

Lời cam đoan

Tôi xin cam đoan đây là công trình nghiên cứu của cá nhân tôi dưới sự hướng dẫn của tập thể các nhà khoa học và các tài liệu tham khảo đã trích dẫn. Kết quả nghiên cứu là trung thực và chưa được công bố trên bất cứ một công trình nào khác.

Tác giả

Hoàng Đức Quỳnh

Lời cảm ơn

Trong quá trình làm luận án với đề tài **Ứng dụng lọc Kalman mở rộng (EKF) trong điều khiển dự báo cho một lớp đối tượng phi tuyến**, tôi đã nhận được rất nhiều góp ý về chuyên môn cũng như sự ủng hộ của các 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 Thái nguyên, 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 Thái nguyên đã tạo điều kiện thuận lợi cho tôi trong suốt quá trình thực hiện đề tài luận án.

Tác giả luận án

Hoàng Đức Quỳnh

MỤC LỤC

Lời cam đoan.....	i
Lời cảm ơn	ii
Các ký hiệu được sử dụng	vi
Các ký hiệu viết tắt.....	viii
Danh mục các hình vẽ	ix
Danh mục các thuật toán.....	xi
PHẦN MỞ ĐẦU	1
Tính cấp thiết của đề tài luận án	1
Mục tiêu và nhiệm vụ của luận án	2
Phạm vi, đối tượng và phương pháp nghiên cứu của luận án	2
Ý nghĩa khoa học và thực tiễn của luận án	3
Ý nghĩa khoa học	3
Ý nghĩa thực tiễn	4
Bố cục của luận án	4
Những đóng góp của luận án.....	5
CHƯƠNG 1 TỔNG QUAN VỀ PHƯƠNG PHÁP ĐIỀU KHIỂN DỰ	7
BÁO PHẢN HỒI ĐẦU RA	7
1.1 Điều khiển dự báo phản hồi đầu ra hệ có mô hình tuyến tính	11
1.1.1 Phương pháp MAC (Model algorithmic control)	12
1.1.2 Phương pháp DMC (Dynamic matrix control)	13
1.1.3 Phương pháp GPC (Generalized predictive control)	15
1.1.4 Điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ có mô hình tuyến tính.....	17
Điều khiển dự báo phản hồi trạng thái	17
Quan sát trạng thái hệ tuyến tính với lọc Kalman	19
Sử dụng KF vào điều khiển dự báo phản hồi đầu ra cho hệ tuyến tính	22
1.2 Phương pháp điều khiển dự báo phản hồi đầu ra cho hệ có mô hình phi tuyến	22
1.2.1 Điều khiển dự báo phản hồi trạng thái	23
1.2.2 Lọc Kalman mở rộng (EKF-extended Kalman filter)	25

1.2.3 UKF - Unscented Kalman Filter	31
1.2.4 Điều khiển dự báo phản hồi đầu ra theo nguyên lý tách với bộ lọc Kalman phi tuyến	33
1.3 Một số công trình của các tác giả trong và ngoài nước thời gian gần đây nghiên cứu về điều khiển dự báo phản hồi đầu ra	34
1.4 Định hướng nghiên cứu của luận án	37
1.5 Kết luận chương 1	38
CHƯƠNG 2 THIẾT KẾ LỌC KALMAN ĐỂ QUAN SÁT TỪNG ĐOẠN TRẠNG THÁI THEO NGUYÊN LÝ TỐI ƯU VÀ ỨNG DỤNG VÀO ĐIỀU KHIỂN DỰ BÁO PHẢN HỒI ĐẦU RA HỆ PHI TUYẾN THEO NGUYÊN LÝ TÁCH	40
2.1 Xây dựng bộ quan sát Kalman từng đoạn cho hệ phi tuyến	40
2.1.1 Quan sát Kalman từng đoạn cho hệ song tuyến.....	40
Xuất phát điểm của phương pháp	41
Xây dựng bộ quan sát Kalman từng đoạn cho hệ song tuyến	42
2.1.2 Thiết kế bộ quan sát Kalman từng đoạn cho hệ phi tuyến	50
Quan sát trạng thái hệ phi tuyến khi đã biết trạng thái đầu	50
Xác định xấp xỉ trạng thái đầu theo tiêu chuẩn tối ưu	53
Thuật toán quan sát từng đoạn trạng thái cho hệ phi tuyến	55
2.2 Điều khiển dự báo phản hồi trạng thái hệ phi tuyến trên cơ sở sử dụng mô hình dự báo tuyến tính.....	56
2.2.1 Điều khiển hệ song tuyến.....	57
Điều khiển hệ hợp thức không chặt	57
Điều khiển hệ hợp thức chặt	61
2.2.2 Điều khiển hệ phi tuyến.....	63
Điều khiển hệ hợp thức không chặt	63
Điều khiển hệ hợp thức chặt	66
2.3 Điều khiển dự báo phản hồi đầu ra hệ phi tuyến với các bộ lọc Kalman mở rộng	67
2.3.1 Thuật toán điều khiển	67
2.3.2 Tính ổn định ISS của bộ điều khiển phản hồi đầu ra	70
2.4 Kết luận chương 2.....	73
CHƯƠNG 3 THỰC NGHIỆM KIỂM CHỨNG CHẤT LƯỢNG CỦA BỘ ĐIỀU KHIỂN ĐÃ ĐỀ XUẤT	75
3.1 Điều khiển đối tượng con lắc ngược.....	75

3.1.1	Điều khiển dự báo phản hồi trạng thái	77
	Chỉ điều khiển bám ổn định góc lắc	77
	Điều khiển bám ổn định góc lắc và vị trí	88
3.1.2	Quan sát trạng thái với lọc Kalman mở rộng.....	89
3.1.3	Điều khiển phản hồi đầu ra theo nguyên lý tách.....	92
3.2	Kiểm chứng chất lượng bộ điều khiển dự báo phản hồi đầu ra trên mô hình thí nghiệm con lắc ngược quay.....	95
3.2.1	Mô hình toán của đối tượng con lắc ngược quay.....	95
3.2.2	Kết quả mô phỏng bộ điều khiển phản hồi đầu ra cho đối tượng con lắc ngược quay	105
3.2.3	Mô tả hệ thống và kết quả thí nghiệm	107
	3.2.3.1 Mô tả hệ thống thí nghiệm	107
	3.2.3.2 Kết quả thí nghiệm	113
3.3	Kết luận chương 3.....	120
	KẾT LUẬN VÀ KIẾN NGHỊ	121
	Những vấn đề đã làm được.....	121
	Các vấn đề còn tồn tại và hướng nghiên cứu tiếp theo.....	123
	Các công trình khoa học đã công bố	124
	Tài liệu tham khảo	125
	PHỤ LỤC	132
P1	Điều khiển dự báo phản hồi trạng thái hệ con lắc ngược	132
P2	Quan sát trạng thái hệ con lắc ngược	143
P3	Điều khiển dự báo phản hồi đầu ra hệ con lắc ngược	147
P4	Điều khiển dự báo phản hồi đầu ra cho hệ con lắc ngược quay ...	152
	P4a Quan sát trạng thái EKF loại 3 hệ con lắc ngược quay	152
	P4b Điều khiển dự báo phản hồi trạng thái hệ con lắc ngược quay	156
	P4c Điều khiển dự báo phản hồi đầu ra hệ con lắc ngược quay	161
P5	Mã nguồn chương trình của khối NonlinPreControl.....	166
P6	Mã nguồn chương trình của khối Nonlinear Estimate và EsUpdate	171

Các ký hiệu được sử dụng

$\underline{x}(kT_a) \in \mathbb{R}^n$	Vector của n giá trị biến trạng thái trong hệ tại thời điểm $t = kT_a$ với T_a là chu kỳ trích mẫu.
$\underline{u}_k = \underline{u}(kT_a) \in \mathbb{R}^m$	là vector của m giá trị các tín hiệu vào (tín hiệu điều khiển)
$\underline{y}_k = \underline{y}(kT_a) \in \mathbb{R}^r$	là vector của r giá trị các tín hiệu ra.
$\mathcal{Q}_k, \mathcal{R}_k$	là hai ma trận tham số được thay đổi một cách thích hợp cùng với cửa sổ dự báo
N	Cửa sổ dự báo
$\underline{\triangle}$	Ký hiệu định nghĩa hàm $\underline{d}_j(\underline{\mathbf{u}}) : \triangleq \underline{d}_j(\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N-1}) = \underline{d}_j(\underline{\mathbf{u}})$
$\underline{\mathbf{u}}^* = \arg \min_{\underline{u}_k \in \mathcal{U}} J(\underline{\mathbf{u}})$	Trả về giá trị \underline{u}_k thỏa mãn điều kiện ràng buộc $\underline{u}_k \in \mathcal{U}$ làm cho hàm $J(\underline{\mathbf{u}})$ đạt giá trị nhỏ nhất
$\underline{y}_k = \underline{y}(kT_a) \in \mathbb{R}^r$	là vector của r giá trị các tín hiệu ra.
$\text{diag}(M)$	Ma trận đường chéo có các phần tử trên đường chéo chính là M

M^T	Chuyển vị của ma trận M
Θ	Ký hiệu ma trận có tất cả các phần tử bằng 0
$\frac{\partial f}{\partial x}$	Đạo hàm Jacobi của hàm f theo x
$M\{\cdot\}$	Phép tính lấy kỳ vọng
$\underline{x}_k(-), \underline{x}_k(+)$	Giá trị ước lượng trung gian của giá trị thực \underline{x}_k
$\underline{e}_k(+), \underline{e}_k(-)$	Hai giá trị sai lệch của ước lượng trung gian: $\underline{e}_k(+) = \underline{x}_k(+) - \underline{x}_k; \underline{e}_k(-) = \underline{x}_k(-) - \underline{x}_k$
$P_k(+), P_k(-)$	Giá trị hiệp phương sai tương quan hai giá trị ước lượng $P_k(+) = M\{\underline{e}_k(+) \underline{e}_k^T(+)\}; P_k(-) = M\{\underline{e}_k(-) \underline{e}_k^T(-)\}$
$col(\underline{w}_k, \underline{w}_{k+1}, \dots, \underline{w}_{k+N-1})$	Vector cột có các phần tử là $\underline{w}_k, \underline{w}_{k+1}, \dots, \underline{w}_{k+N-1}$
I	Ký hiệu ma trận đơn vị
$(\underline{x}_k^{nor}, \underline{u}_k^{nor})$	Quỹ đạo tiền định (norminal trajectory), là quỹ đạo thỏa mãn $\underline{x}_{k+1}^{nor} = f_{-k}(\underline{x}_k^{nor}, \underline{u}_k^{nor})$
$\delta(h)$	Ký hiệu xung dirac

Các ký hiệu viết tắt

DMC	Dynamic Matrix Control
EKF	Extended Kalman Filter
GMV	Generalized Minimum Variance
GPC	Generalized Predictive Control
KF	Kalman Filter
LTI	Linear Time-Invariant
MAC	Model Algorithmic Control
MIMO	Multiple Input, Multiple Output
MPC	Model Predictive Control
MV	Minimum Variance
PID	Proportional–Integral–Derivative
QP	Quadratic Programming
SQP	Sequential Quadratic Programming
SISO	Single Input, Single Output
SIMO	Single Input, Multiple Output
ISS	Input – to – State Stability

Danh mục các hình vẽ

Hình 1.1: Cấu trúc hệ điều khiển dự báo	7
Hình 1.2: Điều khiển phản hồi đầu ra theo nguyên lý tách	17
Hình 1.3: Xác định xấp xỉ trạng thái của hệ nhờ lọc Kalman.....	19
Hình 2.1: Trạng thái quan sát được và trạng thái thực của đối tượng khi có nhiều đầu vào và nhiều đầu ra có giá trị kỳ vọng $\mu = 0$	49
Hình 2.2: Biến trạng thái $x_1[k]$ khi có nhiều đầu vào và nhiều đầu ra có giá trị kỳ vọng $\mu = 0$	50
Hình 2.3: Biến trạng thái $x_2[k]$ khi có nhiều đầu vào và nhiều đầu ra có giá trị kỳ vọng $\mu = 0$	50
Hình 2.4: Nguyên lý điều khiển dự báo phản hồi trạng thái phi tuyến trên cơ sở sử dụng mô hình dự báo tuyến tính	56
Hình 2.5: Cấu trúc hệ điều khiển phản hồi đầu ra theo nguyên lý tách	68
Hình 2.6: Tính ổn định ISS của hệ kín phản hồi đầu ra theo nguyên lý tách.....	72
Hình 3.1: Cấu trúc vật lý hệ con lắc ngược.	76
Hình 3.2: Lưu đồ thuật toán thiết kế bộ điều khiển DBPHTT bám theo giá trị đầu ra đặt cho con lắc ngược theo Thuật toán 2.6 với $t \leq t_{lv}$ là thời gian mô phỏng.	81
Hình 3.3: So sánh góc lắc thực có với góc lắc đặt trước khi sử dụng trực tiếp mô hình phi tuyến để thiết kế bộ điều khiển dự báo phản hồi trạng thái theo Thuật toán 2.6	82
Hình 3.4: Lưu đồ thuật toán thiết kế bộ điều khiển DBPHTT bám theo giá trị đầu ra đặt cho con lắc ngược theo Thuật toán 2.4 với $t \leq t_{lv}$ là thời gian mô phỏng.....	86
Hình 3.5: So sánh góc lắc thực có với góc lắc đặt trước khi sử dụng mô hình song tuyến (3.10).	86
Hình 3.6: So sánh góc lắc thực có với góc lắc đặt trước khi sử dụng mô hình song tuyến (3.16).	87
Hình 3.7: So sánh kết quả điều khiển vị trí góc theo giá trị đặt khi sử dụng 2 bộ điều khiển theo Thuật toán 2.4 và Thuật toán 2.6.....	87
Hình 3.8: So sánh vị trí thực có với vị trí đặt trước	89
Hình 3.9: So sánh góc lắc thực có với góc lắc đặt trước	89
Hình 3.10: Giá trị trạng thái x_3, x_4 quan sát được so sánh với giá trị thực khi có nhiều hệ thống và nhiều đầu ra là ồn trắng.....	91
Hình 3.11: Giá trị trạng thái x_1, x_2 quan sát được so sánh với giá trị thực khi có nhiều hệ thống và nhiều đầu ra là ồn trắng.....	91
Hình 3.12: Lưu đồ thuật toán thiết kế bộ ĐKDB PHDR bám theo giá trị đầu ra đặt cho đối tượng con lắc ngược theo Thuật toán 2.7 với $t \leq t_{lv}$ là thời gian mô phỏng	92

Hình 3.13: Góc lắc thực $y_2 = \varphi$ so sánh với góc lắc đặt thu được nhờ bộ điều khiển dự báo phản hồi đầu ra khi có nhiễu hệ thống và nhiễu đầu ra ở dạng ồn trắng.....	93
Hình 3.14: Vị trí xe thực có $y_1 = y_c$ so sánh với vị trí thu được nhờ bộ điều khiển dự báo phản hồi đầu ra khi có nhiễu hệ thống và nhiễu đầu ra ở dạng ồn trắng.....	94
Hình 3.15: Mô hình động học con lắc ngược quay.....	95
Hình 3.16: Sơ đồ tương đương động cơ điện 1 chiều và bộ điều chế độ rộng xung (bộ khuếch đại xung PWM).....	96
Hình 3.17: Hình chiếu bằng của hệ con lắc ngược quay.....	98
Hình 3.18: Lưu đồ thuật toán thiết kế bộ ĐKDBPHĐR bám theo giá trị đầu ra đặt cho con lắc ngược quay theo Thuật toán 2.7	104
Hình 3.19: Kết quả quan sát trạng thái α và $\dot{\alpha}$ bằng EKF loại 3	106
Hình 3.20: Kết quả quan sát trạng thái β và $\dot{\beta}$ bằng EKF loại 3	106
Hình 3.21: Kết quả mô phỏng bộ điều khiển phản hồi đầu ra so sánh với bộ điều khiển MPC phản hồi trạng thái cho đối tượng con lắc ngược quay (đầu ra là vị trí góc con lắc theo trục z).....	107
Hình 3.22: Mô hình con lắc ngược quay Kri-300.....	108
Hình 3.23: Mô hình thí nghiệm con lắc ngược quay Kri PP-300 tại Phòng thí nghiệm Đo lường – Điều khiển, Trường ĐH KTCN Thái Nguyên.....	108
Hình 3.24: Vi mạch Arduino Mega 2560	109
Hình 3.25: Bo mạch công suất điều khiển động cơ	109
Hình 3.26: Bộ nguồn RPS 305DU	109
Hình 3.27: Bộ nguồn RPS 305DU	110
Hình 3.28: Vị trí Encoder	110
Hình 3.29: Đầu vào, ra số trên Arduino Mega 2560.....	111
Hình 3.30: Đầu ra PWM trên Arduino Mega 2560.....	112
Hình 3.31: Cổng Communication trên Arduino Mega 2560.....	112
Hình 3.32: Đầu vào tương tự trên Arduino Mega 2560	113
Hình 3.33: Sơ đồ kết nối thiết bị thí nghiệm.....	113
Hình 3.34: Hình ảnh bàn thí nghiệm con lắc ngược quay tại trường ĐH KTCN – Đại học Thái Nguyên.....	114
Hình 3.35: Giao diện thí nghiệm cho hệ con lắc ngược quay, thực hiện trên Matlab/Simulink Real Time.....	115
Hình 3.36: Góc và vận tốc góc của con lắc khi sử dụng bộ điều khiển DBPHTT theo Thuật toán 2.4.....	116
Hình 3.37: Tín hiệu điều khiển từ đầu ra của bộ điều khiển dự báo theo Thuật toán 2.4 (đưa tới đầu vào của bộ khuếch đại xung PWM).....	116
Hình 3.38: Góc và vận tốc góc của con lắc khi điều khiển bằng bộ điều khiển dự báo phản hồi đầu ra theo Thuật toán 2.7	117
Hình 3.39: Tín hiệu điều khiển đưa ra bởi bộ điều khiển dự báo phản hồi đầu ra theo Thuật toán 2.7 (đưa tới đầu vào bộ khuếch đại xung PWM).....	117
Hình 3.40: Hình ảnh tại bàn thí nghiệm khi điều khiển con lắc ở vị trí thẳng đứng hướng lên trên (góc con lắc bám theo giá trị 0) bằng bộ điều khiển dự báo phản hồi đầu ra theo Thuật toán 2.7	118
Hình 3.41: So sánh góc con lắc được điều khiển trong hai trường hợp phản hồi đầu ra và phản hồi trạng thái	119

Hình 3.42: So sánh tín hiệu điều khiển trong hai trường hợp phản hồi đầu ra và phản hồi trạng thái..... 119

Danh mục các thuật toán

Thuật toán 1.1(MAC [1,25,45])	13
Thuật toán 1.2 (DMC [1,25]).....	14
Thuật toán 1.3 (GPC [1,25,45])	16
Thuật toán 1.4 (điều khiển dự báo phản hồi trạng thái hệ tuyến tính [12,31]).....	18
Thuật toán 1.5 (KF)	22
Thuật toán 1.6 (điều khiển dự báo phản hồi trạng thái hệ phi tuyến).....	24
Thuật toán 1.7 (EKF loại 1, [12,35,61]).....	27
Thuật toán 1.8 (EKF loại 2, [12,35,61]).....	28
Thuật toán 1.9 (EKF loại 3, [12,35,61]).....	30
Thuật toán 1.10 (UKF- theo tài liệu [45])	31
Thuật toán 2.1: Quan sát Kalman từng đoạn hệ song tuyến (2.5)	46
Thuật toán 2.2: Quan sát trạng thái hệ phi tuyến.	55
Thuật toán 2.3: Điều khiển bám tín hiệu đầu ra mẫu cho hệ song tuyến (2.28) bằng bộ điều khiển dự báo phản hồi trạng thái.	60
Thuật toán 2.4: Điều khiển bám tín hiệu đầu ra mẫu cho hệ song tuyến (2.36) bằng bộ điều khiển dự báo phản hồi trạng thái.	62
Thuật toán 2.5: Điều khiển bám tín hiệu đầu ra mẫu cho hệ phi tuyến (2.39) bằng bộ điều khiển dự báo phản hồi trạng thái.	65
Thuật toán 2.6: Điều khiển bám tín hiệu đầu ra mẫu cho hệ phi tuyến (2.49) bằng bộ điều khiển dự báo phản hồi trạng thái.	66
Thuật toán 2.7: Điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ phi tuyến (2.52) hoặc (2.53) với bộ lọc Kalman EKF loại 3.....	69

PHẦN MỞ ĐẦU

Tính cấp thiết của đề tài luận án

Điều khiển dự báo (MPC-Model Predictive Control), còn được biết dưới tên gọi là điều khiển trượt dọc trên trục thời gian (RHC-Receding Horizon Control), là một kỹ thuật điều khiển dựa trên nền tối ưu hóa mang tính ứng dụng cao trong thực tế, nhất là đối với các quá trình nhiều biến phức tạp, có thêm điều kiện ràng buộc cho bài toán điều khiển [18,51]. Điều này đã được chứng minh qua hơn 3000 ứng dụng thành công của kỹ thuật này trong điều khiển quá trình, công nghiệp hóa chất, dầu khí, chế biến... [12,25,31,40,48,28]. Tuy nhiên, nếu so sánh với số lượng các ứng dụng thành công cho các quá trình mang tính tuyến tính thuần túy thì khi áp dụng vào điều khiển các đối tượng công nghiệp mang tính phi tuyến, bị tác động bởi nhiễu, cả ở bên trong hệ thống và đầu ra của hệ, tỷ lệ số lượng ứng dụng của kỹ thuật này còn khá khiêm tốn [25,31,28]. Điều này có nhiều nguyên nhân của nó mà chủ yếu có thể kể đến như:

Thứ nhất: các biến trạng thái của quá trình phi tuyến bị nhiễu tác động phần lớn, thậm chí là không thể đo được một cách đủ chính xác, để đảm bảo có được một chất lượng điều khiển tốt [11,19,39].

