh(wh)=kfhn*sign(wh)*(wh)^2
kfvp=1.9506e-5; % Fv(wv)=kfvp*sign(wv)*(wv)^2
kfvn=1.1012e-5; % Fv(wv)=kfvn*sign(wv)*(wv)^2
kvfh=4.91e-3;   % Ma sat nhot theo phuong ngang
kvfv=5.47e-3;   % Ma sat nhot theo phuong thang dung
kcfh=3.96e-5;
kcfv=1.5e-4;
kchp=5.60e-3;   % He so ma sat (cable force coefficient)
kchn=5.60e-3;   % He so ma sat (cable force coefficient)
km=0.00023;     % Anh huong cua rotor chinh len goc theo phuong ngang
kt=0.000026;    % Anh huong cua rotor duoi len goc theo phuong thang dung
TC=0.0202;      % He so mo men xoan (mo men xoan khong doi)

```

```

k1=8.5;           % He so giao dien theo phuong thang dung (doc)
k2=6.5;           % He so giao dien ngang
kg=0.2;
Ts=0.2;
nx=6;
nu=2;
ny=2;
%===== Cac bien so =====
persistent Xp Pk
if isempty(Xp)
    Xp=zeros(nx,1);
end
if isempty(Pk)
    Pk=zeros(nx,nx);
end
ahp=Xp(3,1);
avp=Xp(6,1);
whp=Xp(1,1);
Shp=Xp(2,1);
wvp=Xp(4,1);
Svp=Xp(5,1);
Q=eye(nx,nx)*0.01;
R=eye(ny,ny)*0.01;
As=zeros(nx,nx);
Bs=zeros(nx,nu);
Cs=[0 0 1 0 0 0;0 0 0 0 0 1];
if whp>=0
    whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp-kthp*whp^2)/Jtr;
    Shn=Shp+Ts*(lt*kfhp*whp^2*cos(avp+av0)-kvfh*Shp-...

```

$$\begin{aligned} & kchp*ahp)/(D*(\cos(avp+av0))^2+E*(\sin(avp+av0))^2+F)-... \\ & Ts*kvfh*km*wvp*\cos(avp+av0)/(D*(\cos(avp+av0))^2+E*... \\ & (\sin(avp+av0))^2+F)^2; \end{aligned}$$

else

$$\begin{aligned} whn &= whp + Ts*(TC*(k2*Uhp - TC*whp)/Rah - Btr*whp + kthn*whp^2)/Jtr; \\ Shn &= Shp + Ts*(-lt*kfhn*whp^2*\cos(avp+av0) - kvfh*Shp - ... \\ & kchp*ahp)/(D*(\cos(avp+av0))^2+E*(\sin(avp+av0))^2+F)-... \\ & Ts*kvfh*km*wvp*\cos(avp+av0)/(D*(\cos(avp+av0))^2+E*... \\ & (\sin(avp+av0))^2+F)^2; \end{aligned}$$

end

if wvp >= 0

$$\begin{aligned} wvn &= wvp + Ts*(TC*(k1*Uvp - TC*wvp)/Rav - Bmr*wvp - \\ & ktvp*wvp^2)/Jmr; \\ Svn &= Svp + Ts*(lm*kfvp*wvp^2 - kvfv*Svp - kvfv*kt*whp/Jv + ... \\ & g*((A-B)*\cos(avp+av0) - C*\sin(avp+av0)))/Jv; \end{aligned}$$

else

$$\begin{aligned} wvn &= wvp + Ts*(TC*(k1*Uvp - TC*wvp)/Rav - \\ & Bmr*wvp + ktvn*wvp^2)/Jmr; \\ Svn &= Svp + Ts*(-lm*kfvn*wvp^2 - kvfv*Svp - kvfv*kt*whp/Jv + ... \\ & g*((A-B)*\cos(avp+av0) - C*\sin(avp+av0)))/Jv; \end{aligned}$$

end

$$\begin{aligned} ahn &= ahp + Ts*Shp + Ts*km*wvp*\cos(avp+av0)/... \\ & (D*(\cos(avp+av0))^2+E*(\sin(avp+av0))^2+F); \\ avn &= avp + Ts*Svp + Ts*kt*whp/Jv; \\ DEN &= D*(\cos(avp+av0))^2+E*(\sin(avp+av0))^2+F; \end{aligned}$$

if whp >= 0

$$\begin{aligned} As(1,1) &= 1 - Ts*(Btr + kthp*whp + TC^2/Rah)/Jtr; \\ As(2,1) &= Ts*lt*kfhp*whp*\cos(avp+av0)/DEN; \end{aligned}$$

else

$$As(1,1) = 1 - Ts*(Btr - kthn*whp + TC^2/Rah)/Jtr;$$