Thứ hai: với các quá trình phi tuyến, khi sử dụng trực tiếp mô hình phi tuyến cho công việc dự báo tín hiệu đầu ra, công thức dự báo rất phức tạp với độ phức tạp tăng theo tỷ lệ cấp lũy thừa với độ rộng cửa sổ dự báo, trong khi cửa sổ dự báo càng nhỏ, chất lượng điều khiển càng kém. Việc sử dụng trực tiếp mô hình phi tuyến cho công việc dự báo trong nhiều trường hợp là không khả thi, lý do cho tính không khả thi này nằm ở việc xác định các vector hàm $f_i^g(\cdot)$, $i = 1, 2, \dots, N$ và việc tìm nghiệm tối ưu \underline{u}^* của hàm mục tiêu lúc này có dạng phi tuyến rất cao [31,40].

Thứ ba: với cửa sổ dự báo hữu hạn, kỹ thuật điều khiển dự báo luôn phải đòi hỏi có thêm hàm chặn trong hàm mục tiêu, vì chỉ có như vậy, chất lượng ổn định mới được đảm bảo. Song với quá trình phi tuyến thì câu hỏi cần phải lựa chọn hàm chặn như thế nào mới hợp lý, cho tới nay vẫn còn bỏ ngỏ [12,40].

Chính những nguyên nhân cũng như khó khăn cơ bản nêu trên đã cho thấy được tính cấp thiết của đề tài luận án liên quan tới việc nghiên cứu phát triển bộ điều khiển dự báo phản hồi đầu ra mang tính khả dụng cao với những đối tượng phi tuyến.

Mục tiêu và nhiệm vụ của luận án

Mục tiêu tổng quát: nghiên cứu thiết kế được bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách mang tính khả dụng cao cho những đối tượng phi tuyến. Bộ điều khiển dự báo này sử dụng mô hình tuyến tính hóa tại từng thời điểm trích mẫu để làm mô hình dự báo, sử dụng lọc Kalman mở rộng để quan sát trạng thái của đối tượng phi tuyến.

Để vượt qua được khó khăn thứ nhất đã nêu trong phần tính cấp thiết, luận án đặt ra nhiệm vụ sẽ sử dụng lọc Kalman để quan sát trạng thái của quá trình, thay vì dùng cảm biến đo mà thường đối với nhiều biến trạng thái là không thể. Đối với các quá trình phi tuyến thì đó sẽ là lọc Kalman mở rộng, viết tắt là EKF (Extended Kalman Filter).

Với khó khăn thứ hai, luận án đề xuất sử dụng mô hình dự báo tuyến tính thay cho việc sử dụng trực tiếp mô hình phi tuyến của quá trình vào dự báo tín hiệu ra. Cùng với việc sử dụng mô hình dự báo tuyến tính này, khó khăn thứ ba cũng sẽ được giải quyết, vì khi đó hàm mục tiêu trở nên thuần túy là một hàm toàn phương theo tín hiệu điều khiển, do đó dạng hàm phạt thích hợp tương ứng, nếu cần phải bổ sung, thì theo lý thuyết hàm Bellman, cũng sẽ chỉ là một hàm toàn phương [12].

Phạm vi, đối tượng và phương pháp nghiên cứu của luận án

Để thực hiện nhiệm vụ đề tài yêu cầu cho các đối tượng phi tuyến, luận án đặt ra mục tiêu nghiên cứu trước mắt là phát triển lọc Kalman và điều khiển dự báo cho đối tượng song tuyến (bilinear), rồi từ đó mới mở rộng cho đối tượng phi tuyến tổng quát. Bên cạnh đó luận án cũng sẽ nghiên cứu chất lượng của bộ điều khiển dự báo phi tuyến phản hồi đầu ra trên cơ sở ghép chung bộ quan sát trạng thái, mà ở đây là

bộ lọc Kalman, cùng với bộ điều khiển dự báo phản hồi trạng thái với mô hình dự báo tuyến tính rời rạc hóa từng đoạn. Bộ điều khiển đó sẽ được luận án gọi là bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách.

Để thực hiện được nhiệm vụ nghiên cứu và đạt được mục tiêu nghiên cứu của đề tài, Luận án sử dụng các phương pháp nghiên cứu:

Nghiên cứu lý thuyết: Phân tích, tổng hợp các kiến thức về điều khiển dự báo phản hồi đầu ra, về các bộ quan sát trạng thái Kalman. Thiết kế thêm bộ quan sát Kalman theo nguyên lý tối ưu. Thiết kế các thuật toán điều khiển dự báo phản hồi trạng thái, điều khiển dự báo phản hồi đầu ra cho đối tượng song tuyến, phi tuyến có mô hình dạng hợp thức chặt và hợp thức không chặt.

Nghiên cứu mô phỏng: Sử dụng công cụ Matlab – Simulink để mô phỏng kiểm chứng các nhận định lý thuyết và các thuật toán đã đề xuất.

Nghiên cứu thực nghiệm: Sử dụng mô hình thí nghiệm để kiểm chứng tính đúng đắn các lý thuyết đã nghiên cứu được.

Ý nghĩa khoa học và thực tiễn của luận án

Ý nghĩa khoa học

Ý tưởng tuyến tính rời rạc hóa từng đoạn mô hình phi tuyến phục vụ cho công việc điều khiển dự báo là không mới, nó đã được đề xuất bởi tài liệu [12,1,57], song điểm khác biệt ở đây là Luận án sẽ sử dụng cửa sổ dự báo hữu hạn thay vì vô hạn như đã làm ở [12,1,57]. Điều đó sẽ tạo ra thêm khả năng cho bộ điều khiển thu được các tính chất sau:

Có thể dễ dàng xử lý được các điều kiện ràng buộc nhờ các thuật toán tối ưu hóa [53].

Điều khiển được tín hiệu ra bám trực tiếp theo tín hiệu đặt cho trước mà không cần phải chuyển qua bài toán điều khiển ổn định, giống như đã làm khi sử dụng bộ điều khiển tối ưu LQR [1,35,57].

Bằng việc đề xuất kỹ thuật mới trong thiết kế bộ điều khiển dự báo phi tuyến, tổng kết lại bằng các thuật toán khả thi và dễ cài đặt, luận án có những ý nghĩa lý thuyết khoa học như sau:

Khẳng định được chất lượng và tính ứng dụng cao vào thực tế của các bộ điều khiển (các thuật toán) vào thực tế điều khiển các đối tượng công nghiệp.

Đóng góp thêm các ứng dụng của các bộ lọc Kalman mở rộng trong các bộ điều khiển dự báo phản hồi đầu ra được thiết kế theo nguyên lý tuyến tính hóa từng đoạn mô hình dự báo.

Đề xuất và chứng minh định lý ổn định, qua đó khẳng định được tính triệt để của các bộ điều khiển dự báo đề xuất.

Ý nghĩa thực tiễn

Nhu cầu vận dụng các kiến thức của khoa học điều khiển luôn luôn hiện hữu trong mọi quá trình sản xuất. Chính vì vậy, mục tiêu ban đầu của luận án đó là ứng dụng được các bộ điều khiển dự báo các đối tượng phi tuyến trong công nghiệp. Luận án đã đáp ứng được nhu cầu thực tiễn trên, với các ý nghĩa thực tiễn cụ thể như sau:

Cung cấp được các bộ điều khiển dự báo (cụ thể bằng các thuật toán) cho các đối tượng phi tuyến trong công nghiệp.

Thiết kế và kiểm chứng chất lượng các bộ điều khiển dự báo phản hồi đầu ra trên cơ sở ứng dụng các bộ lọc Kalman mở rộng cho các đối tượng: Con lắc ngược và con lắc ngược quay bằng mô phỏng và trên mô hình thí nghiệm.

Bố cục của luận án

Luận án có bố cục gồm 3 chương. Sau chương 1 trình bày về những kết quả đã có của kỹ thuật điều khiển dự báo phản hồi đầu ra với các ý kiến nhận xét phân tích riêng về từng phương pháp cụ thể, trong chương 2 luận án trình bày chi tiết các kỹ thuật cải tiến để nâng cao khả năng ứng dụng của kỹ thuật này cho các đối tượng phi tuyến trong công nghiệp. Trong chương 3 luận án sẽ chứng minh tính khả dụng

của các đề xuất cải tiến này trên một số đối tượng phi tuyến cụ thể thông qua thực nghiệm mô phỏng và thí nghiệm trên hệ thống thực. Cuối cùng, luận án sẽ tổng kết lại các kết quả cơ bản đã đạt được, những vấn đề còn tồn tại, các phương hướng khắc phục và những vấn đề cần được nghiên cứu tiếp sau này để hoàn thiện.

Những đóng góp của luận án

Đóng góp mới của luận án, tập trung chính ở chương 2 và chương 3, bao gồm các nội dung sau:

1) Bên cạnh các phương pháp lọc Kalman mở rộng (EKF) truyền thống cho hệ phi tuyến, luận án đã đề xuất một phương pháp khác cũng để mở rộng lọc Kalman tuyến tính (KF) cho việc áp dụng vào hệ phi tuyến trên cơ sở tuyến tính hóa từng đoạn mô hình phi tuyến của hệ dọc theo trục thời gian và được dịch chuyển trên trục thời gian cùng với cửa sổ dự báo của bộ điều khiển dự báo. Cụ thể, luận án đã xây dựng được:

- a) Thuật toán 2.1 để quan sát trạng thái hệ song tuyến.
- b) Thuật toán 2.2 để quan sát trạng thái hệ phi tuyến.

Khả năng áp dụng của hai thuật toán trên vào thực tế cũng đã được luận án thực hiện mô phỏng trên: Hệ song tuyến theo tín hiệu vào (2.14), (2.15) ở ví dụ 2.1 và ví dụ 2.2, kết quả mô phỏng thu được đã xác nhận chất lượng tốt của bộ quan sát này.

2) Luận án đã xây dựng được phương pháp điều khiển dự báo phản hồi trạng thái hệ phi tuyến trên cơ sở sử dụng mô hình dự báo tuyến tính từng đoạn với cửa sổ dự báo hữu hạn, mà cụ thể là đã xây dựng được các thuật toán:

- a) Thuật toán 2.3 và Thuật toán 2.4 để điều khiển phản hồi trạng thái hệ song tuyến.

- b) Thuật toán 2.5 và Thuật toán 2.6 để điều khiển phản hồi trạng thái hệ phi tuyến.

Khả năng áp dụng của các thuật toán trên vào thực tế cũng đã được luận án mô phỏng với: Hệ con lắc ngược và con lắc ngược quay

Kết quả mô phỏng thu được đã xác nhận chất lượng tốt của bộ điều khiển dự báo phi tuyến sử dụng mô hình dự báo tuyến tính từng đoạn này, đúng như nhận định từ lý thuyết.

- 3) Bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách trên cơ sở ghép chung bộ quan sát trạng thái Kalman và bộ điều khiển dự báo phản hồi trạng thái do luận án đề xuất. Chi tiết các bước làm việc của bộ điều khiển này đã được luận án thể hiện ở Thuật toán 2.7 và phiên bản chỉnh sửa của nó dành riêng cho hệ song tuyến.

Khả năng áp dụng của thuật toán trên vào thực tế cũng đã được luận án mô phỏng thành công trên: Hệ con lắc ngược và con lắc ngược quay.

Kết quả mô phỏng thu được cũng đã khẳng định tính khả dụng cao của phương pháp vào thực tế công nghiệp.

- 4) Tiến hành thí nghiệm kiểm chứng lý thuyết trên mô hình thực: đối tượng con lắc ngược quay tại Phòng thí nghiệm Đo lường – Điều khiển của Trường Đại học Kỹ thuật Công nghiệp. Kết quả thí nghiệm đã kiểm chứng tính đúng đắn của các thuật toán đề xuất trong luận án đồng thời khẳng định tính khả dụng vào thực tế của phương pháp đề xuất và hoàn toàn phù hợp với nhận định lý thuyết.

CHƯƠNG 1 TỔNG QUAN VỀ PHƯƠNG PHÁP ĐIỀU KHIỂN DỰ BẢO PHẢN HỒI ĐẦU RA

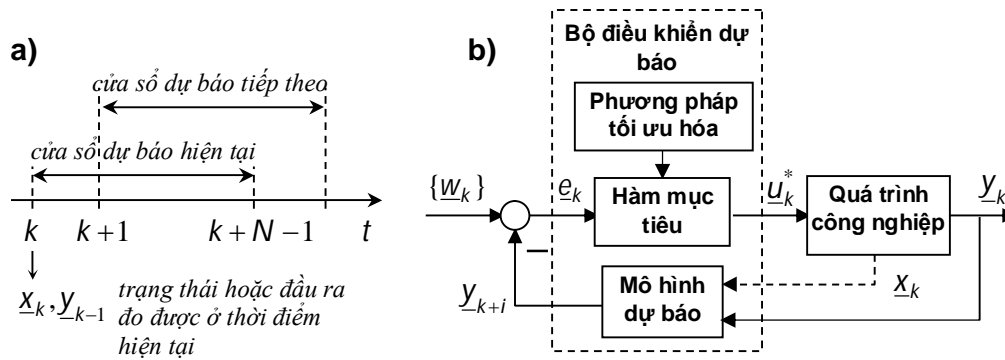
Theo [25] thì để phân biệt điều khiển dự báo tuyến tính và phi tuyến, người ta đã sử dụng khái niệm điều khiển dự báo tuyến tính như sau: Một hệ điều khiển dự báo sẽ được gọi là tuyến tính khi nó thỏa mãn đồng thời các giả thiết sau:

Mô hình dự báo là tuyến tính.

Hàm mục tiêu đánh giá chất lượng hệ thống ở dạng toàn phương theo tín hiệu điều khiển.

Các tập ràng buộc mô tả điều kiện giới hạn về tín hiệu điều khiển và trạng thái hệ thống là những tập lồi.

Tất nhiên bên cạnh khái niệm vừa trình bày trên cũng còn nhiều những khái niệm khác không tương đương về điều khiển dự báo phi tuyến, chẳng hạn như theo [61] thì chỉ cần đối tượng điều khiển là tuyến tính thì hệ điều khiển dự báo cũng đã được gọi là tuyến tính. Trong số nhiều các khái niệm khác nhau đó, luận án sẽ nhất quán sử dụng khái niệm điều khiển dự báo phi tuyến nêu trong [25].



Hình 1.1: Cấu trúc hệ điều khiển dự báo

Hình 1.1 mô tả cấu trúc cơ bản của hệ điều khiển dự báo phản hồi trạng thái để điều khiển đối tượng điều khiển (quá trình) có mô hình không liên tục:

$$\begin{cases} \underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{u}_k) \\ \underline{y}_k = \underline{g}(\underline{x}_k, \underline{u}_k) \end{cases} \quad (1.1)$$

trong đó:

- $\underline{x}_k = \underline{x}(kT_a) \in \mathbb{R}^n$ là vector của n giá trị biên trạng thái trong hệ tại thời điểm $t = kT_a$ với T_a là chu kỳ trích mẫu,
- $\underline{u}_k = \underline{u}(kT_a) \in \mathbb{R}^m$ là vector của m giá trị các tín hiệu vào (tín hiệu điều khiển), và
- $\underline{y}_k = \underline{y}(kT_a) \in \mathbb{R}^r$ là vector của r giá trị các tín hiệu ra. sao cho đầu ra \underline{y}_k bám theo được quỹ đạo mẫu \underline{w}_k cho trước. Bộ điều khiển này làm việc theo chu kỳ lặp. Độ lớn của chu kỳ lặp đó đúng bằng chu kỳ trích mẫu tín hiệu T_a của các tín hiệu vào $\underline{u}(t)$ và ra $\underline{y}(t)$ của quá trình, tức là của đối tượng điều khiển. Tại mỗi thời điểm trích mẫu $k = 0, 1, \dots$ của tín hiệu vào $\underline{u}(t)$ để có $\underline{u}_k = \underline{u}(kT_a)$ và ra $\underline{y}(t)$ để có $\underline{y}_k = \underline{y}(kT_a)$, bộ điều khiển sẽ dựa vào mô hình dự báo, thường được xây dựng trực tiếp từ mô hình toán (1.1) mô tả đối tượng điều khiển, mà xác định dãy các giá trị tín hiệu điều khiển trong tương lai tính từ thời điểm k hiện tại, tức là dãy giá trị:

$$\underline{\mathbf{u}} = \{\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N-1}\}$$

trong khoảng thời gian $[k, N)$, được gọi là khoảng dự báo hiện tại (Hình 1.1a), sao cho với nó hàm mục tiêu được xây dựng từ chất lượng hệ thống:

$$J(\underline{\mathbf{u}}) = \sum_{i=0}^{N-1} q_j(\underline{u}_{k+i}, \underline{y}_{k+i}) \quad (1.2)$$

đạt giá trị nhỏ nhất. Nói cách khác bộ điều khiển dự báo cần xác định nghiệm tối ưu:

$$\underline{\mathbf{u}}^* = \arg \min_{\underline{\mathbf{u}} \in U} J(\underline{\mathbf{u}}) \quad (1.3)$$

với U là điều kiện ràng buộc của bài toán về tín hiệu điều khiển. Thông thường ở điều khiển quá trình, tập ràng buộc này có dạng siêu diện, tức là một tập lồi.

Trong số các giá trị tối ưu tìm được từ (1.3) của cửa sổ dự báo hiện tại:

$$\underline{\mathbf{u}}^* = \{\underline{u}_k^*, \underline{u}_{k+1}^*, \dots, \underline{u}_{k+N-1}^*\}$$

giá trị đầu tiên là \underline{u}_k^* sẽ được đưa vào điều khiển đối tượng tại thời điểm k .

Hình 1.1b biểu diễn cấu trúc bên trong của bộ điều khiển dự báo. Nó gồm 3 thành phần chính, đó là [25,40,61]:

1. *Khối mô hình dự báo.* Khối này có nhiệm vụ xác định dãy tín hiệu ra tương lai \underline{y}_{k+i} trong cửa sổ dự báo hiện tại $[k, N)$ dưới dạng các hàm phụ thuộc $\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+i}$ là những biến cần tìm trong cửa sổ dự báo hiện tại:

$$\underline{y}_{k+i} = \underline{d}_i(\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N-1})$$

Chẳng hạn, nếu sử dụng ngay mô hình (1.1) của đối tượng làm mô hình dự báo, thì khi đã có trạng thái \underline{x}_k đo được ở thời điểm k hiện tại, ta sẽ có:

$$\begin{aligned} \underline{y}_{k+i} &= \underline{g}(\underline{x}_{k+i}, \underline{u}_{k+i}) = \underline{g}(f(\underline{x}_{k+i-1}, \underline{u}_{k+i-1}), \underline{u}_{k+i}) \\ &= \underline{g}(f(f(\underline{x}_{k+i-2}, \underline{u}_{k+i-2}), \underline{u}_{k+i-1}), \underline{u}_{k+i}) \\ &\vdots \\ &= \underline{g}(f(\dots (f(\underline{x}_k, \underline{u}_k), \underline{u}_{k+1}), \dots), \underline{u}_{k+i}) \\ &\triangleq \underline{d}_i(\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N-1}) = \underline{d}_i(\underline{\mathbf{u}}) \end{aligned} \quad (1.4)$$

trong đó \triangleq là ký hiệu định nghĩa hàm $\underline{d}_i(\underline{\mathbf{u}})$.

2. *Khối hàm mục tiêu.* Đây là khối xác định hàm mục tiêu (1.2) cho bài toán tối ưu hóa (1.3) thỏa mãn yêu cầu chất lượng của bài toán điều khiển. Ví dụ để điều khiển đầu ra bám theo được dãy giá trị đặt $\{\underline{w}_k\}$ ở đầu vào thì một trong số các hàm mục tiêu thích hợp sẽ là:

$$\begin{aligned}
J(\underline{\mathbf{u}}) &= \sum_{i=0}^{N'} (\underline{\mathbf{w}}_{k+i} - \underline{\mathbf{y}}_{k+i})^T Q_i (\underline{\mathbf{w}}_{k+i} - \underline{\mathbf{y}}_{k+i}) + \sum_{i=0}^{N-1} \underline{\mathbf{u}}_{k+i}^T R_i \underline{\mathbf{u}}_{k+i} \\
&= \sum_{i=0}^{N'} (\underline{\mathbf{w}}_{k+i} - \underline{\mathbf{d}}_i(\underline{\mathbf{u}}))^T Q_i (\underline{\mathbf{w}}_{k+i} - \underline{\mathbf{d}}_i(\underline{\mathbf{u}})) + \sum_{i=0}^{N-1} \underline{\mathbf{u}}_{k+i}^T R_i \underline{\mathbf{u}}_{k+i}
\end{aligned} \tag{1.5}$$

với q_i, r_i là những hằng số dương tùy chọn và $N' \leq N$. Tất nhiên hàm mục tiêu (1.5) lúc này là hàm của đối số $\underline{\mathbf{u}} = \{\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N-1}\}$. Ta cũng có thể chọn $N' = N$ và khi chọn cửa sổ dự báo N càng lớn, chất lượng bám sẽ càng tốt.

Ở trường hợp $N' = N$ thì với ký hiệu:

$$\underline{\mathbf{e}}_i = \underline{\mathbf{w}}_{k+i} - \underline{\mathbf{y}}_{k+i} = \underline{\mathbf{w}}_{k+i} - \underline{\mathbf{d}}_i(\underline{\mathbf{u}}), \quad \underline{\mathbf{e}} = (\underline{e}_0, \underline{e}_1, \dots, \underline{e}_{N-1})^T, \quad \mathcal{Q} = \text{diag}(Q_i)$$

và

$$\mathcal{R} = \text{diag}(r_i)$$

hàm mục tiêu (1.5) trên sẽ viết lại được thành:

$$J(\underline{\mathbf{u}}) = \underline{\mathbf{e}}^T \mathcal{Q} \underline{\mathbf{e}} + \underline{\mathbf{u}}^T \mathcal{R} \underline{\mathbf{u}} \tag{1.6}$$

Như vậy có thể thấy để thỏa mãn được một yêu cầu chất lượng điều khiển đặt ra ta có nhiều cách chọn hàm mục tiêu khác nhau. Do đó để tăng tính mềm dẻo cho việc thiết kế bộ điều khiển dự báo cũng như cơ hội sau này chuyển bài toán tối ưu có ràng buộc (1.3) thành bài toán tối ưu không ràng buộc:

$$\underline{\mathbf{u}}^* = \arg \min J(\underline{\mathbf{u}}) \tag{1.7}$$

tài liệu [1] đã giới thiệu cấu trúc hàm mục tiêu có tham số biến đổi như sau:

$$J(\underline{\mathbf{u}}) = \underline{\mathbf{e}}^T \mathcal{Q}_k \underline{\mathbf{e}} + \underline{\mathbf{u}}^T \mathcal{R}_k \underline{\mathbf{u}} \tag{1.8}$$

trong đó $\mathcal{Q}_k, \mathcal{R}_k$ là hai ma trận tham số được thay đổi một cách thích hợp cùng với cửa sổ dự báo, nhằm chuyển bài toán có ràng buộc thành bài toán không ràng buộc.

3. *Khối phương pháp tối ưu hóa.* Đây là khối thực thi bài toán tối ưu hóa có ràng buộc (1.3) hoặc không bị ràng buộc (1.7) bằng những thuật toán được chọn trước. Thuật toán thường được áp dụng nhiều cho bài toán tối ưu không ràng buộc là Newton-Raphson, Gauss-Newton, Levenberg-Marquardt... và cho bài toán có ràng buộc là QP hay SQP... [53].

Một hệ điều khiển dự báo sẽ được gọi là phản hồi đầu ra nếu tín hiệu phản hồi về khâu mô hình dự báo chỉ là giá trị y_k đầu ra đo được của hệ, ngược lại khi sử dụng x_k nó sẽ được gọi là hệ điều khiển dự báo phản hồi trạng thái.

Để có thể làm rõ được một cách tốt nhất những đóng góp mới của luận án, chương này sẽ trình bày tổng quan một cách ngắn gọn nhất những đặc điểm cơ bản của phương pháp điều khiển dự báo đã có trong thời gian qua.

1.1 Điều khiển dự báo phản hồi đầu ra hệ có mô hình tuyến tính

Theo tổng kết của [1] thì “điều khiển dự báo là một trong số các phương pháp điều khiển thu được nhiều thành công trong ứng dụng vào điều khiển các quá trình công nghiệp. Ra đời vào những năm 70 của thế kỷ trước, dưới dạng ban đầu chỉ là phương pháp bổ sung cho việc tự chỉnh định thích nghi tham số bộ điều khiển công nghiệp PID, song điều khiển dự báo đã nhanh chóng cho thấy tính ưu việt của nó so với các phương pháp tự chỉnh thông thường khác, chẳng hạn như phương pháp cực tiểu tương quan (*minimum variance MV*), dự báo Smith (*Smith predictor*), cực tiểu tương quan tổng quát (*generalized minimum variance GMV*) ..., nhất là khi áp dụng vào những quá trình công nghiệp có tính pha không cực tiểu”.

Cũng theo [1,25] thì về mặt ứng dụng thực tế điều khiển dự báo đã được nghiên cứu, phát triển rất nhanh trong thời gian qua. Kể từ thời điểm xuất hiện bộ điều khiển dự báo đầu tiên do các kỹ sư công ty dầu khí Shell giới thiệu năm 1977, cho tới nay đã có khá nhiều phiên bản khác nhau của điều khiển dự báo phản hồi đầu ra được ra đời, khẳng định được vị trí trong ứng dụng vào điều khiển nhiều đối tượng công nghiệp khác nhau. Tuy vậy chúng vẫn chỉ mới dừng lại chủ yếu ở các đối tượng tuyến tính. Các phương pháp này bao gồm [25]:

- Thuật toán điều khiển theo mô hình MAC (Model Algorithmic Control).
- Phương pháp ma trận động học điều khiển DMC (Dynamic Matrix Control)
- Phương pháp điều khiển dự báo tổng quát GPC (Generalized Predictive Control).
- Điều khiển dự báo tuyến tính phản hồi trạng thái.

Ngoài ra, để có thể biến đổi một bộ điều khiển dự báo phản hồi trạng thái thành bộ điều khiển dự báo phản hồi đầu ra, một xu hướng rất tự nhiên là sử dụng thêm bộ quan sát trạng thái. Bộ quan sát trạng thái được luận án quan tâm là *bộ lọc Kalman*. Do đó ở phần tổng quan này luận án cũng sẽ trình bày thêm về khả năng điều khiển dự báo phản hồi đầu ra hệ tuyến tính trên cơ sở ghép nối bộ điều khiển dự báo phản hồi trạng thái và bộ lọc Kalman tuyến tính, được gọi ngắn gọn là *bộ điều khiển phản hồi đầu ra theo nguyên lý tách*.

1.1.1 Phương pháp MAC (Model algorithmic control)

Phương pháp MAC này được xây dựng cho hệ tuyến tính LTI ổn định một vào một ra. Nó sử dụng ngay mô hình của hệ:

$$y_k = \sum_{i=0}^{\infty} g_i u_{k-i} \quad (1.9)$$

làm mô hình dự báo, trong đó $\{g_k\}$ là dãy các giá trị của hàm trọng lượng, tức là đáp ứng của hệ ứng với xung dirac ở đầu vào và y_k, u_k, g_k là ký hiệu giá trị tín hiệu $y(t), u(t), g(t)$ tại thời điểm trích mẫu kT_a với T_a là chu kỳ trích mẫu. Do không thể thực hiện được tổng vô hạn (1.9) nên giá trị dự báo y_{k+i} lấy từ (1.9) được thay bằng tổng hữu hạn:

$$\hat{y}_{k+j} = \sum_{i=0}^M g_i u_{k+j-i} \cdot \quad (1.10)$$

Cùng với mô hình dự báo (1.10) này, hàm mục tiêu (1.6) trở thành:

$$J = \sum_{j=0}^{N-1} \left[(\hat{y}_{k+j} - w_{k+j})^2 + \lambda_j u_{k+j}^2 \right], \quad \lambda_j > 0 \text{ có tùy chọn,} \quad (1.11)$$

trong đó $\{w_k\}$ là quỹ đạo mẫu mà đầu ra của hệ cần phải bám theo và N là độ rộng của cửa sổ dự báo. Tương ứng, nghiệm tối ưu (1.7) sẽ là [1]:

$$u_k^* = -(1, 0, \dots, 0) (G_1^T G_1 + \Lambda)^{-1} G_1^T \mathbf{b} \quad (1.12)$$

trong đó:

$$\mathbf{b} = (\mathbf{G}_2 \mathbf{u}_b + \mathbf{e} - \mathbf{w}), \quad \mathbf{e} = (e_k, e_{k+1}, \dots, e_{k+N-1})^T, \quad e_{k+j} = \hat{y}_{k+j} - w_{k+j}$$

$$\mathbf{G}_1 = \begin{pmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{pmatrix}, \quad \mathbf{G}_2 = \begin{pmatrix} g_M & \cdots & g_2 & g_1 \\ 0 & \cdots & g_3 & g_2 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & g_{N+1} & g_N \end{pmatrix}, \quad \Lambda = \text{diag}(\lambda_j) \quad (1.13)$$

$$\mathbf{u}_b = (u_{k-M}, u_{k-M+1}, \dots, u_{k-1}), \quad \mathbf{w} = (w_k, w_{k+1}, \dots, w_{k+N-1})$$

Một cách tóm tắt thì phương pháp MAC được thể hiện ở thuật toán sau:

Thuật toán 1.1 (MAC [1,25,45])

- 1) Chọn cửa sổ quan sát N và M cho mô hình dự báo. Gán $k=0$ và $\mathbf{u}_b = \underline{0}$.
- 2) Tính u_k^* theo (1.12) và (1.13)
- 3) Đưa u_k^* vào điều khiển đối tượng trong khoảng thời gian $kT_a \leq t < (k+1)T_a$.
- 4) Gán $k := k+1$ và sắp xếp lại mảng \mathbf{u}_b theo thứ tự:

$$\leftarrow (u_{k-M}, u_{k-M+1}, \dots, u_{k-1}) \leftarrow u_k^*$$

rồi quay lại bước 2).

Ngoài ra, cũng theo [25,45] thì thuật toán MAC trên chỉ áp dụng được cho các quá trình SISO ổn định, chỉ có nhiễu hằng (nhiều tải) tác động ở đầu ra và có tính động học đủ chậm.

1.1.2 Phương pháp DMC (Dynamic matrix control)

Phương pháp DMC sử dụng mô hình đáp ứng bước nhảy của hệ tuyến tính SISO ổn định:

$$y_k = \sum_{i=1}^{\infty} h_i \Delta u_{k-i}, \quad \Delta u_k = u_k - u_{k-1} \quad (1.14)$$

làm mô hình dự báo, trong đó $\{h_k\}$ là dãy các giá trị của hàm quá độ (đáp ứng với hàm bước nhảy đơn vị $1(kT_a)$ ở đầu vào). Do mô hình (1.14) có tổng vô hạn, không thể xác định được bằng các phương pháp số, nên mô hình dự báo được thay bằng tổng hữu hạn:

$$\hat{y}_{k+j} = \sum_{i=0}^M h_i \Delta u_{k+j} \quad (1.15)$$

Cùng với mô hình dự báo trên, hàm mục tiêu (1.6) cũng trở thành dạng như đã cho ở phương pháp MAC là (1.11) và bài toán tối ưu (1.7) có nghiệm [1,25]:

$$u_k^* = u_{k-1} - (1, 0, \dots, 0) (H_1^T H_1 + \Lambda)^{-1} H_1^T \mathbf{b} \quad (1.16)$$

trong đó:

$$\begin{aligned} \mathbf{b} &= (H_2 \Delta_b + \mathbf{e} - \mathbf{w}), \quad \mathbf{e} = (e_k, e_{k+1}, \dots, e_{k+N-1})^T, \quad e_{k+j} = \hat{y}_{k+j} - w_{k+j} \\ \Delta_b &= (\Delta u_{k-M}, \Delta u_{k-M+1}, \dots, \Delta u_{k-1})^T, \quad \mathbf{w} = (w_k, w_{k+1}, \dots, w_{k+N-1}) \end{aligned} \quad (1.17)$$

$$H_1 = \begin{pmatrix} h_0 & 0 & \dots & 0 \\ h_1 & h_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{N-1} & h_{N-2} & \dots & h_0 \end{pmatrix}, \quad H_2 = \begin{pmatrix} h_M & \dots & h_2 & h_1 \\ 0 & \dots & h_3 & h_2 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & h_{N+1} & h_N \end{pmatrix}, \quad \Lambda = \text{diag}(\lambda_j)$$

Vậy bộ điều khiển dự báo DMC có cấu trúc như sau:

Thuật toán 1.2 (DMC [1,25])

- 1) Chọn cửa sổ quan sát N và M cho mô hình dự báo. Gán $k=0$ và $\Delta_b = \underline{0}$.
- 2) Tính u_k^* theo (1.16) và (1.17)
- 3) Đưa u_k^* vào điều khiển đối tượng trong khoảng thời gian $kT_a \leq t < (k+1)T_a$.
- 4) Gán $k := k+1$ và sắp xếp lại mảng Δ_b theo thứ tự:

$$\leftarrow (\Delta u_{k-M}, \Delta u_{k-M+1}, \dots, \Delta u_{k-1}) \leftarrow u_k^* - u_{k-1}$$

rồi quay lại bước 2).

Theo [25,45] thì giống như MAC, bộ điều khiển DMC cũng chỉ thích ứng với hệ LTI ổn định, pha cực tiểu. Ngoài ra, so với MAC thì nó còn có tính bền vững cao hơn, dễ cài đặt, song lại hạn chế là cần phải có tham số M, N đủ lớn khi có thêm điều kiện ràng buộc là tín hiệu điều khiển bị chặn.

1.1.3 Phương pháp GPC (Generalized predictive control)

Phương pháp GPC áp dụng cho đối tượng tuyến tính tham số hằng mô tả bởi hàm truyền:

$$G(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{n_b} z^{-n_b}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{n_a} z^{-n_a}} = \frac{B(z^{-1})}{A(z^{-1})} \Leftrightarrow A(q^{-1})y_k = B(q^{-1})\Delta u_{k-1}$$

trong đó $\Delta = 1 - q^{-1}$ và q là ký hiệu của phép dịch trục thời gian, tức là:

$$q^{-i}x_k = x_{k-i} \Leftrightarrow \Delta x_k = x_k - x_{k-1} = (1 - q^{-1})x_k$$

Từ mô hình hàm truyền trên ta có được mô hình dự báo [1,25,45]:

$$\hat{y}_{k+j} = F_j(q^{-1})y_k + G_j(q^{-1})\Delta u_{k+j-1}$$

trong đó $G_j(q^{-1})$, $F_j(q^{-1})$ là hai đa thức bậc $j-1$ và n_a được xác định từ phương trình Diophantine:

$$1 = G_j(q^{-1})A(q^{-1}) + q^{-j}F_j(q^{-1}) \quad (1.18)$$

Cùng với mô hình dự báo (1.18) này, hàm mục tiêu (1.6) ban đầu trở thành:

$$\begin{aligned} J &= \sum_{j=1}^N \left[\phi_j (\hat{y}(k+j|k) - w(k+j))^2 + \lambda_j (\Delta u(k+j-1))^2 \right] \\ &= \mathbf{u}_f^T \left(\mathbf{G}_1^T \Phi \mathbf{G}_1 + \Lambda \right) \mathbf{u}_f + 2\mathbf{b}^T \Phi \mathbf{G}_1 \mathbf{u}_f + \mathbf{b}^T \Phi \mathbf{b} \end{aligned}$$

với ϕ_j , λ_j là các trọng số dương. Do đó bài toán tối ưu hóa (1.7) sẽ có nghiệm:

$$\mathbf{u}_k^* = \mathbf{u}_{k-1} + \Delta \mathbf{u}_k = \mathbf{u}_{k-1} - (1, 0, \dots, 0) \left(\mathbf{G}_1^T \Phi \mathbf{G}_1 + \Lambda \right)^{-1} \mathbf{G}_1^T \Phi^T \mathbf{b} \quad (1.19)$$

trong đó:

$$\begin{aligned}
\Phi &= \text{diag}(\phi_j), \Lambda = \text{diag}(\lambda_j), \mathbf{b} = (\mathbf{G}_2 \mathbf{u}_b + \mathbf{F} \mathbf{y}_b - \mathbf{w}) \\
\mathbf{w} &= (w_{k+1}, w_{k+2}, \dots, w_{k+N})^T, \mathbf{u}_b = (\Delta u_{k-1}, \Delta u_{k-2}, \dots, \Delta u_{k-n_b})^T \\
\mathbf{y}_b &= (y_k, y_{k-1}, \dots, y_{k-n_a}), \mathbf{u}_f = (\Delta u_k, \Delta u_{k+1}, \dots, \Delta u_{k+N-1}) \\
\mathbf{G}_1 &= \begin{pmatrix} g_{1,0} & 0 & \cdots & 0 \\ g_{2,1} & g_{2,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{M,M-1} & g_{M,M-2} & \cdots & g_{M,0} \end{pmatrix}, \mathbf{G}_2 = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,n_b} \\ g_{2,2} & g_{2,3} & \cdots & g_{2,n_b+1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{M,M} & g_{M,M+1} & \cdots & g_{M,n_b+M-1} \end{pmatrix} \\
\mathbf{F} &= \begin{pmatrix} f_{1,0} & f_{1,1} & \cdots & f_{1,n_a} \\ f_{2,0} & f_{2,1} & \cdots & f_{2,n_a} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M,0} & f_{M,1} & \cdots & f_{M,n_a} \end{pmatrix}
\end{aligned}$$

và $g_{j,i}, f_{j,i}$ là các hệ số của hai đa thức nghiệm phương trình Diophantine (1.18):

$$\begin{aligned}
G_j(q^{-1}) &= g_{j,0} + g_{j,1}q^{-1} + \cdots + g_{j,n_b+j-1}q^{-(n_b+j-1)} \\
F_j(z^{-1}) &= f_{j,0} + f_{j,1}q^{-1} + \cdots + f_{j,n_a}q^{-n_a}
\end{aligned} \tag{1.20}$$

Từ đây ta có bộ điều khiển dự báo GPC như mô tả ở thuật toán sau:

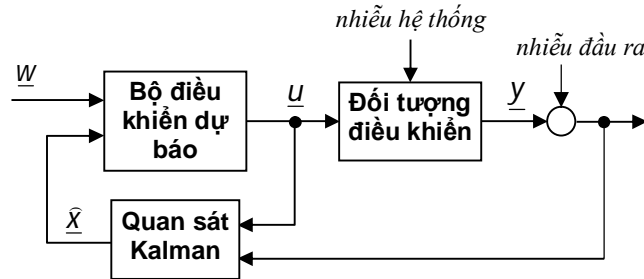
Thuật toán 1.3 (GPC [1,25,45])

- 1) Chọn cửa sổ quan sát N và M cho mô hình dự báo.
- 2) Tính N cặp đa thức $G_j(q^{-1}), F_j(q^{-1}), j=1, \dots, N$ là nghiệm của phương trình Diophantine (1.18) và từ đó là các hệ số $g_{j,i}, f_{j,i}$ của chúng theo (1.20).
- 3) Gán $k=0$.
- 4) Tính u_k^* theo (1.19) và đưa u_k^* vào điều khiển đối tượng trong khoảng thời gian $kT_a \leq t < (k+1)T_a$.
- 5) Gán $k := k+1$ rồi quay lại bước 4).