```

    As(2,1)=-Ts*lt*kfhn*whp*cos(avp+av0)/DEN;
end
As(2,2)=1-Ts*kvfh/DEN;
if ahp>=0
    As(2,3)=-Ts*kchp/DEN;
else
    As(2,3)=-Ts*kchn/DEN;
end
As(2,4)=-Ts*kvfh*km*cos(avp+av0)/DEN^2;
As(3,2)=Ts;
As(3,3)=1;
As(3,4)=Ts*km*cos(avp+av0)/DEN;
if wvp>=0
    As(4,4)=1-Ts*(Bmr+ktvp*wvp+TC^2/Rav)/Jmr;
    As(5,4)=Ts*lm*kfvp*wvp/Jv;
else
    As(4,4)=1-Ts*(Btr-ktvn*wvp+TC^2/Rav)/Jmr;
    As(5,4)=-Ts*lm*kfvn*wvp/Jv;
end
As(5,1)=-Ts*kvfv*kt/(Jv^2);
As(5,5)=1-Ts*kvfv/Jv;
if avp==0
    As(5,6)=Ts*g*((A-B)*cos(avp+av0)-C*sin(avp+av0))/Jv;
else
    As(5,6)=Ts*g*((A-B)*cos(avp+av0)-C*sin(avp+av0))/(Jv*avp);
end
As(6,1)=Ts*kt/Jv;
As(6,5)=Ts;
As(6,6)=1;

```

```

Bs(1,1)=Ts*TC*k2/(Jtr*Rah);
Bs(4,2)=Ts*TC*k1/(Jmr*Rav);
% =====Buoc du bao
Xp(3,1)=ahn;
Xp(6,1)=avn;
Xp(1,1)=whn;
Xp(2,1)=Shn;
Xp(4,1)=wvn;
Xp(5,1)=Svn;
Pk=As*Pk*As'+Q;
% ===== Buoc cap nhap
Lk=Pk*Cs'*inv(Cs*Pk*Cs'+R);
Xp=Xp+Lk*([ah;av]-Cs*Xp);
Pk=(eye(nx,nx)-Lk*Cs)*Pk;
Xe=Xp;
function OUTU =UKF(INU)
Uhp=INU(1);    %Dau vao doi tuong (bien duoc dieu khien)
Uvp=INU(2);    %Dau vao doi tuong (bien duoc dieu khien)
ah=INU(3);     %Dau ra doi tuong (bien duoc dieu khien)
av=INU(4);     %Dau ra doi tuong (bien duoc dieu khien)
ahp=INU(5);    %Dau ra doi tuong truoc do
avp=INU(6);    %Dau ra doi tuong truoc do
whp=INU(7);    %Trang thai duoc uoc luong truoc do
Shp=INU(8);    %Trang thai duoc uoc luong truoc do
wvp=INU(9);    %Trang thai duoc uoc luong truoc do
Svp=INU(10);   %Trang thai duoc uoc luong truoc do
nx=6;          %So bien trang thai
ny=2;          %So bien dau vao
ny=2;          %So bien dau ra

```