Khác với MAC và DMC, phương pháp GPC áp dụng được cho những đối tượng, quá trình không ổn định hoặc pha không cực tiểu, và cho tới nay nó được biết là phương pháp điều khiển dự báo phổ thông và áp dụng nhiều nhất trong thực tế. Ngoài ra GPC cũng dễ dàng được mở rộng cho cả hệ MIMO.

Nhược điểm chính là GPC cần phải sử dụng tới thuật toán tìm nghiệm phương trình Diophantine.

1.1.4 Điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ có mô hình tuyến tính



Hình 1.2: Điều khiển phản hồi đầu ra theo nguyên lý tách

Hình 1.2 mô tả hệ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách sử dụng bộ lọc Kalman để xác định xấp xỉ giá trị trạng thái $\hat{x}_k \approx x_k$ từ các giá trị vào ra u_k, y_k đo được từ đối tượng điều khiển rồi đưa về bộ điều khiển dự báo phản hồi trạng thái. Nếu một bộ quan sát trạng thái nói chung, hay bộ lọc Kalman nói riêng, với trạng thái xấp xỉ \hat{x}_k không làm thay đổi chất lượng hệ thống mà bộ điều khiển dự báo phản hồi đầu ra mang lại khi sử dụng trạng thái thực có x_k của hệ thống, thì người ta nói hệ điều khiển phản hồi đầu ra đó đã thỏa mãn nguyên lý tách [19] (*separation principle*).

Nếu so sánh với Hình 1.1b) trước đây thì khối *bộ điều khiển dự báo* ở Hình 1.2 này sẽ gồm cả ba khối con là *khối mô hình dự báo*, *khối hàm mục tiêu* và *khối tối ưu hóa*.

Điều khiển dự báo phản hồi trạng thái

Điều khiển dự báo phản hồi trạng thái hệ (1.1), nhưng bây giờ được giả thiết là LTI, tức là hệ có mô hình:

$$\begin{cases} \underline{x}_{k+1} = A\underline{x}_k + B\underline{u}_k \\ \underline{y}_k = C\underline{x}_k + D\underline{u}_k \end{cases} \quad (1.21)$$

Khi đó, cùng với các ký hiệu:

$$\begin{aligned} \underline{w} &= \text{col}(\underline{w}_k, \underline{w}_{k+1}, \dots, \underline{w}_{k+N-1}) \\ \underline{y} &= \text{col}(\underline{y}_k, \underline{y}_{k+1}, \dots, \underline{y}_{k+N-1}) \\ \underline{u} &= \text{col}(\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N-1}) \end{aligned}$$

hàm mục tiêu (1.6) sẽ viết lại được thành:

$$\begin{aligned} J(\underline{u}) &= \underline{e}^T Q \underline{e} + \underline{u}^T \mathcal{R} \underline{u} \\ &= (\Phi \underline{x}_k + \Psi \underline{u} - \underline{w})^T Q (\Phi \underline{x}_k + \Psi \underline{u} - \underline{w}) + \underline{u}^T \mathcal{R} \underline{u} \\ &= \underline{u}^T (\Psi^T Q \Psi + \mathcal{R}) \underline{u} + 2 [\Psi^T Q (\Phi \underline{x}_k + \underline{w})]^T \underline{u} + (\Phi \underline{x}_k - \underline{w})^T Q (\Phi \underline{x}_k - \underline{w}) \end{aligned}$$

trong đó:

$$\Phi = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{pmatrix}, \Psi = \begin{pmatrix} D & \Theta & \dots & \Theta & \Theta \\ CB & D & \dots & \Theta & \Theta \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{N-2}B & CA^{N-3}B & \dots & CB & D \end{pmatrix} \quad (1.22)$$

và Θ là ký hiệu ma trận có tất cả các phần tử bằng 0. Suy ra bài toán tối ưu không ràng buộc (1.7) lúc này sẽ có nghiệm là:

$$\underline{u}^* = -(\Psi^T Q \Psi + \mathcal{R})^{-1} \Psi^T Q (\Phi \underline{x}_k + \underline{w}) \quad (1.23)$$

Thuật toán 1.4 (điều khiển dự báo phản hồi trạng thái hệ tuyến tính [12,31])

- 1) Chọn cửa sổ quan sát N và hai ma trận Q, \mathcal{R} đối xứng xác định dương có số chiều tương ứng. Xây dựng các ma trận Φ, Ψ theo (1.22). Gán $k = 0$.
- 2) Đo \underline{x}_k và tính \underline{u}^* theo (1.23).
- 4) Đưa:

$$\underline{u}_k^* = (I, \Theta, \dots, \Theta) \underline{u}^*$$

vào điều khiển đối tượng trong khoảng thời gian $kT_a \leq t < (k+1)T_a$.

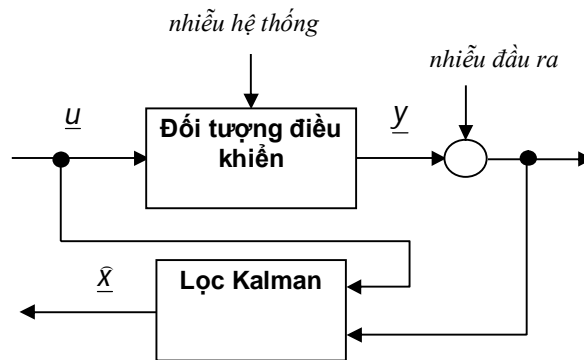
- 5) Gán $k := k+1$ rồi quay lại bước 2).

Trong trường hợp bài toán điều khiển có ràng buộc $\underline{u}_k \in U$ về tín hiệu điều khiển thì thay vì sử dụng công thức (1.23) người ta phải sử dụng đến phương pháp tối ưu QP (quadratic programming) đã được trình bày chi tiết trong [53].

Phương pháp điều khiển dự báo phản hồi trạng thái trình bày trong Thuật toán 1.4 như trên đã được chứng minh trong thực tế là một bộ điều khiển có tính bền vững cao với các nhiễu hằng số từng đoạn ở đầu ra [12,25,45]. Nhược điểm của phương pháp này là các biến trạng thái của hệ phải được giả thiết là đo được trực tiếp.

Quan sát trạng thái hệ tuyến tính với lọc Kalman

Để sử dụng bộ điều khiển dự báo trên cho trường hợp không đo được trực tiếp trạng thái \underline{x}_k từ hệ, ta sẽ sử dụng thêm bộ lọc Kalman để xác định xấp xỉ trạng thái từ các tín hiệu đo được khác mà ở đây chính là các tín hiệu vào và ra (Hình 1.3). Cùng với bộ lọc Kalman này, bộ điều khiển dự báo phản hồi trạng thái sẽ trở thành bộ điều khiển dự báo phản hồi đầu ra như đã được mô tả ở Hình 1.2.



Hình 1.3: Xác định xấp xỉ trạng thái của hệ nhờ lọc Kalman

Lọc Kalman tuyến tính (KF) đã được trình bày chi tiết trong [34,39], là bộ lọc liên quan tới hệ tuyến tính tham số biến đổi theo thời gian, bị nhiễu ồn trắng tác động cả trong hệ thống và đầu ra, mô tả bởi:

$$\begin{cases} \underline{x}_{k+1} = \mathbf{A}_k \underline{x}_k + \mathbf{B}_k \underline{u}_k + \underline{\xi}_k \\ \underline{y}_k = \mathbf{C}_k \underline{x}_k + \mathbf{D}_k \underline{u}_k + \underline{\zeta}_k \end{cases} \quad (1.24)$$

với $\underline{\xi}_k, \underline{\zeta}_k$ là các nhiễu phân bố chuẩn, có kỳ vọng bằng 0 (nhiều Gauss) và:

$$\begin{aligned} \underline{x}_k \in \mathbb{R}^n, \underline{u}_k \in \mathbb{R}^m, \underline{\xi}_k \in \mathbb{R}^n, \underline{y}_k \in \mathbb{R}^r, \underline{\zeta}_k \in \mathbb{R}^r, A_k \in \mathbb{R}^{n \times n}, B_k \in \mathbb{R}^{n \times m}, \\ C_k \in \mathbb{R}^{r \times n}, D_k \in \mathbb{R}^{r \times m}, M\left\{\begin{matrix} \underline{\xi}_k \\ \underline{\zeta}_k \end{matrix}\right\} = M\left\{\begin{matrix} \underline{\xi}_k \\ \underline{\zeta}_k \end{matrix}\right\} = 0, \\ M\left\{\begin{matrix} \underline{\xi}_k \underline{\xi}_k^T \\ \underline{\zeta}_k \underline{\zeta}_k^T \end{matrix}\right\} = \Delta(k-i)\Psi_k, M\left\{\begin{matrix} \underline{\xi}_k \underline{\zeta}_i^T \\ \underline{\zeta}_k \underline{\xi}_i^T \end{matrix}\right\} = \Delta(k-i)\Phi_k \end{aligned} \quad (1.25)$$

trong đó $M\{\cdot\}$ là ký hiệu phép tính lấy kỳ vọng.

Chọn khoảng thời gian quan sát T_a đúng bằng chu kỳ trích mẫu. Khi đó bộ lọc Kalman không liên tục có cấu trúc tương tự bộ quan sát Luenberger [1]. Ở đây ta sử dụng hai giá trị ước lượng trung gian $\underline{x}_k(-)$ và $\underline{x}_k(+)$ để ước lượng giá trị thực của \underline{x}_k theo các công thức sau:

$$\begin{aligned} \underline{x}_k(-) &= A_{k-1}\underline{x}_{k-1}(+) + B_{k-1}\underline{u}_{k-1} \\ \underline{x}_k(+) &= \underline{x}_k(-) + K_k(\underline{y}_k - C_k\underline{x}_k(-) - D_k\underline{u}_k) \end{aligned} \quad (1.26)$$

trong đó K_k chưa biết, và cần phải được xác định theo tiêu chuẩn tối ưu:

$$K_k = \arg \min M\left\{(\underline{x}_k - \underline{x}_k(+))^2\right\}$$

Nếu so sánh với cấu trúc bộ quan sát Luenberger:

$$\bar{\underline{x}}_k = A_{k-1}\bar{\underline{x}}_{k-1} + B_{k-1}\underline{u}_{k-1} + K_k(\underline{y}_{k-1} - C_k\bar{\underline{x}}_{k-1} - D_k\underline{u}_{k-1})$$

ta thấy ngay được là trong bộ quan sát Kalman (1.26) giá trị $\underline{x}_k(+)$ cũng chính là giá trị trạng thái quan sát được $\bar{\underline{x}}_k$ của hệ ở thời điểm k , tức là $\underline{x}_k(+) = \bar{\underline{x}}_k$ cũng như thành phần $\underline{y}_{k-1} - C_k\bar{\underline{x}}_{k-1} - D_k\underline{u}_{k-1}$ nay được thay bởi $\underline{y}_k - C_k\underline{x}_k(-) - D_k\underline{u}_k$ nhằm tận dụng luôn cặp giá trị vào ra đo được $\underline{u}_k, \underline{y}_k$ ở cùng thời điểm k đó.

Nếu định nghĩa thêm:

- Hai giá trị sai lệch: $e_k(+) = \underline{x}_k(+) - \underline{x}_k$
 $e_k(-) = \underline{x}_k(-) - \underline{x}_k$
- Giá trị hỗ tương quan của sai các lệch đó:

$$P_k(+) = M\left\{e_k(+)e_k^T(+)\right\}$$

$$P_k(-) = M\left\{e_k(-)e_k^T(-)\right\}$$

thì bài toán tối ưu trên rút gọn được thành:

$$\begin{aligned}\underline{x}_k(+)-\underline{x}_k &= \underline{x}_k(-)+K_k C_k (\underline{x}_k-\underline{x}_k(-))+\bar{K}_k \underline{\zeta}_k-\underline{x}_k \\ &= (I-K_k C_k) \underline{e}_k(-)+\bar{K}_k \underline{\zeta}_k\end{aligned}\quad (1.27)$$

$$\Leftrightarrow \underline{e}_k(+)= (I-K_k C_k) \underline{e}_k(-)+K_k \underline{\zeta}_k \quad (1.28)$$

cũng như:

$$\underline{x}_k(-)-\underline{x}_k = A_{k-1} (\underline{x}_{k-1}(+)-\underline{x}_{k-1})-\underline{\xi}_{k-1}$$

$$\Leftrightarrow \underline{e}_k(-)= A_{k-1} \underline{e}_{k-1}(+)-\underline{\xi}_{k-1} \quad (1.29)$$

Khi đó, với nguyên lý trực giao [10] ta được:

$$M \left\{ (\underline{x}_k-\underline{x}_k(+))^2 \right\} \rightarrow \min$$

$$\begin{aligned}\Leftrightarrow \Theta &= M \left\{ (\underline{x}_k-\underline{x}_k(+)) (y_k-C_k \underline{x}_k(-)-D_k \underline{u}_k)^T \right\} \\ &= M \left\{ \left[-(I-K_k C_k) \underline{e}_k(-)-K_k \underline{\zeta}_k \right] \left[C_k \underline{e}_k(-)-\underline{\zeta}_k \right]^T \right\} \\ &= (I-K_k C_k) P_k(-) C_k^T - K_k \Phi_k\end{aligned}$$

với $P_k(-) = M \left\{ \underline{e}_k(-) \underline{e}_k^T(-) \right\}$. Do đó, ta được:

$$K_k = P_k(-) C_k^T \left(C_k P_k(-) C_k^T + \Phi_k \right)^{-1}$$

Để tìm $P_k(-)$ ta dựa vào công thức sau:

$$\begin{aligned}P_k(+)&= M \left\{ \underline{e}_k(+)\underline{e}_k^T(+)\right\} \\ &= M \left\{ \left[(I-K_k C_k)\underline{e}_k(-)+K_k \underline{\zeta}_k \right] \left[(I-K_k C_k)\underline{e}_k(-)+K_k \underline{\zeta}_k \right]^T \right\} \\ &= M \left\{ (I-K_k C_k)\underline{e}_k(-)\underline{e}_k^T(-)(I-K_k C_k)^T + K_k \underline{\zeta}_k \underline{\zeta}_k^T K_k^T \right\} \\ &= (I-K_k C_k) P_k(-) (I-K_k C_k)^T + K_k \Phi_k K_k^T\end{aligned}$$

sẽ có:

$$\begin{aligned}P_k(+)&= (I-K_k C_k) P_k(-) - \underbrace{(I-K_k C_k) P_k(-) C_k^T}_{K_k \Phi_k} K_k^T + K_k \Phi_k K_k^T \\ &= (I-K_k C_k) P_k(-)\end{aligned}$$

Tổng hợp lại ta có thuật toán mô tả bộ lọc Kalman không liên tục cho hệ tuyến tính không dừng như sau (được gọi tắt là KF [34,37,39]):

Thuật toán 1.5 (KF)

- 1) Tùy chọn điều kiện đầu $\underline{x}_0 = \underline{x}_0(+)$ và $P_0(+)$. Gán $k = 1$.
- 2) Tính $\underline{x}_k(-) = A_{k-1}\underline{x}_{k-1}(+) + B_{k-1}\underline{u}_{k-1}$
 $P_k(-) = A_{k-1}P_{k-1}(+)A_{k-1}^T + \Psi_{k-1}$
 $K_k = P_k(-)C_k^T(C_kP_k(-)C_k^T + \Phi_k)^{-1}$
 $P_k(+)= (I - K_kC_k)P_k(-)$
 $\underline{x}_k(+)= \underline{x}_k(-) + K_k(\underline{y}_k - C_k\underline{x}_k(-) - D_k\underline{u}_k)$
- 3) Xuất $\underline{x}_k(+)$ làm giá trị trạng thái quan sát được của hệ (1.24) ở thời điểm k , ký hiệu là $\underline{\hat{x}}_k$.
 Gán $k := k + 1$ và quay về 2.

Thuật toán KF trên được xây dựng tổng quát cho hệ tuyến tính không dừng (1.24) nên tất nhiên nó cũng áp dụng được cho cả hệ tuyến tính dừng (1.21).

Sử dụng KF vào điều khiển dự báo phản hồi đầu ra cho hệ tuyến tính

Nội dung nguyên lý tách trong điều khiển phản hồi đầu ra nói chung và điều khiển dự báo phản hồi đầu ra nói riêng mà ở đó bộ lọc Kalman được sử dụng như một khâu quan sát trạng thái đã được trình bày ở Hình 1.2. Các tài liệu [1,11] cũng đã khẳng định rằng trong hệ phản hồi đầu ra này, bộ lọc Kalman với việc xấp xỉ trạng thái $\underline{x}_k \approx \underline{\hat{x}}_k$ sẽ không làm thay đổi chất lượng mà bộ điều khiển phản hồi trạng thái đã mang lại cho hệ thống.

1.2 Phương pháp điều khiển dự báo phản hồi đầu ra cho hệ có mô hình phi tuyến

Những phương pháp điều khiển dự báo phản hồi đầu ra được áp dụng nhiều trong thực tế vẫn chỉ dừng lại là điều khiển dự báo tuyến tính [25]. Lý do là vì phần lớn các kết quả nghiên cứu về điều khiển dự báo phi tuyến chỉ tập trung vào phản hồi trạng thái [31,40]. Bởi vậy việc ứng dụng điều khiển dự báo vào những quá

trình phi tuyến trong công nghiệp mà ở đó nhiều biến trạng thái không đo được là rất hạn chế.

Để có thể chuyển đổi được các bộ điều khiển dự báo phản hồi trạng thái thành bộ điều khiển phản hồi đầu ra, một xu hướng rất tự nhiên là sử dụng thêm bộ quan sát trạng thái nhằm xác định xấp xỉ trạng thái của đối tượng điều khiển từ những giá trị tín hiệu đo được khác mà ở đây chính là các tín hiệu vào ra của đối tượng (Hình 1.2). Để có thể thực hiện được điều này trước tiên người ta phải xác định được cặp bộ điều khiển phản hồi trạng thái và bộ quan sát trạng thái phù hợp nhau, tức là khi được ghép chung với nhau chúng sẽ tạo ra cùng một chất lượng hệ thống giống như khi chỉ sử dụng một mình bộ điều khiển phản hồi trạng thái [31]. Trong trường hợp như vậy chúng sẽ được gọi là cặp bộ điều khiển phản hồi trạng thái và quan sát trạng thái thỏa mãn nguyên lý tách [11].

Trong khuôn khổ nghiên cứu xây dựng bộ điều khiển dự báo phi tuyến phản hồi đầu ra theo nguyên lý tách như trên, luận án sẽ tập trung nghiên cứu sử dụng bộ lọc Kalman làm bộ quan sát trạng thái hệ phi tuyến và khảo sát, xác định lớp phương pháp điều khiển dự báo phi tuyến phản hồi trạng thái mang tính thỏa mãn nguyên lý tách cùng với bộ lọc Kalman.

1.2.1 Điều khiển dự báo phản hồi trạng thái

Xét hệ phi tuyến có mô hình trạng thái không liên tục dạng tổng quát:

$$\begin{cases} \underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{u}_k) \\ \underline{y}_k = \underline{g}(\underline{x}_k) \end{cases} \quad (1.30)$$

Giả sử ở thời điểm k hiện tại giá trị trạng thái \underline{x}_k là đo được. Khi đó đầu ra dự báo của nó \underline{y}_{k+i} , $i = 1, 2, \dots, N$, xác định theo mô hình (1.30) sẽ là:

$$\begin{aligned} \text{Khi } i = 1: \underline{y}_{k+1} &= \underline{g}(\underline{x}_{k+1}) = \underline{g}(\underline{f}(\underline{x}_k, \underline{u}_k)) = (\underline{g} \circ \underline{f})(\underline{x}_k, \underline{u}_k) \\ &:= \underline{f}_1^g(\underline{x}_k, \underline{u}_k) \end{aligned} \quad (1.31)$$

với $\underline{f}_1^g = \underline{g} \circ \underline{f}$ là ký hiệu của tích của hai hàm \underline{g} và \underline{f} .

$$\begin{aligned}
\text{Khi } i = 2: \underline{y}_{k+2} &= (\underline{g} \circ \underline{f})(\underline{x}_{k+1}, \underline{u}_{k+1}) = (\underline{g} \circ \underline{f})(\underline{f}(\underline{x}_k, \underline{u}_k), \underline{u}_{k+1}) \\
&= (\underline{g} \circ \underline{f} \circ \underline{f})(\underline{x}_k, \underline{u}_k, \underline{u}_{k+1}) \\
&:= \underline{f}_{-2}^g(\underline{x}_k, \underline{u}_k, \underline{u}_{k+1})
\end{aligned} \tag{1.32}$$

trong đó \underline{f}_{-2}^g là ký hiệu của tích của ba hàm \underline{g} , \underline{f} và \underline{f} .

⋮

$$\begin{aligned}
\text{Khi } i = N: \underline{y}_{k+N} &= (\underline{g} \circ \underbrace{\underline{f} \circ \underline{f} \circ \dots \circ \underline{f}}_N)(\underline{x}_k, \underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N}) \\
&:= \underline{f}_{-N}^g(\underline{x}_k, \underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N-1})
\end{aligned} \tag{1.33}$$

với \underline{f}_{-N}^g là ký hiệu của tích của hàm \underline{g} và N hàm \underline{f} .

Từ đây ta có vector hàm sai lệch dự báo \underline{e} đầu ra trong hàm mục tiêu (1.8) là:

$$\underline{e} = \begin{pmatrix} \underline{w}_{k+1} - \underline{f}_{-1}^g(\underline{x}_k, \underline{u}_k) \\ \underline{w}_{k+2} - \underline{f}_{-2}^g(\underline{x}_k, \underline{u}_k, \underline{u}_{k+1}) \\ \vdots \\ \underline{w}_N - \underline{f}_{-N}^g(\underline{x}_k, \underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N}) \end{pmatrix} \tag{1.34}$$

Do \underline{x}_k là đã có nên vector hàm sai lệch dự báo \underline{e} cho bởi (1.34) thực chất là hàm của N biến số:

$$\underline{u} = \{\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N-1}\} \tag{1.35}$$

Bởi vậy ra có thể viết nó gọn lại thành:

$$\underline{e} = \underline{e}(\underline{u})$$

và tương ứng hàm mục tiêu (1.8) trở thành:

$$J(\underline{u}) = \underline{e}(\underline{u})^T \underline{Q}_k \underline{e}(\underline{u}) + \underline{u}^T \underline{R}_k \underline{u} \tag{1.36}$$

Vậy bộ điều khiển dự báo phản hồi trạng thái cho hệ phi tuyến (1.30) sẽ làm việc theo cấu trúc của thuật toán sau:

Thuật toán 1.6 (điều khiển dự báo phản hồi trạng thái hệ phi tuyến)

- 1) Gán $k := 0$. Chọn cửa sổ quan sát N và xây dựng N vector hàm $\underline{f}_{-i}^g(\cdot)$, $i = 1, 2, \dots, N$ theo các công thức (1.31)-(1.33).

- 2) Chọn hai ma trận đối xứng xác định dương $\mathcal{Q}_k, \mathcal{R}_k$.
- 3) Đo \underline{x}_k và tính $\underline{e}(\underline{u})$ theo (1.34).
- 4) Tính nghiệm tối ưu \underline{u}^* của hàm mục tiêu (1.36).
- 5) Đưa:

$$\underline{u}_k^* = (I, \Theta, \dots, \Theta) \underline{u}^*$$

vào điều khiển đối tượng trong khoảng thời gian $kT_a \leq t < (k+1)T_a$.

- 6) Gán $k := k+1$ rồi quay lại bước 2).

Nguyên tắc chung là như vậy, song việc cài đặt Thuật toán 1.6 trên hoàn toàn không đơn giản, thậm chí trong nhiều trường hợp là không thể. Lý do cho tính không khả thi này nằm ở việc xác định các vector hàm $f_i^g(\cdot), i = 1, 2, \dots, N$ và việc tìm nghiệm tối ưu \underline{u}^* của hàm mục tiêu (1.36) lúc này có dạng phi tuyến rất cao.

Để tránh nhược điểm nêu trên của Thuật toán 1.6, sau này, trong chương 2, luận án sẽ đề xuất một hướng đi khác để xây dựng bộ điều khiển dự báo phản hồi trạng thái hệ phi tuyến trên nền tuyến tính hóa từng đoạn mô hình phi tuyến (1.30). thuật toán này sẽ mang tính khả thi cao và dễ cài đặt hơn rất nhiều so với Thuật toán 1.6.

1.2.2 Lọc Kalman mở rộng (EKF-extended Kalman filter)

Bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ phi tuyến là bộ điều khiển dự báo có sử dụng bộ quan sát trạng thái. Trong khuôn khổ đề tài của luận án, bộ quan sát hệ phi tuyến này sẽ là bộ quan sát Kalman áp dụng cho hệ có mô hình phi tuyến, hay còn gọi là lọc Kalman mở rộng.

Lọc Kalman mở rộng, viết tắt là EKF, được hiểu là những ứng dụng của Thuật toán 1.5 (lọc Kalman tuyến tính) vào hệ phi tuyến không liên tục, mô tả bởi:

$$\begin{cases} \underline{x}_{k+1} &= \underline{f}_k(\underline{x}_k, \underline{u}_k) + \underline{\xi}_k \\ \underline{y}_k &= \underline{g}_k(\underline{x}_k, \underline{u}_k) + \underline{\zeta}_k, \quad k = 1, 2, \dots \end{cases} \quad (1.37)$$

trong đó $\underline{\xi}_k, \underline{\zeta}_k$ là nhiễu Gauss có hai ma trận tương quan Ψ, Φ và được giả thiết thêm:

$$\begin{aligned} \underline{x}_k \in \mathbb{R}^n, \underline{u}_k \in \mathbb{R}^m, \underline{\xi}_k \in \mathbb{R}^n, \underline{y}_k \in \mathbb{R}^r, \underline{\zeta}_k \in \mathbb{R}^r, A_k \in \mathbb{R}^{n \times n}, B_k \in \mathbb{R}^{n \times m}, \\ C_k \in \mathbb{R}^{r \times n}, D_k \in \mathbb{R}^{r \times m}, M \left\{ \begin{matrix} \underline{\xi}_k \\ \underline{\zeta}_k \end{matrix} \right\} = M \left\{ \begin{matrix} \underline{\xi}_k \\ \underline{\zeta}_k \end{matrix} \right\} = 0, \\ M \left\{ \begin{matrix} \underline{\xi}_k \underline{\xi}_i^T \\ \underline{\zeta}_k \underline{\zeta}_i^T \end{matrix} \right\} = \Psi \delta_{k-i}, M \left\{ \begin{matrix} \underline{\xi}_k \underline{\zeta}_i^T \\ \underline{\zeta}_k \underline{\xi}_i^T \end{matrix} \right\} = \Phi \delta_{k-i} \end{aligned}$$

trong đó δ_{k-i} là ký hiệu hàm của Kronecker.

Để ứng dụng được Thuật toán 1.5, vốn được xây dựng cho hệ tuyến tính không dừng (1.24), vào hệ phi tuyến (1.37), tất nhiên trước đó ta phải thực hiện tuyến tính hóa. Với giả thiết các vector hàm $\underline{f}_k(\cdot), \underline{g}_k(\cdot)$ là khả vi, ta sẽ có 3 phương pháp tuyến tính hóa cơ bản thường được dùng trong EKF là [12,35,61]:

- Tuyến tính hóa nhờ phép đổi biến vi phân.
- Tuyến tính hóa xung quanh quỹ đạo tiên định (nominal trajectory)
- Tuyến tính hóa các bước tính bên trong Thuật toán 1.5.

Tùy thuộc vào dạng của mô hình phi tuyến (1.37) mà ta có thể áp dụng các cách khác nhau để tiến hành tuyến tính hóa.

A) EKF loại 1: Tuyến tính hóa nhờ phép đổi biến vi phân

Giả sử tồn tại phép đổi biến vi phân (ánh xạ một-một và khả vi):

$$\underline{z}_k = \underline{m}_k(\underline{x}_k) \Leftrightarrow \underline{x}_k = \underline{m}_k^{-1}(\underline{z}_k) \quad (1.38)$$

chuyển hệ (1.37) về được dạng tuyến tính (1.24):

$$\begin{cases} \underline{z}_{k+1} &= A_k \underline{z}_k + B_k \underline{u}_k + \underline{\xi}'_k \\ \underline{y}_k &= C_k \underline{z}_k + D_k \underline{u}_k + \underline{\zeta}_k, \quad k=1, 2, \dots \end{cases} \quad (1.39)$$

với:

$$\begin{aligned} \underline{z}_{k+1} &= \underline{m}_{k+1}(\underline{x}_{k+1}) = \underline{m}_{k+1} \left(\underline{f}_k(\underline{x}_k, \underline{u}_k) + \underline{\xi}_k \right) \\ &= \underline{m}_{k+1} \left(\underline{f}_k(\underline{m}_k^{-1}(\underline{z}_k), \underline{u}_k) + \underline{\xi}_k \right) = A_k \underline{z}_k + B_k \underline{u}_k + \underline{\xi}'_k \\ \underline{y}_k &= \underline{g}_k(\underline{x}_k, \underline{u}_k) + \underline{\zeta}_k = \underline{g}_k(\underline{m}_k^{-1}(\underline{z}_k), \underline{u}_k) + \underline{\zeta}_k \\ &= C_k \underline{z}_k + D_k \underline{u}_k + \underline{\zeta}_k \end{aligned}$$

thì Thuật toán 1.5 sẽ áp dụng được để xác định trạng thái \underline{x}_k cho hệ (1.37) như sau:

Thuật toán 1.7 (EKF loại 1, [12,35,61])

1) Tùy chọn điều kiện đầu \underline{x}_0 và $P_0(+)$.

$$\text{Tính } \underline{z}_0(+) = \underline{m}_0(\underline{x}_0).$$

Gán $k = 1$.

2) Tính $\underline{z}_k(-) = \underline{A}_{k-1}\underline{z}_{k-1}(+) + \underline{B}_{k-1}\underline{u}_{k-1}$

$$P_k(-) = \underline{A}_{k-1}P_{k-1}(+)\underline{A}_{k-1}^T + \Psi_{k-1}$$

$$K_k = P_k(-)C_k^T (C_k P_k(-)C_k^T + \Phi_k)^{-1}$$

$$P_k(+) = (I - K_k C_k) P_k(-)$$

$$\underline{z}_k(+) = \underline{x}_k(-) + K_k (\underline{y}_k - C_k \underline{z}_k(-) - D_k \underline{u}_k)$$

3) Xuất $\underline{x}_k = \underline{m}_k^{-1}(\underline{z}_k(+))$ là giá trị trạng thái quan sát được của hệ (1.37) ở thời điểm k . Gán $k := k + 1$ và quay về 2.

Tất nhiên để sử dụng được Thuật toán 1.7 cho hệ (1.37) ta cần phải xác định được phép đổi biến (1.38) thỏa mãn các điều kiện (1.39), tức là:

$$\underline{m}_{k+1} \left(\underline{f}_k(\underline{m}_k^{-1}(\underline{z}_k), \underline{u}_k) + \underline{\xi}_k \right) = \underline{A}_k \underline{z}_k + \underline{B}_k \underline{u}_k + \underline{\xi}'_k$$

$$\underline{g}_k(\underline{m}_k^{-1}(\underline{z}_k), \underline{u}_k) = C_k \underline{z}_k + D_k \underline{u}_k$$

và điều này phụ thuộc rất nhiều vào cấu trúc mô hình (1.37) của hệ phi tuyến.

B) EKF loại 2: Tuyến tính hóa xung quanh quỹ đạo tiên định

Nếu ở hệ phi tuyến (1.37) tồn tại quỹ đạo tiên định $(\underline{x}_k^{nor}, \underline{u}_k^{nor})$, được hiểu là quỹ đạo thỏa mãn:

$$\underline{x}_{k+1}^{nor} = \underline{f}_k(\underline{x}_k^{nor}, \underline{u}_k^{nor}) \quad (1.40)$$

thì khi xấp xỉ $\underline{f}_k(\underline{x}_k, \underline{u}_k)$ và $\underline{g}_k(\underline{x}_k, \underline{u}_k)$ xung quanh quỹ đạo tiên định (1.40) nhờ phân tích chuỗi Taylor, ta sẽ được:

$$\begin{aligned} \underline{f}_k(\underline{x}_k, \underline{u}_k) &\approx \underline{f}_k(\underline{x}_k^{nor}, \underline{u}_k^{nor}) + \left. \frac{\partial \underline{f}_k}{\partial \underline{x}_k} \right|_{\underline{x}_k^{nor}, \underline{u}_k^{nor}} (\underline{x}_k - \underline{x}_k^{nor}) + \left. \frac{\partial \underline{f}_k}{\partial \underline{u}_k} \right|_{\underline{x}_k^{nor}, \underline{u}_k^{nor}} (\underline{u}_k - \underline{u}_k^{nor}) \\ &= \underline{x}_{k+1}^{nor} + \underline{A}_k(\underline{x}_k - \underline{x}_k^{nor}) + \underline{B}_k(\underline{u}_k - \underline{u}_k^{nor}) \end{aligned}$$

và:

$$\begin{aligned} \underline{g}_k(\underline{x}_k, \underline{u}_k) &\approx \underline{g}_k(\underline{x}_k^{nor}, \underline{u}_k^{nor}) + \left. \frac{\partial \underline{g}_k}{\partial \underline{x}_k} \right|_{\underline{x}_k^{nor}, \underline{u}_k^{nor}} (\underline{x}_k - \underline{x}_k^{nor}) + \left. \frac{\partial \underline{g}_k}{\partial \underline{u}_k} \right|_{\underline{x}_k^{nor}, \underline{u}_k^{nor}} (\underline{u}_k - \underline{u}_k^{nor}) \\ &= \underline{y}_k^{nor} + \underline{C}_k(\underline{x}_k - \underline{x}_k^{nor}) + \underline{D}_k(\underline{u}_k - \underline{u}_k^{nor}) \end{aligned}$$

trong đó:

$$\underline{A}_k = \left. \frac{\partial \underline{f}_k}{\partial \underline{x}_k} \right|_{\underline{x}_k^{nor}, \underline{u}_k^{nor}}, \quad \underline{B}_k = \left. \frac{\partial \underline{f}_k}{\partial \underline{u}_k} \right|_{\underline{x}_k^{nor}, \underline{u}_k^{nor}}, \quad \underline{C}_k = \left. \frac{\partial \underline{g}_k}{\partial \underline{x}_k} \right|_{\underline{x}_k^{nor}, \underline{u}_k^{nor}}, \quad \underline{D}_k = \left. \frac{\partial \underline{g}_k}{\partial \underline{u}_k} \right|_{\underline{x}_k^{nor}, \underline{u}_k^{nor}}$$

và

$$\underline{y}_k^{nor} = \underline{g}_k(\underline{x}_k^{nor}, \underline{u}_k^{nor}) \quad (1.41)$$

hệ phi tuyến (1.37) ban đầu trở thành:

$$\begin{cases} \underline{x}'_{k+1} &= \underline{A}_k \underline{x}'_k + \underline{B}_k \underline{u}'_k + \underline{\xi}_k \\ \underline{y}'_k &= \underline{C}_k \underline{x}'_k + \underline{D}_k \underline{u}'_k + \underline{\zeta}_k \end{cases}$$

$$\text{vói: } \underline{x}'_k = \underline{x}_k - \underline{x}_k^{nor}, \quad \underline{u}'_k = \underline{u}_k - \underline{u}_k^{nor} \quad \text{và} \quad \underline{y}'_k = \underline{y}_k - \underline{y}_k^{nor} \quad (1.42)$$

Từ đây ta có được:

Thuật toán 1.8 (EKF loại 2, [12,35,61])

1) Xác định quỹ đạo $(\underline{x}_k^{nor}, \underline{u}_k^{nor})$ thỏa mãn (1.40). Từ đó tính các ma trận và vector

$\underline{A}_k, \underline{B}_k, \underline{C}_k, \underline{D}_k, \underline{y}_k^{nor}$ theo (1.41) cũng như $\underline{x}'_k, \underline{u}'_k, \underline{y}'_k$ theo (1.42).

2) Tùy chọn điều kiện đầu \bar{x}_0 và $P_0(+)$.

Tính $\underline{x}'_0(+)=\bar{x}_0-\underline{x}_0^{nor}$. Gán $k=1$.

$$\begin{aligned}
3) \text{ Tính } \underline{x}'_k(-) &= \underline{A}_{k-1} \underline{x}'_{k+1}(+) + \underline{B}_{k-1} \underline{u}'_{k-1} \\
P_k(-) &= \underline{A}_{k-1} P_{k-1}(+) \underline{A}_{k-1}^T + \Psi_{k-1} \\
K_k &= P_k(-) C_k^T (C_k P_k(-) C_k^T + \Phi_k)^{-1} \\
P_k(+) &= (I - K_k C_k) P_k(-) \\
\underline{x}'_k(+) &= \underline{x}'_k(-) + K_k (\underline{y}'_k - C_k \underline{x}'_k(-) - D_k \underline{u}'_k)
\end{aligned}$$

3) Xuất $\bar{\underline{x}}_k = \underline{x}'_k(+) + \underline{x}_k^{nor}$ là giá trị trạng thái quan sát được của hệ (1.37) ở thời điểm k . Gán $k := k+1$ và quay về 3.

C) *EKF loại 3: Tuyến tính hóa các bước tính bên trong Thuật toán 1.5.*

Tương tự như công thức (1.27) của KF, thì ở đây, với hệ phi tuyến (1.37) ta cũng định nghĩa hai giá trị ước lượng trung gian $\underline{x}_k(-)$, $\underline{x}_k(+)$ cho \underline{x}_k như sau:

$$\begin{aligned}
\underline{x}_k(-) &= \underline{f}_{-k}(\underline{x}_{k-1}(+), \underline{u}_{k-1}) \\
\underline{x}_k(+) &= \underline{x}_k(-) + K_k [\underline{y}_k - \underline{g}_k(\underline{x}_k(-), \underline{u}_k)]
\end{aligned} \tag{1.43}$$

trong đó ma trận K_k được xác định từ:

$$K_k = \arg \min M \left\{ (\underline{x}_k - \underline{x}_k(+))^2 \right\}$$

Có thể thấy (1.43) có cấu trúc gần giống như bộ quan sát Luenberger mở rộng:

$$\bar{\underline{x}}_k = \underline{f}_{-k}(\bar{\underline{x}}_{k-1}, \underline{u}_{k-1}) + K_k [\underline{y}_{k-1} - \underline{g}_k(\bar{\underline{x}}_{k-1}, \underline{u}_{k-1})]$$

với một chỉnh sửa nhỏ là thành phần $\underline{y}_{k-1} - \underline{g}_k(\bar{\underline{x}}_{k-1}, \underline{u}_{k-1})$ nay được thay bởi $\underline{y}_k - \underline{g}_k(\underline{x}_k(-), \underline{u}_k)$ để tận dụng giá trị đầu $\underline{u}_k, \underline{y}_k$ vừa đo được ở thời điểm hiện tại.

Khi đó, cùng với các đại lượng $\underline{e}_k(-)$, $\underline{e}_k(+)$, $P_k(-)$, $P_k(+)$ đã có của KF, ta thấy ngay được rằng để áp dụng được Thuật toán 1.5 của KF cho hệ phi tuyến (1.37), cần thiết phải tuyến tính hóa hai công thức (1.43).