```

Pkpv=INU(13:12+nx*nx); %Dang vector cua P(k-1|k-1)
Pkp=zeros(nx,nx); %Dang matran cua P(k-1|k-1)
for i=1:nx
    Pkp(:,i)=Pkvp([1:nx]+(i-1)*nx,1);
end
% Cac tham so TRMS
lts=2.82e-1; % Chieu dai cua phan canh tay don duoi
lm=2.46e-1; % Chieu dai cua phan canh tay don chinh
Jv=5.26e-2; % Mo men quan tinh cua truc ngang
D=5.52877e-2;
E=5.85764e-3;
F=5.85077e-3;
g=9.81 % Gia toc trong trung
A=9.82347e-2;
B=1.135659e-1;
C=2.21788e-2;
H=4.94734e-2;
Jtr=3.1432e-5; % Mo men quan tinh cua dong co duoi/tai
Rah=8; % Dien tro phan ung
Lah=8.6e-4; % Dien cam phan ung
kthp=5e-8; % Th=kthp*sign(wh)*(wh)^2
kthn=4.41e-8; % Th=kthn*sign(wh)*(wh)^2
Btr=2.3e-5; % He so ma sat nhot cua rotor duoi
Jmr=2.0098e-4; % Mo men quan tinh cua dong co chinh/tai
Rav=8; % Dien tro phan ung
Lav=8.6e-4; % Dien cam phan ung
ktvp=5.6e-7; % Th=ktvp*sign(wv)*(wv)^2
ktvn=5.1e-7; % Th=ktvn*sign(wv)*(wv)^2
Bmr=2.97e-5; % He so ma sat nhot cua rotor chinh

```

```

av0=-6.048e-1;    % Goc chao doc ban dau
kfhp=2.1377e-6;  % Fh(wh)=kfhp*sign(wh)*(wh)^2
kfhn=1.9056e-6;  % Fh(wh)=kfhn*sign(wh)*(wh)^2
kfvp=1.9506e-5;  % Fv(wv)=kfvp*sign(wv)*(wv)^2
kfvn=1.1012e-5;  % Fv(wv)=kfvn*sign(wv)*(wv)^2
kvfh=4.91e-3;    % Ma sat nhot theo phuong ngang
kvfv=5.48e-3;    % Ma sat nhot theo phuong thang dung
kchp=5.60e-3;    % He so ma sat (cable force coefficient)
km=0.00023;      % Anh huong cua rotor chinh len goc theo phuong ngang
kt=0.000026;     % Anh huong cua rotor duoi len goc theo phuong thang dung
TC=0.0202;       % He so mo men xoan (mo men xoan khong doi)
k1=8.5;          % He so giao dien theo phuong thang dung (doc)
k2=6.5;          % He so giao dien ngang
Ts=0.2;          % Thoi gian lay mau
% Gia tri trung binh va hiep phuong sai
Ew=[1 0.1 0.032 1 0.1 0.032] % Vector trung binh cua nhieu qua trinh
Ev=ones(ny,1)*0.032;        % Vector trung binh cua nhieu do duoc
Q=eye(nx,nx)*0.4;           % Ma tran hiep phuong sai cua nhieu qua trinh
R=eye(ny,ny)*0.4;           % Ma tran hiep phuong sai cua nhieu do duoc
% Cac tham so UKF
a1=0.001;                   % Tham so alpha cua UKF
bet=2;                       % Tham so beta cua UKF
kai=0;                       % Tham so kai cua UKF
lam=a1^2*(2*nx+kai)-2*nx;    % Tham so lamda cua UKF
% Trang thai tang (trang thai duoc bo sung) va hiep phuong sai
Xkp=[whp;Shp;ahp;wvp;Svp;avp]; % x(k-1|k-1)
Xkpa=zeros(2*nx,1);          % xa(k-1|k-1)
Xkpa(1:nx,1)=Xkp;
Xkpa(nx+1:2*nx)=Ew;

```