Sử dụng phân tích chuỗi Taylor cho (1.43) xung quanh $\underline{x}_{k-1}(+)$, $\underline{x}_k(-)$ ta có:

$$\begin{aligned}
\underline{e}_k(-) &= \underline{x}_k(-) - \underline{x}_k = \underline{f}_{-k}(\underline{x}_{k-1}(+), \underline{u}_{k-1}) - \underline{f}_{-k}(\underline{x}_{k-1}, \underline{u}_{k-1}) \\
&\approx \left. \frac{\partial \underline{f}_{-k}(\underline{x}_{k-1}, \underline{u}_{k-1})}{\partial \underline{x}_{k-1}} \right|_{\underline{x}_{k-1}(+)} (\underline{x}_{k-1} - \underline{x}_{k-1}(+)) - \underline{\xi}_k = -\underline{F}_{k-1} \underline{e}_{k-1}(+) - \underline{\xi}_k
\end{aligned} \tag{1.44}$$

với:

$$F_{k-1} = \left. \frac{\partial f_{-k}(\underline{x}_{k-1}, \underline{u}_{k-1})}{\partial \underline{x}_{k-1}} \right|_{\underline{x}_{k-1}(+)}, \quad \text{và } \underline{e}_k(+) = \underline{x}_k(+) - \underline{x}_k \quad (1.45)$$

và:

$$\begin{aligned} \underline{y}_k - \underline{g}_k(\underline{x}_k(-), \underline{u}_k) &= \underline{g}_k(\underline{x}_k, \underline{u}_k) + \underline{\zeta}_k - \underline{g}_k(\underline{x}_k(-), \underline{u}_k) \\ &\approx - \left. \frac{\partial \underline{g}_k(\underline{x}_k, \underline{u}_k)}{\partial \underline{x}_k} \right|_{\underline{x}_k(-)} \underline{e}_k(-) + \underline{\zeta}_k = -\underline{G}_k \underline{e}_k(-) + \underline{\zeta}_k \end{aligned} \quad (1.46)$$

với:

$$\underline{G}_k = \left. \frac{\partial \underline{g}_k(\underline{x}_k, \underline{u}_k)}{\partial \underline{x}_k} \right|_{\underline{x}_k = \underline{x}_k(-)} \quad (1.47)$$

Suy ra:

$$\begin{aligned} \underline{e}_k(+) &= \underline{x}_k(+) - \underline{x}_k = \underline{x}_k(-) + \underline{K}_k \left[\underline{y}_k - \underline{g}_k(\underline{x}_k(-), \underline{u}_k) \right] - \underline{x}_k \\ &\approx \underline{e}_k(-) + \underline{K}_k \left[-\underline{G}_k \underline{e}_k(-) + \underline{\zeta}_k \right] = (\underline{I} - \underline{K}_k \underline{G}_k) \underline{e}_k(-) + \underline{K}_k \underline{\zeta}_k \end{aligned} \quad (1.48)$$

Từ đây, qua so sánh (1.44) với (1.29) cũng như (1.48) với (1.28) ta thấy vai trò của \underline{A}_{k-1} trong (1.29) đã được thay bởi F_{k-1} ở (1.44) và \underline{G}_k trong (1.48) chính là \underline{C}_k trong (1.28). Vậy ta đi đến được:

Thuật toán 1.9 (EKF loại 3, [12,35,61])

1) Tùy chọn điều kiện đầu $\bar{\underline{x}}_0 = \underline{x}_0(+)$ và $P_0(+)$. Gán $k = 1$.

2) Tính $\underline{x}_k(-) = \underline{f}_{-k}(\underline{x}_{k-1}(+), \underline{u}_{k-1})$

\underline{G}_k theo (1.47), F_{k-1} theo (1.45) và:

$$P_k(-) = F_{k-1} P_{k-1}(+) F_{k-1}^T + \Psi_{k-1} \quad .$$

$$\underline{K}_k = P_k(-) \underline{G}_k^T (\underline{G}_k P_k(-) \underline{G}_k^T + \Phi_k)^{-1}$$

$$P_k(+) = (\underline{I} - \underline{K}_k \underline{G}_k) P_k(-)$$

$$\underline{x}_k(+) = \underline{x}_k(-) + \underline{K}_k (\underline{y}_k - \underline{G}_k \underline{x}_k(-))$$

3) Xuất $\bar{\underline{x}}_k = \underline{x}_k(+)$ là giá trị trạng thái quan sát được của hệ (1.37) ở thời điểm k .

Gán $k := k+1$ và quay về 2.

1.2.3 UKF - Unscented Kalman Filter

Lọc UKF (*Unscented Kalman filter*) là bộ lọc Kalman được phát triển cho lớp hệ phi tuyến mà ở đó các nhiễu hệ thống và đầu ra $\underline{\xi}_k, \underline{\zeta}_k$ nằm ẩn trong mô hình phi tuyến của hệ, tức là bộ lọc phát triển cho lớp hệ phi tuyến:

$$\begin{cases} \underline{x}_{k+1} &= \underline{f}_k(\underline{x}_k, \underline{\xi}_k, \underline{u}_k) \\ \underline{y}_k &= \underline{g}_k(\underline{x}_k, \underline{\zeta}_k, \underline{u}_k), \quad k=1, 2, \dots \end{cases} \quad (1.49)$$

trong đó, cũng giống như (1.25), hai vector nhiễu $\underline{\xi}_k, \underline{\zeta}_k$ được giả thiết là ồn trắng.

Phương pháp UKF được xây dựng trên cơ sở nhận định trực quan rằng nếu giá trị hiệp phương sai của nhiễu, vốn được xác định theo công thức tuyến tính, bị lan truyền trong mô hình quan sát phi tuyến, kết quả quan sát sẽ không tốt [44, 45]. Bởi vậy, tài liệu [44] đã đề xuất chỉ sử dụng những dữ liệu ngẫu nhiên tập trung xung quanh giá trị kỳ vọng cho công việc quan sát (gọi là điểm sigma), thay vì sử dụng tất cả các dữ liệu đã có.

Từ đây, nếu như ký hiệu lại:

$$\underline{x}_k^a = \text{col}(\underline{x}_k, \underline{\xi}_k, \underline{\zeta}_k) \in \mathbb{R}^{n+v+s}$$

trong đó n, v, s lần lượt là số chiều của $\underline{x}_k, \underline{\xi}_k, \underline{\zeta}_k$, thì nội dung của phương pháp UKF tập trung vào điểm sigma, sẽ có dạng như sau [44, 45]:

Thuật toán 1.10 (UKF- theo tài liệu [45])

- 1) Chọn các tham số $\mathcal{K} \geq 0, 0 \leq \alpha \leq 1, \beta \geq 0$ và $N > 0$ (càng lớn càng tốt). Tùy chọn dãy $2N+1$ giá trị ngẫu nhiên $\underline{x}_0[0], \underline{x}_0[1], \dots, \underline{x}_0[2N]$ làm điểm khởi phát.
- 2) Xác định các giá trị khởi phát (ứng với $k=0$):

$$\begin{aligned} \bar{\underline{x}}_0 &= M\{\underline{x}_0\}, \quad P_{\underline{x}_0} = M\{(\underline{x}_0 - \bar{\underline{x}}_0)(\underline{x}_0 - \bar{\underline{x}}_0)^T\} \\ \bar{\underline{x}}_0^a &= M\{\underline{x}_0^a\} = M\{\text{col}(\bar{\underline{x}}_0, \underline{0}, \underline{0})\} \\ P_0^a &= M\{(\underline{x}_0^a - \bar{\underline{x}}_0^a)(\underline{x}_0^a - \bar{\underline{x}}_0^a)^T\} = \begin{pmatrix} P_x & \Theta & \Theta \\ \Theta & P_\xi & \Theta \\ \Theta & \Theta & P_\zeta \end{pmatrix} \end{aligned}$$

với P_x, P_ξ, P_ζ là các ma trận con có số chiều tương ứng n, v, s của $\underline{x}_k, \underline{\xi}_k, \underline{\zeta}_k$.

Gán $k = 1$.

3) Tính $2N + 1$ vector:

$$X_{i,k-1}^a = \begin{cases} \underline{x}_{k-1}^a & \text{khi } i = 0 \\ \underline{x}_{k-1}^a + \gamma S_i & \text{khi } i = 1, \dots, N \\ \underline{x}_{k-1}^a - \gamma S_i & \text{khi } i = N + 1, \dots, 2N \end{cases}$$

trong đó S_i là vector cột thứ i của ma trận $S = \sqrt{P_{k-1}^a}$ và $\gamma = \sqrt{\alpha^2(N + \mathcal{K})}$.

4) Sắp xếp lại $X_{i,k-1}^a$, $i = 0, 1, \dots, 2N$ thành:

$$X_{i,k-1}^a = \text{col}\left(X_{i,k-1}^x, X_{i,k-1}^\xi, X_{i,k-1}^\zeta\right)$$

với $X_{i,k-1}^x, X_{i,k-1}^\xi, X_{i,k-1}^\zeta$ có số chiều lần lượt là n, v, s giống như $\underline{x}_k, \underline{\xi}_k, \underline{\zeta}_k$.

5) Tính:

$$X_{i,k|k-1}^x = f\left(X_{i,k-1}^x, X_{i,k-1}^\xi, \underline{u}_{k-1}\right), \quad i = 0, 1, \dots, 2N$$

và từ đó là:

$$\begin{aligned} \bar{\underline{x}}_k^- &= \sum_{i=0}^{2N} w_m^{(i)} X_{i,k|k-1}^x \\ P_{\underline{x}_k}^- &= \sum_{i=0}^{2N} w_c^{(i)} \left(X_{i,k|k-1}^x - \bar{\underline{x}}_k^-\right) \left(X_{i,k|k-1}^x - \bar{\underline{x}}_k^-\right)^T \end{aligned}$$

với các trọng số $w_m^{(i)}, w_c^{(i)}, i = 0, 1, \dots, 2N$ được xác định như sau:

$$\begin{aligned} w_m^{(0)} &= \frac{\lambda}{N + \lambda}, \quad w_c^{(0)} = \frac{\lambda}{N + \lambda} + (1 - \alpha^2 + \beta) \\ w_m^{(i)} &= w_c^{(i)} = \frac{1}{2(N + \lambda)}, \quad i = 1, 2, \dots, 2N \quad \text{cũng như } \lambda = \alpha^2(N + \mathcal{K}) - N \end{aligned}$$

6) Tính:

$$Y_{i,k|k-1} = g\left(X_{i,k-1}^x, X_{i,k-1}^\zeta, \underline{u}_{k-1}\right), \quad i = 0, 1, \dots, 2N$$

và từ đó là:

$$\begin{aligned} \bar{\underline{y}}_k^- &= \sum_{i=0}^{2N} w_m^{(i)} Y_{i,k|k-1} \\ P_{\underline{y}_k} &= \sum_{i=0}^{2N} w_c^{(i)} \left(Y_{i,k|k-1} - \bar{\underline{y}}_k^-\right) \left(Y_{i,k|k-1} - \bar{\underline{y}}_k^-\right)^T \\ P_{\underline{x}_k \underline{y}_k} &= \sum_{i=0}^{2N} w_c^{(i)} \left(X_{i,k|k-1}^x - \bar{\underline{x}}_k^-\right) \left(Y_{i,k|k-1} - \bar{\underline{y}}_k^-\right)^T \quad (\text{các ma trận hiệp phương sai}) \end{aligned}$$

7) Xác định hệ số Kalman:

$$K_k = P_{\underline{x}_k} P_{\underline{y}_k}^{-1} \text{ rồi từ đó là } \underline{\hat{x}}_k = \underline{\hat{x}}_k^- + K_k (\underline{y}_k - \underline{\hat{y}}_k^-) \text{ và } P_{\underline{x}_k} = P_{\underline{x}_k}^- - K_k P_{\underline{y}_k} K_k^T.$$

Xuất $\underline{\hat{x}}_k$ làm giá trị quan sát được của hệ tại k . Gán $k := k+1$ và quay về 3).

Về chất lượng của bộ lọc UKF thì cũng theo nhận định của tài liệu [45], Thuật toán 1.10 (UKF) cho ra chất lượng quan sát chính xác hơn các thuật toán EKF (Thuật toán 1.7 - Thuật toán 1.9), nhất là khi hệ có tính phi tuyến cao và nhiễu lan truyền phi tuyến trong mô hình quan sát. Hơn nữa trong Thuật toán 1.10 ta không cần phải xác định đạo hàm của các hàm phi tuyến $\underline{f}(\cdot)$ và $\underline{g}(\cdot)$ như ở EKF loại 2 và loại 3 (Thuật toán 1.8 và Thuật toán 1.9). Tuy nhiên ở phương pháp UKF người sử dụng cần phải chọn lựa được các tham số $\kappa \geq 0$, $0 \leq \alpha \leq 1$, $\beta \geq 0$ trong khi gợi ý cho việc lựa chọn một cách hợp lý lại chưa có và tất nhiên điều này ảnh hưởng không nhỏ tới kết quả quan sát. Bởi vậy, theo nhận định của tác giả luận án, với lớp hệ phi tuyến có nhiễu $\underline{\xi}_k$, $\underline{\zeta}_k$ mang tính cộng tính với mô hình và các vector hàm $\underline{f}(\cdot)$, $\underline{g}(\cdot)$ là khả vi như mô tả ở công thức (1.37), thì đơn giản hơn cả trong điều khiển theo nguyên lý tách vẫn là các thuật toán quan sát EKF.

1.2.4 Điều khiển dự báo phản hồi đầu ra theo nguyên lý tách với bộ lọc Kalman phi tuyến

Bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ phi tuyến (1.37) bị tác động bởi nhiễu Gauss, như mô tả ở Hình 1.2, được xây dựng trên nền Thuật toán 1.6 (điều khiển dự báo phản hồi trạng thái hệ phi tuyến) và các bộ lọc Kalman mở rộng (Thuật toán 1.7 - Thuật toán 1.9) hoặc UKF (Thuật toán 1.10).

Tuy nhiên, như đã phân tích ở mục 1.2.1 về tính không khả thi trong việc cài đặt của Thuật toán 1.6 nên xu hướng ghép chung dạng tổng quát trên hiện mới chỉ dừng lại ở một số trường hợp đơn giản, chẳng hạn như tính phi tuyến của hệ tổng quát (1.37) chỉ nằm ở đầu ra, hoặc hệ (1.37) chỉ ở dạng song tuyến.

1.3 Một số công trình của các tác giả trong và ngoài nước thời gian gần đây nghiên cứu về điều khiển dự báo phản hồi đầu ra

Như đã trình bày trong phần mở đầu và các mục 1.1, mục 1.2, điều khiển dự báo với những ưu điểm của nó đang ngày càng được các tác giả trong và ngoài nước nghiên cứu ứng dụng vào thực tế công nghiệp. Tuy nhiên, khi sử dụng điều khiển dự báo thì một số cân nhắc quan trọng đó là: tính bền vững của bộ điều khiển, độ chính xác của mô hình dự báo, sự ảnh hưởng của nhiễu đầu vào và nhiễu đo đến hệ thống. Để giảm bớt các vấn đề về sự không chắc chắn, nhiễu và các rối loạn trong hệ thống thì MPC thường được thiết kế kèm theo giả thiết phản hồi đầy đủ (hoàn toàn) các trạng thái. Trong thực tế, toàn bộ trạng thái của đối tượng thường không thể đo được và không phải là có sẵn, điều này thúc đẩy các tác giả trên thế giới nghiên cứu về điều khiển dự báo phản hồi đầu ra, trong đó sử dụng một thuật toán độc lập để ước lượng trạng thái. Thuật toán độc lập để ước lượng trạng thái này có thể là: các bộ quan sát, các bộ lọc, các bộ ước lượng... Một số công trình luận án của các tác giả trên thế giới thời gian gần đây nghiên cứu về điều khiển dự báo phản hồi đầu ra cho hệ phi tuyến trên cơ sở sử dụng các bộ quan sát trạng thái [29,59,42,62], trong đó:

Luận án của tác giả David A. Coop (2016)[29]: Thực hiện nghiên cứu ứng dụng thuật toán ước lượng trạng thái của bộ ước lượng Moving Horizon trong bộ điều khiển MPC phi tuyến.

Luận án của tác giả Rafał Noga. (2015)[59]: Bộ điều khiển dự báo phản hồi đầu ra cho hệ phi tuyến ở đây được tác giả thiết kế dựa trên việc tuyến tính hóa mô hình đối tượng và sử dụng bộ quan sát Hybrid Luenberger và Moving Horizon Estimation để quan sát trạng thái.

Luận án của tác giả Huiping Li. (2013) [42]: Tác giả tiến hành thiết kế bộ quan sát trạng thái dựa trên sai lệch để ứng dụng cho bộ điều khiển MPC phi tuyến.

Luận án của tác giả Rui Huang (2010)[62]: Sử dụng bộ ước lượng Moving Horizon Estimate (MHE) và sau đó là bộ quan sát EKF để quan sát trạng thái. Kết quả dừng lại ở mô phỏng cho đối tượng cụ thể là hệ thống điều khiển quá trình.

Các công trình trên dừng lại ở việc thiết kế thành công các bộ điều khiển dự báo phản hồi đầu ra cho hệ phi tuyến trên cơ sở sử dụng các bộ quan sát trạng thái cho một số các đối tượng cụ thể. Các kết quả mô phỏng được sử dụng để kiểm chứng.

Ngoài ra còn một số các công trình của các tác giả nước ngoài khác [49,17,38,65...] cũng đã công bố nghiên cứu thiết kế bộ điều khiển dự báo phản hồi đầu ra cho hệ phi tuyến trên cơ sở sử dụng các bộ quan sát trạng thái như: high gain observer, EKF, UKF tuy nhiên cũng chỉ dừng lại ở việc thiết kế cho các đối tượng cụ thể. Vấn đề tổng quát hóa phương pháp, đưa ra những thuật toán dễ cài đặt, cũng như chứng minh tính ổn định của hệ kín bằng lý thuyết và kiểm chứng phương pháp bằng mô phỏng, thực nghiệm cũng chưa được thực hiện.

Cùng với xu hướng nghiên cứu trên thế giới, thời gian gần đây các tác giả trong nước cũng đã công bố một số công trình nghiên cứu về điều khiển dự báo phản hồi đầu ra [12,57,1,41,4]:

Trong tài liệu [12], tác giả đã đề xuất ý tưởng về tuyến tính hóa từng đoạn mô hình dự báo với cửa sổ dự báo vô hạn trong thiết kế bộ điều khiển dự báo hệ phi tuyến. Phương pháp đề xuất này chỉ có thể áp dụng điều khiển ổn định, muốn điều khiển bám thì phải thêm phép chuyển đổi trung gian. Việc xử lý các điều kiện ràng buộc kèm theo phải xử lý gián tiếp thông qua chọn các tham số Q_k, R_k trong hàm mục tiêu. Trong tài liệu [57], tác giả cũng sử dụng ý tưởng tuyến tính hóa từng đoạn trong việc thiết kế bộ điều khiển phản hồi đầu ra cho hệ thống song tuyến. Cửa sổ dự báo ở đây vẫn là vô hạn.

Trong tài liệu [1], tác giả luận án đã cụ thể hóa ý tưởng tuyến tính hóa từng đoạn mô hình dự báo cho lớp đối tượng phi tuyến có mô hình dạng song tuyến không liên tục. Trong tài liệu này, tác giả đã có những đóng góp mới, đó là:

- Sử dụng mô hình tuyến tính hóa từng đoạn làm mô hình dự báo trong thiết kế bộ điều khiển dự báo.
- Thiết kế được 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.

- Thiết kế được bộ quan sát trạng thái tối ưu cho hệ phi tuyến và điều kiện để bộ quan sát đó trở thành bộ quan sát có thời gian hữu hạn.

- Đưa ra điều kiện đủ để bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách, sử dụng bộ quan sát trạng thái tối ưu RHO là ổn định tiệm cận.

Tuy nhiên, ở đây vẫn có một số hạn chế:

- Chỉ áp dụng cho hệ song tuyến không liên tục.

- Cửa sổ dự báo ở đây được đẩy đến vô hạn dẫn đến không cho phép sử dụng các phương pháp quy hoạch phi tuyến để xử lý các điều kiện ràng buộc.

- Chỉ có thể áp dụng điều khiển ổn định, muốn điều khiển bám thì phải thêm phép chuyển đổi trung gian.

- Sử dụng bộ quan sát RHO để quan sát trạng thái nên bộ điều khiển khó có khả năng lọc nhiễu ồn trắng.

Trong tài liệu [41,4], tác giả đã đề xuất chuyển đổi phương pháp của tài liệu [1] sang cho hệ song tuyến liên tục, qua đó loại bỏ được sai số xấp xỉ trong quá trình điều khiển. Tuy nhiên vẫn gặp phải những hạn chế giống như tài liệu [1], đó là cửa sổ dự báo là vô hạn nên không cho phép sử dụng các phương pháp quy hoạch phi tuyến để xử lý điều kiện ràng buộc, không điều khiển được trực tiếp hệ bám theo quỹ đạo mẫu cho trước mà phải thông qua điều khiển ổn định, không xử lý được trực tiếp các điều kiện ràng buộc.

Ngoài ra, có một số tài liệu, luận án khác gần đây như [6,14,5] nghiên cứu về điều khiển dự báo, tuy nhiên các tài liệu này đều không đi sâu vào nghiên cứu điều khiển dự báo phản hồi đầu ra. Tài liệu [6] đề xuất phương pháp thiết kế bộ điều khiển dự báo dựa hệ phi tuyến dựa vào mô hình mờ. Luận án [14] đi sâu vào nghiên cứu 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 [5] đề xuất phương pháp nhận dạng nhiễu cho hệ thống có trễ trong điều khiển trên cơ sở mô hình song song, trong đó các mạng nơron RBF được sử dụng để xấp xỉ các hàm nhiễu phi tuyến bất định bởi mạng này có cấu trúc đơn giản và có khả năng xấp xỉ hầu hết các hàm phi tuyến với độ chính xác tùy ý.

1.4 Định hướng nghiên cứu của luận án

Thông qua việc tổng hợp các kết quả nghiên cứu về điều khiển dự báo phản hồi đầu ra trên thế giới và của các tác giả trong nước thời gian gần đây, có thể thấy được một số tồn tại, hạn chế cần được tiếp tục nghiên cứu về điều khiển dự báo phản hồi đầu ra, đó là:

- Cần phải có bộ quan sát trạng thái phù hợp cho các đối tượng phi tuyến, có khả năng giảm thiểu ảnh hưởng của nhiễu tác động, đặc biệt là các nhiễu ồn trắng, để đảm bảo có được một chất lượng điều khiển tốt.
- Cần thiết phải có một thuật toán điều khiển dự báo phản hồi trạng thái khác để cài đặt hơn so với Thuật toán 1.6 để mang lại khả năng cài đặt đơn giản song vẫn đảm bảo được chất lượng cho hệ điều khiển dự báo phản hồi đầu ra cho hệ phi tuyến.
- Bộ điều khiển dự báo phản hồi đầu ra trên cơ sở kết hợp giữa bộ quan sát trạng thái và bộ điều khiển dự báo phản hồi trạng thái cần phải được nghiên cứu và kiểm chứng chất lượng để có thể áp dụng trực tiếp vào thực tế đối với các đối tượng phi tuyến có các trạng thái là không đo trực tiếp được hoặc có thể đo nhưng với chi phí rất cao.
- Cần phải tổng quát hóa phương pháp thành các bước của thuật toán và chứng minh tính ổn định của hệ kín đồng thời kiểm chứng bằng mô phỏng và thực nghiệm.

Từ đây, luận án đưa ra định hướng nghiên cứu đó là:

- *Nghiên cứu phát triển bộ điều khiển dự báo phản hồi đầu ra mang tính khả dụng cao với những đối tượng phi tuyến.*
- *Sử dụng mô hình tuyến tính hóa từng đoạn để làm mô hình dự báo thay cho việc sử dụng trực tiếp mô hình phi tuyến trong thiết kế bộ điều khiển dự báo. Nhưng sử dụng của số dự báo hữu hạn để có thể sử dụng trực tiếp các phương pháp quy hoạch phi tuyến để xử lý điều kiện ràng buộc (nếu có).*
- *Sử dụng lọc Kalman mở rộng (EKF) để quan sát trạng thái, qua đó giảm thiểu ảnh hưởng của nhiễu ồn trắng.*

- *Thiết kế điều khiển bám trực tiếp theo tín hiệu đầu ra mẫu mà không cần chuyển về bài toán điều khiển ổn định.*
- *Ứng dụng vào điều khiển đối tượng có mô hình phi tuyến.*
- *Kiểm chứng chất lượng của bộ điều khiển bằng mô phỏng và thực nghiệm.*

Tính mới trong định hướng nghiên cứu của luận án so với các tác giả khác đó là:

- Sử dụng mô hình tuyến tính rời rạc hóa từng đoạn.
- Sử dụng cửa sổ dự báo hữu hạn: cho phép sử dụng trực tiếp các phương pháp quy hoạch phi tuyến để xử lý các điều kiện ràng buộc.
- Sử dụng bộ Lọc Kalman mở rộng (EKF) để quan sát trạng thái.
- Mở rộng phương pháp cho hệ phi tuyến tổng quát.
- Điều khiển bám trực tiếp đầu ra mẫu.
- Tổng quát hóa phương pháp điều khiển dự báo phản hồi đầu ra, đưa ra các thuật toán dễ cài đặt.
- Chứng minh tính ổn định của hệ kín bằng lý thuyết và kiểm chứng phương pháp bằng mô phỏng và thực nghiệm.

1.5 Kết luận chương 1

Trong chương I luận án đã trình bày tổng quan những phương pháp điều khiển dự báo phản hồi đầu ra, bao gồm cả những phương pháp phản hồi đầu ra trực tiếp đơn giản như MAC, DMC, GPC (Thuật toán 1.1 - Thuật toán 1.3), chủ yếu cho hệ SISO, cho đến phương pháp phản hồi đầu ra gián tiếp, hay còn gọi là phản hồi đầu ra theo nguyên lý tách, dùng được cho cả hệ MIMO.

Tất cả các phương pháp điều khiển đầu ra được trình bày trên đều chủ yếu là xây dựng cho hệ tuyến tính (Thuật toán 1.4), còn đối với hệ phi tuyến, nó mới chỉ dừng lại ở phương hướng. Lý do là vì thuật toán điều khiển dự báo phản hồi trạng thái (Thuật toán 1.6) rất khó được cài đặt và nếu như cài đặt được, sai số tính toán lớn của nó là không tránh khỏi bởi tính phi tuyến cao của hàm mục tiêu (1.36) tính theo tín hiệu điều khiển \underline{u} xác định theo các công thức (1.34) và (1.35).

Do mục tiêu của luận án là sử dụng lọc Kalman như một bộ quan trạng thái, phục vụ bài toán điều khiển dự báo hệ phi tuyến bằng phản hồi đầu ra theo nguyên lý tách, nên ở chương I, luận án cũng đã trình bày tóm tắt nội dung về bộ lọc Kalman tuyến tính (KF - Thuật toán 1.5) cũng như các dạng mở rộng của nó (EKF - Thuật toán 1.7 - Thuật toán 1.9) và UKF (Thuật toán 1.10) cho hệ phi tuyến. Mỗi dạng mở rộng này của lọc Kalman đều có những ưu nhược điểm riêng và cũng chỉ nên áp dụng cho một lớp các hệ phi tuyến đặc biệt.

Thông qua việc tổng hợp các kết quả nghiên cứu trong và ngoài nước, phân tích những điểm hạn chế cần tiếp tục được nghiên cứu, luận án cũng đã đưa ra định hướng nghiên cứu trong phần cuối chương.

Trên cơ sở kết quả phân tích về khả năng cài đặt ít thành công của Thuật toán 1.6 dùng cho điều khiển dự báo hệ phi tuyến, nên sau đây, trong chương 2, luận án sẽ đề xuất một phương pháp điều khiển dự báo phản hồi trạng thái khác mang tính khả thi cao hơn, dùng được cho hệ phi tuyến MIMO nói chung và hệ song tuyến nói riêng. Tất nhiên, cùng với phương pháp điều khiển dự báo phản hồi trạng thái phi tuyến được đề xuất mới này, bộ lọc Kalman mở rộng cũng sẽ được cải tiến một cách phù hợp hơn để có thể tương thích với phương pháp đó trong điều khiển phản hồi đầu ra theo nguyên lý tách.

CHƯƠNG 2 THIẾT KẾ LỌC KALMAN ĐỂ QUAN SÁT TỪNG ĐOẠN TRẠNG THÁI THEO NGUYÊN LÝ TỐI ƯU VÀ ỨNG DỤNG VÀO ĐIỀU KHIỂN DỰ BÁO PHẢN HỒI ĐẦU RA HỆ PHI TUYẾN THEO NGUYÊN LÝ TÁCH

Trong chương này luận án sẽ trình bày một phương pháp mở rộng lọc Kalman cho hệ phi tuyến trên cơ sở tuyến tính từng đoạn mô hình phi tuyến của hệ được quan sát, sau đó xác định xấp xỉ trạng thái của hệ trong đoạn đó theo một tiêu chuẩn tối ưu, sẽ được luận án gọi là *quan sát Kalman từng đoạn*. Nếu so sánh với những bộ lọc Kalman mở rộng đã có trước đây (EKF loại 1-3), thì bộ quan sát Kalman từng đoạn này sẽ phù hợp hơn khi sử dụng cùng với bộ điều khiển dự báo phản hồi trạng thái có mô hình dự báo cũng ở dạng tuyến tính từng đoạn (*optimal receding horizon control*). Đó cũng là bộ điều khiển phản hồi trạng thái được luận án sử dụng chính trong bộ điều khiển dự báo phản hồi đầu ra để điều khiển các đối tượng phi tuyến trong công nghiệp.

2.1 Xây dựng bộ quan sát Kalman từng đoạn cho hệ phi tuyến

2.1.1 Quan sát Kalman từng đoạn cho hệ song tuyến

Một hệ phi tuyến với mô hình:

$$\begin{cases} \underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{u}_k) + \underline{\xi}_k \\ \underline{y}_{k+1} = \underline{g}(\underline{x}_k, \underline{u}_k) + \underline{\zeta}_k \end{cases} \quad (2.1)$$

được gọi là song tuyến (bilinear), nếu hai vector hàm $\underline{f}(\underline{x}_k, \underline{u}_k)$, $\underline{g}(\underline{x}_k, \underline{u}_k)$ có tính chất [11]:

- khi \underline{x}_k là hằng số chúng sẽ tuyến tính theo \underline{u}_k và ngược lại
- khi \underline{u}_k là hằng số chúng sẽ tuyến tính theo \underline{x}_k .

Thực chất khái niệm tuyến tính ở trên không mang được tính chặt chẽ của một ánh xạ tuyến tính mà chỉ dừng lại là đồ thị của chúng chiếu theo hai trục tọa độ \underline{x}_k và

\underline{u}_k có dạng đường thẳng (có thể không đi qua gốc). Hơn nữa, trong nhiều tài liệu người ta còn quan niệm rằng hệ song tuyến là hệ mà mô hình của nó chỉ cần thỏa mãn một trong hai điều kiện ở trên là đủ, tức là hai vector hàm $\underline{f}(\underline{x}_k, \underline{u}_k)$, $\underline{g}(\underline{x}_k, \underline{u}_k)$ chỉ cần tuyến tính theo \underline{u}_k khi \underline{x}_k là hằng số, hoặc tuyến tính theo \underline{x}_k khi \underline{u}_k là hằng số.

Xuất phát điểm của phương pháp

Phương pháp đề xuất của luận án có xuất phát điểm là tuyến tính hóa từng đoạn hệ phi tuyến, sau đó xây dựng bộ lọc Kalman tuyến tính cho từng đoạn này trên cơ sở là giả sử đã biết giá trị trạng thái xấp xỉ ở đầu đoạn tuyến tính hóa đó. Khoảng thời gian tuyến tính hóa từng đoạn sẽ chính là khoảng thời gian trích mẫu của hệ phi tuyến không liên tục (2.1).

Để minh họa tư tưởng thiết kế bộ quan sát Kalman từng đoạn cho hệ song tuyến của luận án, trước tiên ta xét hệ song tuyến không dừng có dạng tuyến tính theo \underline{u}_k khi \underline{x}_k là hằng số như sau:

$$\begin{cases} \underline{x}_{k+1} = \underline{A}(\underline{x}_k, k)\underline{x}_k + \underline{B}(\underline{x}_k, k)\underline{u}_k \\ \underline{y}_k = \underline{C}(\underline{x}_k, k)\underline{x}_k + \underline{D}(\underline{x}_k, k)\underline{u}_k \end{cases} \quad (2.2)$$

trong đó $\underline{A}(\underline{x}_k, k)$, $\underline{B}(\underline{x}_k, k)$, $\underline{C}(\underline{x}_k, k)$, $\underline{D}(\underline{x}_k, k)$ được giả thiết là các ma trận hàm liên tục theo trạng thái \underline{x}_k và thời gian k , cũng như chu kỳ trích mẫu T_a là không quá lớn để có thể xấp xỉ được thành:

$$\begin{aligned} \underline{A}_k &= \underline{A}(\underline{\bar{x}}_k, k) \approx \underline{A}(\underline{x}_k, k), \quad \underline{B}_k = \underline{B}(\underline{\bar{x}}_k, k) \approx \underline{B}(\underline{x}_k, k) \\ \underline{C}_k &= \underline{C}(\underline{\bar{x}}_k, k) \approx \underline{C}(\underline{x}_k, k), \quad \underline{D}_k = \underline{D}(\underline{\bar{x}}_k, k) \approx \underline{D}(\underline{x}_k, k) \end{aligned} \quad (2.3)$$

với $\underline{\bar{x}}_k \approx \underline{x}_k$ là giá trị trạng thái đã quan sát được ở thời điểm k .

Như vậy, ở thời điểm hiện tại $t_k = kT_a$ và trong một khoảng thời gian ngắn ngay sau đó $kT_a \leq t < (k+1)T_a$ hệ song tuyến ban đầu (2.2) sẽ được xấp xỉ bằng mô hình tuyến tính tham số hằng (mô hình LTI):

$$\begin{cases} \underline{x}_{k+1} = \underline{A}_k \underline{x}_k + \underline{B}_k \underline{u}_k + \underline{\xi}_k \\ \underline{y}_k = \underline{C}_k \underline{x}_k + \underline{D}_k \underline{u}_k + \underline{\zeta}_k \end{cases} \quad (2.4)$$

Tất nhiên hệ LTI (2.4) này chỉ có ý nghĩa xấp xỉ trong khoảng thời gian của một chu kỳ tricha mẫu $kT_a \leq t < (k+1)T_a$. Ở khoảng thời gian tiếp theo, ta sẽ có một mô hình LTI khác với các ma trận hằng $A_{k+1}, B_{k+1}, C_{k+1}, D_{k+1}$ khác, được tính theo (2.3) nhưng với giá trị trạng thái mới bây giờ là $\underline{\bar{x}}_{k+1}$.

Cùng với mô hình tuyến tính từng đoạn (2.4) của hệ song tuyến (2.2), cũng như cùng với các giả thiết về nhiễu $\underline{\xi}_k, \underline{\zeta}_k$ đã cho ở (1.25), thuật toán KF (Thuật toán 1.5) cũng sẽ áp dụng được cho hệ (2.4) trong khoảng thời gian ngắn $kT_a \leq t < (k+1)T_a$ để xác định xấp xỉ $\underline{\bar{x}}_k$ trên cơ sở đã có $\underline{\bar{x}}_{k-1} \approx \underline{x}_{k-1}$ trước đó, với một sửa đổi nhỏ là ở ngay đầu các vòng lặp ta cần phải xác định lại bốn ma trận $A_{k-1}, B_{k-1}, C_k, D_k$ theo công thức xấp xỉ (2.3).

Ngoài ra, có thể thấy ngay được rằng để áp dụng thành công thuật toán KF (Thuật toán 1.5) một cách chính xác, ta cần phải có giá trị trạng thái xấp xỉ của hệ ở thời điểm đầu của đoạn đó, được ký hiệu là $\underline{\bar{x}}_{k-1} \approx \underline{x}_{k-1}$. Tuy nhiên, từ (2.3) ta còn thấy ngay được rằng tính chính xác của kết quả thu được phụ thuộc rất nhiều vào độ chính xác của công thức xấp xỉ này, nhất là ở ngay bước đầu tiên. Bởi vậy để có được $\underline{\bar{x}}_0 = \underline{x}_0(+)$ đủ chính xác, ta cần phải có một thuật toán con phục vụ cho việc đó trước khi sử dụng KF (Thuật toán 1.5), thay vì chọn nó một cách vô định như đã làm ở Thuật toán 1.5 trước đây. Nội dung của thuật toán con này cũng như độ phức tạp của nó phụ thuộc vào độ phi tuyến của hệ (2.1) đã cho. Bởi vậy, để đơn giản, trước tiên luận án sẽ xây dựng thuật toán con đó cho trường hợp hệ phi tuyến (2.1) có cấu trúc gần tuyến tính nhất và đó là hệ song tuyến (bilinear), sau đó mới mở rộng cho hệ phi tuyến bất kỳ.

Xây dựng bộ quan sát Kalman từng đoạn cho hệ song tuyến

Để thuận tiện cho việc xây dựng thuật toán xác định $\underline{\bar{x}}_0 = \underline{x}_0(+)$, trước tiên ta sẽ xét trường hợp đơn giản là hệ (2.2) còn có thể chuyển đổi được về dạng:

$$\begin{cases} \underline{x}_{k+1} = A(\underline{u}_k, k)\underline{x}_k + B(\underline{u}_k, k)\underline{u}_k + \underline{\xi}_k \\ \underline{y}_k = C(\underline{u}_k, k)\underline{x}_k + D(\underline{u}_k, k)\underline{u}_k + \underline{\zeta}_k \end{cases} \quad (2.5)$$

Để có \underline{x}_0 tất nhiên cần phải có một khoảng thời gian quan sát. Nếu ký hiệu khoảng thời gian quan sát này là $i = 0, 1, \dots, M$ thì giá trị trạng thái quan sát được là \underline{x}_M sẽ được xem như giá trị khởi phát \underline{x}_0 cho Thuật toán 1.5.

Sau đây ta sẽ xác định \underline{x}_M cho hệ (2.5) theo tiêu chuẩn tối ưu là sai lệch quan sát trong một cửa sổ quá khứ của nó là $i = 0, 1, \dots, M$ đạt giá trị nhỏ nhất. Thực ra, nội dung quan sát trạng thái \underline{x}_k theo tiêu chuẩn tối ưu tổng quát cho hệ (2.5) đã được xây dựng ở tài liệu [1], song thuật toán này lại chưa có khả năng lọc nhiễu. Do đó, đối với hệ có nhiễu $\underline{\xi}_k, \underline{\zeta}_k$ tác động, ta sẽ kết hợp lọc Kalman và thuật toán quan sát tối ưu đó của tài liệu [1] bằng cách chỉ sử dụng nguyên tắc quan sát tối ưu này của [1] một lần để xác định \underline{x}_M từ các giá trị vào ra đo được trong quá khứ $\underline{u}_i, \underline{y}_i, i = 0, 1, \dots, M-1$, rồi sử dụng \underline{x}_M làm giá trị khởi phát \underline{x}_0 cho việc áp dụng Thuật toán 1.5 vào quan sát trạng thái hệ (2.5) sau này.