```

Pkp_a=zeros(2*n_x,2*n_x);          % Pa(k-1|k-1)
Pkp_a(1+n_x:2*n_x,1+n_x:2*n_x)=Q;
% Dang 4*n_x+1 diem sigma
Ksip=zeros(2*n_x,4*n_x+1);        % Xi(k-1|k-1)
Ksip(:,1)=Xkp_a;
% Tim can bac hai cua ma tran Pkp_a bang viec cheo hoa gia tri rieng va vec to
rieng
VPkp=zeros(n_x,n_x);
DPkp=zeros(n_x,n_x);
[VPkp,DPkp]=myeig(Pkp);
DPkp_sqrt=sqrt(abs(DPkp));
YPkp=VPkp*DPkp_sqrt*inv(VPkp);
YPkp_a=[YPkp zeros(n_x,n_x); zeros(n_x,n_x) sqrt(Q)];
%=====
=====
Pkp_asqr=sqrt(2*n_x+lam)*YPkp_a;
for i=1:2*n_x
    Ksip(:,i+1)=Xkp_a+Pkp_asqr(:,i);
    Ksip(:,i+2*n_x+1)=Xkp_a-Pkp_asqr(:,i);
end
% Su lan truyen diem sigma thong qua cac phuong trinh khong gian trang thai
% roi rac phi tuyen cua TRMS
Ksi=zeros(n_x,4*n_x+1);
for i=4*n_x+1
    whp=Ksip(1,i);
    Shp=Ksip(2,i);
    ahp=Ksip(3,i);
    wvp=Ksip(4,i);
    Svp=Ksip(5,i);
    avp=Ksip(6,i);

```



```

if whp>=0
    whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp-kthp*whp^2)/Jtr;
    Shn=Shp+Ts*(lts*kfhp*whp^2*cos(avp+av0)-kvfh*Shp-kchp*...
        ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
        Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+...
        E*(sin(avp+av0))^2+F)^2;
else
    whn=whp+Ts*(TC*(k2*Uhp-TC*whp)/Rah-Btr*whp+kthn*whp^2)/Jtr;
    Shn=Shp+Ts*(-lts*kfhn*whp^2*cos(avp+av0)-kvfh*Shp-kchp*...
        ahp)/(D*(cos(avp+av0))^2+E*(sin(avp+av0))^2+F)-...
        Ts*kvfh*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+...
        E*(sin(avp+av0))^2+F)^2;
end
if wvp>=0
    wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-Bmr*wvp-ktvp*wvp^2)/Jmr;
    Svn=Svp+Ts*(lm*kfvp*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
        g*((A-B)*cos(avp+av0)-C*sin(avp+av0)))/Jv;
else
    wvn=wvp+Ts*(TC*(k1*Uvp-TC*wvp)/Rav-Bmr*wvp+ktvn*wvp^2)/Jmr;
    Svn=Svp+Ts*(-lm*kfvp*wvp^2-kvfv*Svp-kvfv*kt*whp/Jv+...
        g*((A-B)*cos(avp+av0)-C*sin(avp+av0)))/Jv;
end
ahn=ahp+Ts*Shp+Ts*km*wvp*cos(avp+av0)/(D*(cos(avp+av0))^2+...
    E*(sin(avp+av0))^2+F);
avn=avp+Ts*Svp+Ts*kt*whp/Jv;
Ksi(:,i)=[whn;Shn;ahn;wvn;Svn;avn]+Ksip(nx+1:2*nx,i); % Xi(klk-1)
end
% Trong so cac diem sigma duoc su dung cau truc trang thai du bao va hiep
% phuong sai

```

```

Ws0=lam/(2*nx+lam);           % Trong so cua trang thai du bao
Wc0=lam/(2*nx+lam)+(1-a1^2+bet); % Trong so cua hiep phuong sai duoc DB
Wi=1/(2*(2*nx+lam));         % Trong so cua ca hai
Xk=Ws0*Ksi(:,1)+Wi*sum(Ksi(:,2:4*nx+1),2); % x(klk-1)
Pk=zeros(nx,nx);
for i=1:4*nx
    Pk=Pk+Wi*((Ksi(:,i+1)-Xk)*(Ksi(:,i+1)-Xk)');
end
Pk=Pk+Wc0*(Ksi(:,1)-Xk)*(Ksi(:,1)-Xk)';
% Giai doan Cap nhat cua EKF
lam=a1^2*(nx+ny+kai)-nx-ny;   % Tham so lamda cua UKF
% Trang thai tang (trang thai duoc bo sung) va hiep phuong sai
Xka=zeros(nx+ny,1);          % xa(klk-1)
Xka(1:nx,1)=Xk;
Xka(nx+1:nx+ny)=Ev;
Pka=zeros(nx+ny,nx+ny);     % Pa(klk-1)
Pka(1:nx,1:nx)=Pk;
Pka(1+nx:nx+ny,1+nx:nx+ny)=R;
% Dang 4*nx+1 diem sigma
Ksiy=zeros(nx+ny,2*(nx+ny)+1); % Xi(klk-1)
Ksiy(:,1)=Xka;
% Tim can bac hai cua ma tran Pkpa bang viec cheo hoa gia tri rieng va vec to
rieng
VPk=zeros(nx,nx);
DPk=zeros(nx,nx);
[VPk,DPk]=myeig(Pk);
DPksqrt=sqrt(abs(DPk));
YPk=VPk*DPksqrt*inv(VPk);
YPka=[YPk zeros(nx,ny);zeros(ny,nx) sqrt(R)];

```

```

%=====
=====

Pkasqr=sqrt(nx+ny+lam)*YPka;
for i=1:(nx+ny)
    Ksiy(:,i+1)=Xka+Pkasqr(:,i);
    Ksiy(:,i+nx+ny+1)=Xka-Pkasqr(:,i);
end

Cs=[0 0 1 0 0 0;0 0 0 0 0 1];
gam=zeros(ny,2*(nx+ny)+1);           % Gama(k)
for i=1:2*(nx+ny)+1
    gam(:,i)=Cs*Ksiy(1:nx,i)+Ksiy(nx+1:nx+ny,i);
end

% Trong so cac diem sigma duoc su dung cau truc trang thai du bao va hiep
% phuong sai
Ws0=lam/(nx+ny+lam);                   % Trong so cua trang thai du bao
Wc0=lam/(nx+ny+lam)+(1-a1^2+bet);      % Trong so cua hiep phuong sai duoc
du bao
Wi=1/(2*(nx+ny+lam));                  % Trong so cua ca hai
Yh=Ws0*gam(:,1)+Wi*sum(gam(:,2:2*(nx+ny)+1),2); % yhat(k)
Pzz=zeros(ny,ny);
Pxz=zeros(nx,ny);
for i=1:2*(nx+ny)
    Pzz=Pzz+Wi*(gam(:,i+1)-yh)*(gam(:,i+1)-yh)';
    Pxz=Pxz+Wi*(Ksiy(1:nx,i+1)-Xk)*(gam(:,i+1)-yh)';
end

Pzz=Pzz+Wc0*(gam(:,1)-yh)*(gam(:,1)-yh)';
Pxz=Pxz+Wc0*(Ksiy(1:nx,1)-Xk)*(gam(:,1)-yh)';
Lk=Pxz*inv(Pzz);
Xkn=Xk+Lk*([ah;av]-yh);
Pkn=Pk-Lk*Pzz*Lk';

```

*% Thay doi ma tran duong cheo Pkn ben trong dang vector (Pknv)*

```
Pknv=zeros(nx*nx,1);
```

```
for i=1:nx
```

```
    Pknv([1:nx]+(i-1)*nx,1)=Pkn(:,i);
```

```
end
```

```
OUTU=[Xkn;Pknv];
```

```
function [V,D]=myeig(A)
```

```
L=length(A);
```

```
Q=zeros(L,L);
```

```
V=eye(V,V);
```

```
for i=1:L^3
```

```
    [Q,R]=QRdec(A);
```

```
    A=R*Q;
```

```
    V=V*Q;
```

```
end
```

```
D=zeros(L,L);
```

```
for i=1:L
```

```
    D(i,i)=A(i,i);
```

```
end
```

```
function [Q,R]=QRdec(A)
```

```
L=length(A);
```

```
U=zeros(L,L);
```

```
Q=zeros(L,L);
```

```
if mynorm(A(:,1))~=0
```

```
    Q(:,1)=A(:,1)/mynorm(A(:,1));
```

```
end
```

```
for i=1:L
```

```
    for j=1:i-1
```

```

    if mynorm(Q(:,j))~=0
        U(:,i)=U(:,i)-Q(:,j)'*A(:,i)/(Q(:,j)'*Q(:,j))*Q(:,j);
    end
end
U(:,i)=U(:,i)+A(:,i);
if mynorm(U(:,i))~=0
    Q(:,i)=U(:,i)/mynorm(U(:,i));
end
end
R=Q'*A;
function [Y]=mynorm(X)
Y=sqrt(sum(X.^2));

#define S_FUNCTION_NAME TRMS_Estimation_sf
#define S_FUNCTION_LEVEL 2
#include "simstruc.h"
#include "math.h"
#include "codegen\lib\mdlKalman\mdlKalman.h"
static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumSFcnParams(S, 0);
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
        return; /* Parameter mismatch will be reported by Simulink */
    }
    ssSetNumContStates(S, 0);
    ssSetNumDiscStates(S, 6);
    if (!ssSetNumInputPorts(S, 4)) return;
    ssSetInputPortWidth(S, 0, 1);
    ssSetInputPortWidth(S, 1, 1);
    ssSetInputPortWidth(S, 2, 1);
    ssSetInputPortWidth(S, 3, 1);
}

```

```

ssSetInputPortSampleTime(S, 0,CONTINUOUS_SAMPLE_TIME);
ssSetInputPortSampleTime(S, 1,CONTINUOUS_SAMPLE_TIME);
ssSetInputPortSampleTime(S, 2,CONTINUOUS_SAMPLE_TIME);
ssSetInputPortSampleTime(S, 3,CONTINUOUS_SAMPLE_TIME);
ssSetInputPortOffsetTime(S, 0, 0.0);
ssSetInputPortOffsetTime(S, 1, 0.0);
ssSetInputPortOffsetTime(S, 2, 0.0);
ssSetInputPortOffsetTime(S, 3, 0.0);
ssSetInputPortDirectFeedThrough(S, 0,0);
ssSetInputPortDirectFeedThrough(S, 1,0);
ssSetInputPortDirectFeedThrough(S, 2,0);
ssSetInputPortDirectFeedThrough(S, 3,0);
if (!ssSetNumOutputPorts(S,1)) return;
ssSetOutputPortWidth(S, 0, 6);
ssSetOutputPortSampleTime(S, 0,CONTINUOUS_SAMPLE_TIME);
ssSetOutputPortOffsetTime(S, 0,0.0);
ssSetNumSampleTimes(S, 2);
ssSetSimStateCompliance(S, USE_DEFAULT_SIM_STATE);
ssSetOptions(S,
SS_OPTION_EXCEPTION_FREE_CODE|SS_OPTION_PORT_SAMPLE_
TIMES_ASSIGNED);
static void mdlInitializeSampleTimes(SimStruct *S)
{ ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
  ssSetOffsetTime(S, 0, 0.0);
  ssSetSampleTime(S, 1,0.2);
  ssSetOffsetTime(S, 1, 0.0);
  ssSetModelReferenceSampleTimeDefaultInheritance(S); }
#define MDL_INITIALIZE_CONDITIONS
static void mdlInitializeConditions(SimStruct *S)
{ int i;

```

```

real_T *xdisc = ssGetRealDiscStates(S);
for (i = 0; i < 6; i++) { *xdisc++ = 0.0; } }

#define MDL_UPDATE
void mdlUpdate(SimStruct *S, int_T tid)
{ real_T *Xe = ssGetRealDiscStates(S);
  InputRealPtrsType uPtrs1 = ssGetInputPortRealSignalPtrs(S,0);
  InputRealPtrsType uPtrs2 = ssGetInputPortRealSignalPtrs(S,1);
  InputRealPtrsType uPtrs3 = ssGetInputPortRealSignalPtrs(S,2);
  InputRealPtrsType uPtrs4 = ssGetInputPortRealSignalPtrs(S,3);
if (ssIsSampleHit(S, 1, tid))
mdlKalman((*uPtrs1[0]),(*uPtrs2[0]),(*uPtrs3[0]),(*uPtrs4[0]),Xe);}
static void mdlOutputs(SimStruct *S, int_T tid)
{ int i;
  real_T *Out = ssGetOutputPortRealSignal(S,0);
  real_T *Xe = ssGetRealDiscStates(S);
  for ( i=0;i<6;i++)
  {
    *Out++= *Xe++; }
static void mdlTerminate(SimStruct *S)

```