Để tiện cho việc trình bày, sau đây ta sẽ sử dụng các ký hiệu:

$$\begin{aligned} \hat{A}_i &= A(\underline{u}_i, i), \quad \hat{B}_i = B(\underline{u}_i, i), \\ \text{và} \quad \hat{C}_i &= C(\underline{u}_i, i), \quad \hat{D}_i = D(\underline{u}_i, i), \quad i = 0, 1, \dots, M \end{aligned} \quad (2.6)$$

Do $\underline{u}_i, i = 0, 1, \dots, M$ là đã biết nên tất cả các ma trận $\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i$ tính theo công thức (2.6) ở trên với $i = 0, 1, \dots, M$ cũng đã biết. Từ đây ta suy ra được sự phụ thuộc của các biến trạng thái $\underline{x}_i, i = 1, 2, \dots, M$ vào trạng thái đầu \underline{x}_0 cần xác định và các giá trị tín hiệu đầu vào đo được $\underline{u}_i, i = 0, 1, \dots, M$ như sau:

$$\begin{aligned} \underline{x}_1 &= \hat{A}_0 \underline{x}_0 + \hat{B}_0 \underline{u}_0 = \hat{A}_0 \underline{x}_0 + \underline{d}_0 \\ \underline{x}_2 &= \hat{A}_1 \underline{x}_1 + \hat{B}_1 \underline{u}_1 = \hat{A}_1 (\hat{A}_0 \underline{x}_0 + \underline{d}_0) + \hat{B}_1 \underline{u}_1 = \hat{A}_1 \hat{A}_0 \underline{x}_0 + \hat{A}_1 \underline{d}_0 + \hat{B}_1 \underline{u}_1 \\ &= \hat{A}_1 \hat{A}_0 \underline{x}_0 + \underline{d}_1 \\ &\vdots \\ \underline{x}_i &= \hat{A}_{i-1} \underline{x}_{i-1} + \hat{B}_{i-1} \underline{u}_{i-1} = \hat{A}_{i-1} (\hat{A}_{i-2} \hat{A}_{i-3} \cdots \hat{A}_0 \underline{x}_0 + \underline{d}_{i-2}) + \hat{B}_{i-1} \underline{u}_{i-1} \\ &= \hat{A}_{i-1} \hat{A}_{i-2} \hat{A}_{i-3} \cdots \hat{A}_0 \underline{x}_0 + \underline{d}_{i-1} \\ &\vdots \\ \underline{x}_M &= \hat{A}_{M-1} \hat{A}_{M-2} \cdots \hat{A}_0 \underline{x}_0 + \underline{d}_{M-1} \end{aligned}$$

trong đó:

$$\begin{aligned}
 \underline{d}_0 &= \widehat{B}_0 \underline{u}_0 \\
 \underline{d}_1 &= \widehat{A}_1 \underline{d}_0 + \widehat{B}_1 \underline{u}_1 \\
 &\vdots \\
 \underline{d}_{i-1} &= \widehat{A}_{i-1} \underline{d}_{i-2} + \widehat{B}_{i-1} \underline{u}_{i-1} \\
 &\vdots \\
 \underline{d}_{M-1} &= \widehat{A}_{M-1} \underline{d}_{M-2} + \widehat{B}_{M-1} \underline{u}_{M-1}
 \end{aligned} \tag{2.7}$$

đều là những vector đã biết, tức là xác định được từ những giá trị vào ra đo được trước đó.

Vậy các sai lệch quan sát trong cửa sổ quan sát $i = 0, 1, \dots, M$ sẽ là:

$$\begin{aligned}
 e_1 &= \underline{y}_1 - (\widehat{C}_1 \underline{x}_1 + \widehat{D}_1 \underline{u}_1) = \underline{y}_1 - [\widehat{C}_1 (\widehat{A}_0 \underline{x}_0 + \underline{d}_0) + \widehat{D}_1 \underline{u}_1] \\
 &= \underline{y}_1 - \widehat{C}_1 \underline{d}_0 - \widehat{D}_1 \underline{u}_1 - \widehat{C}_1 \widehat{A}_0 \underline{x}_0 = \underline{g}_1 - \widehat{C}_1 \widehat{A}_0 \underline{x}_0 \\
 e_2 &= \underline{y}_2 - (\widehat{C}_2 \underline{x}_2 + \widehat{D}_2 \underline{u}_2) = \underline{y}_2 - [\widehat{C}_2 (\widehat{A}_1 \widehat{A}_0 \underline{x}_0 + \underline{d}_1) + \widehat{D}_2 \underline{u}_2] \\
 &= \underline{y}_2 - \widehat{C}_2 \underline{d}_1 - \widehat{D}_2 \underline{u}_2 - \widehat{C}_2 \widehat{A}_1 \widehat{A}_0 \underline{x}_0 = \underline{g}_2 - \widehat{C}_2 \widehat{A}_1 \widehat{A}_0 \underline{x}_0 \\
 &\vdots \\
 e_M &= \underline{y}_M - (\widehat{C}_M \underline{x}_M + \widehat{D}_M \underline{u}_M) = \underline{y}_M - [\widehat{C}_M (\widehat{A}_{M-1} \widehat{A}_{M-2} \cdots \widehat{A}_0 \underline{x}_0 + \underline{d}_{M-1}) + \widehat{D}_M \underline{u}_M] \\
 &= \underline{y}_M - \widehat{C}_M \underline{d}_{M-1} - \widehat{D}_M \underline{u}_M - \widehat{C}_M \widehat{A}_{M-1} \widehat{A}_{M-2} \cdots \widehat{A}_0 \underline{x}_0 = \underline{g}_M - \widehat{C}_M \widehat{A}_{M-1} \widehat{A}_{M-2} \cdots \widehat{A}_0 \underline{x}_0
 \end{aligned}$$

với các vector hằng \underline{g}_i , $i = 1, 2, \dots, M$ đã biết, vì chúng được xác định được từ những giá trị cũng đã biết \underline{u}_i , $i = 0, 1, \dots, M$, như sau:

$$\begin{aligned}
 \underline{g}_1 &= \underline{y}_1 - \widehat{C}_1 \underline{d}_0 - \widehat{D}_1 \underline{u}_1 \\
 \underline{g}_2 &= \underline{y}_2 - \widehat{C}_2 \underline{d}_1 - \widehat{D}_2 \underline{u}_2 \\
 &\vdots \\
 \underline{g}_M &= \underline{y}_M - \widehat{C}_M \underline{d}_{M-1} - \widehat{D}_M \underline{u}_M
 \end{aligned} \tag{2.8}$$

Viết chung tất cả sai lệch quan sát e_i , $i = 1, 2, \dots, M$ lại thành vector hợp:

$$\begin{aligned}\underline{\mathbf{e}} &= \begin{pmatrix} \underline{e}_1 \\ \underline{e}_2 \\ \vdots \\ \underline{e}_M \end{pmatrix} = \begin{pmatrix} \underline{g}_1 \\ \underline{g}_2 \\ \vdots \\ \underline{g}_M \end{pmatrix} - \begin{pmatrix} \hat{C}_1 \hat{A}_0 \\ \hat{C}_2 \hat{A}_1 \hat{A}_0 \\ \vdots \\ \hat{C}_M \hat{A}_{M-1} \hat{A}_{M-2} \cdots \hat{A}_0 \end{pmatrix} \underline{x}_0 \\ &= \underline{\mathbf{g}} - G \underline{x}_0\end{aligned}$$

trong đó:

$$\underline{\mathbf{g}} = \begin{pmatrix} \underline{g}_1 \\ \underline{g}_2 \\ \vdots \\ \underline{g}_M \end{pmatrix}, \quad G = \begin{pmatrix} \hat{C}_1 \hat{A}_0 \\ \hat{C}_2 \hat{A}_1 \hat{A}_0 \\ \vdots \\ \hat{C}_M \hat{A}_{M-1} \hat{A}_{M-2} \cdots \hat{A}_0 \end{pmatrix} \quad (2.9)$$

ta sẽ có tổng bình phương các sai lệch quan sát trong toàn bộ cửa sổ quan sát, dưới dạng hàm số của \underline{x}_0 , là:

$$\begin{aligned}J(\underline{x}_0) &= \underline{\mathbf{e}}^T \Lambda \underline{\mathbf{e}} = (\underline{\mathbf{g}} - G \underline{x}_0)^T \Lambda (\underline{\mathbf{g}} - G \underline{x}_0) \\ &= \underline{\mathbf{g}}^T \Lambda \underline{\mathbf{g}} - 2 \underline{\mathbf{g}}^T \Lambda G \underline{x}_0 + \underline{x}_0^T G^T \Lambda G \underline{x}_0\end{aligned} \quad (2.10)$$

trong đó Λ là ma trận trọng số. Ta có thể chọn ma trận trọng số này có dạng đường chéo:

$$\Lambda = \text{diag}(\lambda_i), \quad \lambda_i > 0 \quad (2.11)$$

Khi đó, để ưu tiên một sai lệch quan sát \underline{e}_i nào đó trong số M các sai lệch quan sát \underline{e}_i , $i = 1, 2, \dots, M$ ta sẽ chọn một hằng số λ_i tương ứng lớn hơn các hằng số còn lại.

Nhiệm vụ tiếp theo sẽ là xác định \underline{x}_0 sao cho tổng bình phương các sai lệch quan sát (2.10) là nhỏ nhất. Lúc này, do hàm $J(\underline{x}_0)$ có dạng toàn phương, nên ta có ngay được nghiệm tối ưu:

$$\underline{x}_0^* = \arg \min_{\underline{x}_0} J(\underline{x}_0)$$

như sau [12]:

$$\underline{x}_0^* = (G^T \Lambda G)^{-1} G^T \Lambda \underline{\mathbf{g}}$$

và cùng với nó là:

$$\begin{aligned}\underline{x}_M^* &= \widehat{A}_{M-1}\widehat{A}_{M-2}\cdots\widehat{A}_0\underline{x}_0^* + \underline{d}_{M-1} \\ &= \widehat{A}_{M-1}\widehat{A}_{M-2}\cdots\widehat{A}_0(G^T \Lambda G)^{-1}G^T \Lambda \underline{g} + \underline{d}_{M-1}\end{aligned}\quad (2.12)$$

Sau khi đã có được giá trị trạng thái \underline{x}_M^* theo công thức (2.12), ta sẽ sử dụng nó như giá trị trạng thái xuất phát $\underline{x}_0 = \underline{x}_0(+)$ trong Thuật toán 1.5 để quan sát trạng thái tối ưu cho hệ (2.5).

Toàn bộ nội dung thuật toán này sẽ như sau:

Thuật toán 2.1: Quan sát Kalman từng đoạn hệ song tuyến (2.5).

- 1) Chọn cửa sổ quan sát khởi phát $M \geq 2$ và ma trận trọng số Λ theo (2.11).
- 2) Đo các giá trị vào ra $\underline{u}_i, \underline{y}_i, i = 0, 1, \dots, M$.
 - a) Xác định tất cả các ma trận $\widehat{A}_i, \widehat{B}_i, \widehat{C}_i, \widehat{D}_i, i = 0, 1, \dots, M$ từ mô hình (2.5) của hệ theo công thức (2.6).
 - b) Tính M các vector $\underline{d}_i, i = 0, 1, \dots, M-1$ theo (2.7) và $\underline{g}_i, i = 1, 2, \dots, M$ theo (2.8). Xây dựng vector hợp \underline{g} và ma trận G theo (2.9).
 - c) Tính \underline{x}_M^* theo (2.12). Gán $\bar{\underline{x}}_M = \underline{x}_M^*, A_M = \widehat{A}_M, B_M = \widehat{B}_M$ và xuất $\bar{\underline{x}}_M$ làm giá trị trạng thái quan sát được của hệ (2.5) ở thời điểm M .
- 3) Gán $\underline{x}_M(+)=\bar{\underline{x}}_M$ và chọn $P_M(+)$ tùy ý. Gán $k = M + 1$.
- 4) Đo $\underline{u}_k, \underline{y}_k$. Xác định $C_k = C(\underline{u}_k, k), D_k = D(\underline{u}_k, k)$.
- 5) Tính $\underline{x}_k(-) = A_{k-1}\underline{x}_{k-1}(+) + B_{k-1}\underline{u}_{k-1}$

$$P_k(-) = A_{k-1}P_{k-1}(+)A_{k-1}^T + \Psi_{k-1}$$

$$K_k = P_k(-)C_k^T(C_kP_k(-)C_k^T + \Phi_k)^{-1}$$

$$P_k(+) = (I - K_kC_k)P_k(-)$$

$$\underline{x}_k(+) = \underline{x}_k(-) + K_k(\underline{y}_k - C_k\underline{x}_k(-) - D_k\underline{u}_k)$$
- 6) Xuất $\bar{\underline{x}}_k = \underline{x}_k(+)$ làm giá trị trạng thái quan sát được của hệ (2.5) ở thời điểm k .

Tính $A_k = A(\underline{u}_k, k), B_k = B(\underline{u}_k, k)$. Gán $k := k + 1$ và quay về 4).

Nếu so sánh với bộ quan sát trạng thái đã được giới thiệu ở tài liệu [1] mà ở đó các giá trị trạng thái tiếp theo $\bar{\underline{x}}_k, k = M + 1, M + 1, \dots$ cũng được xác định lặp giống

như \underline{x}_M , thì ở bộ quan sát Kalman trên (Thuật toán 2.1), chúng lại được xác định theo KF (Thuật toán 1.5). Nhờ đó khả năng kháng nhiễu $\underline{\xi}_k, \underline{\zeta}_k$ của nó chắc chắn sẽ tốt hơn.

Hơn nữa, do thuật toán quan sát Kalman từng đoạn mà luận án đề xuất ở trên (Thuật toán 2.1) được xây dựng cho hệ song tuyến không dừng (2.5) nên tất nhiên nó cũng áp dụng được cho cả hệ song tuyến dừng, mô tả bởi:

$$\begin{cases} \underline{x}_{k+1} = A(\underline{u}_k)\underline{x}_k + B(\underline{u}_k)\underline{u}_k + \underline{\xi}_k \\ \underline{y}_k = C(\underline{u}_k)\underline{x}_k + D(\underline{u}_k)\underline{x}_k + \underline{\zeta}_k \end{cases}$$

trong đó, các ma trận $\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i, i=0,1, \dots, M$ thay vì được xác định theo công thức (2.6), bây giờ sẽ là:

$$\begin{aligned} \hat{A}_i &= A(\underline{u}_i), \hat{B}_i = B(\underline{u}_i), \\ \hat{C}_i &= C(\underline{u}_i), \hat{D}_i = D(\underline{u}_i), i=0,1, \dots, M \end{aligned}$$

Cuối cùng, ta có thể nhận thấy thêm được rằng Thuật toán 2.1 do luận án đề xuất trên đây còn áp dụng được cho cả những hệ song tuyến tổng quát:

$$\begin{cases} \underline{x}_{k+1} = A(\underline{x}_{k-1}, \underline{u}_k, k)\underline{x}_k + B(\underline{x}_{k-1}, \underline{u}_k, k)\underline{u}_k + \underline{\xi}_k \\ \underline{y}_k = C(\underline{x}_{k-1}, \underline{u}_k, k)\underline{x}_k + D(\underline{x}_{k-1}, \underline{u}_k, k)\underline{x}_k + \underline{\zeta}_k \end{cases} \quad (2.13)$$

có các ma trận $A(\underline{x}_{k-1}, \underline{u}_k, k), B(\underline{x}_{k-1}, \underline{u}_k, k), C(\underline{x}_{k-1}, \underline{u}_k, k), D(\underline{x}_{k-1}, \underline{u}_k, k)$ liên tục theo \underline{x}_{k-1} , vì khi đó các ma trận $\hat{A}_i, \hat{B}_i, \hat{C}_i, \hat{D}_i, i=0,1, \dots, M$ ở bước 2 của thuật toán, thay vì được xác định theo công thức (2.6), bây giờ sẽ được xác định xấp xỉ bởi:

$$\begin{aligned} \hat{A}_i &\approx A(\bar{x}_{i-1}, \underline{u}_i, i), \hat{B}_i \approx B(\bar{x}_{i-1}, \underline{u}_i, i) \\ \hat{C}_i &\approx C(\bar{x}_{i-1}, \underline{u}_i, i), \hat{D}_i \approx D(\bar{x}_{i-1}, \underline{u}_i, i) \end{aligned}$$

Ví dụ 2.1: Minh họa bộ quan sát Kalman từng đoạn

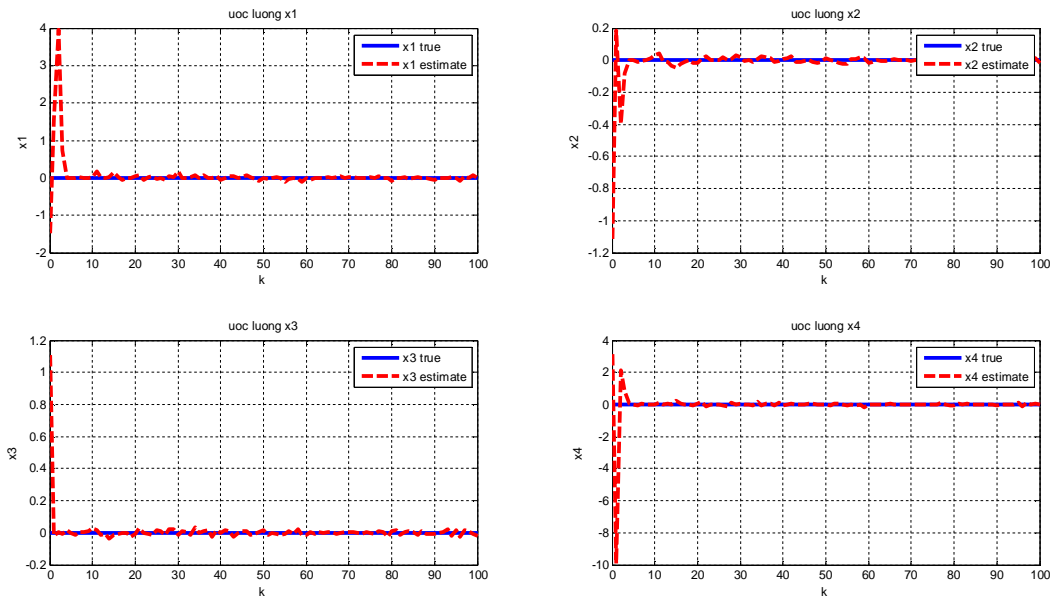
Xét đối tượng có mô hình song tuyến:

$$\begin{cases} \underline{x}_{k+1} = \begin{pmatrix} 0,5 & u_k & 0,5 & -1 \\ 0 & 1 & 1 & -2 \\ 0 & 0,5 & 1 & -3 \\ 1 & 0,5 & u_k & -5 \end{pmatrix} \underline{x}_k \\ y_k = [0 \ 0 \ 0,1 \ 0] \underline{x}_k \end{cases} \quad (2.14)$$

trong đó $\underline{x}_k = (x_1[k] \ x_2[k] \ x_3[k] \ x_4[k])^T$ là véc tơ trạng thái của hệ, u_k là tín hiệu đầu vào điều khiển, y_k là tín hiệu đầu ra. Nếu so sánh với mô hình (2.5) thì:

$$A(u_k, k) = \begin{pmatrix} 0,5 & u_k & 0,5 & -1 \\ 0 & 1 & 1 & -2 \\ 0 & 0,5 & 1 & -3 \\ 1 & 0,5 & u_k & -5 \end{pmatrix}, \quad B(u_k, k) = 0 \quad \text{và} \quad C(u_k, k) = (0 \ 0 \ 0,1 \ 0)$$

Chọn cửa sổ quan sát $M = 10$, ma trận trọng số $\Lambda = 0,05 * \text{eye}(M)$. Tiếp tục thực hiện theo các bước của Thuật toán 2.1 với tín hiệu đầu vào $u_k = 5 \sin(kT_a)$ ta sẽ tính được giá trị $\hat{\underline{x}}_M$ là giá trị quan sát được tại thời điểm M . Tiếp tục gán $\underline{x}_{M(+)} = \hat{\underline{x}}_M$ và chọn $P_{M(+)} = 1$, gán $k = M + 1$, thực hiện các bước tiếp theo của thuật toán. Kết quả so sánh giữa các trạng thái quan sát được và trạng thái thực của hệ sử dụng bộ quan sát Kalman từng đoạn theo Thuật toán 2.1 được thể hiện trên Hình 2.1. Ở Hình 2.1 này, đường nét liền biểu diễn các trạng thái thực của hệ, đường nét đứt biểu diễn trạng thái quan sát được bằng bộ quan sát Kalman từng đoạn. Nhiễu đầu vào và nhiễu đầu ra có kỳ vọng bằng không. Quan sát đáp ứng cho thấy, trạng thái quan sát được bám rất sát với trạng thái thực của hệ dưới tác động của nhiễu chứng tỏ bộ quan sát Kalman từng đoạn làm việc tốt.



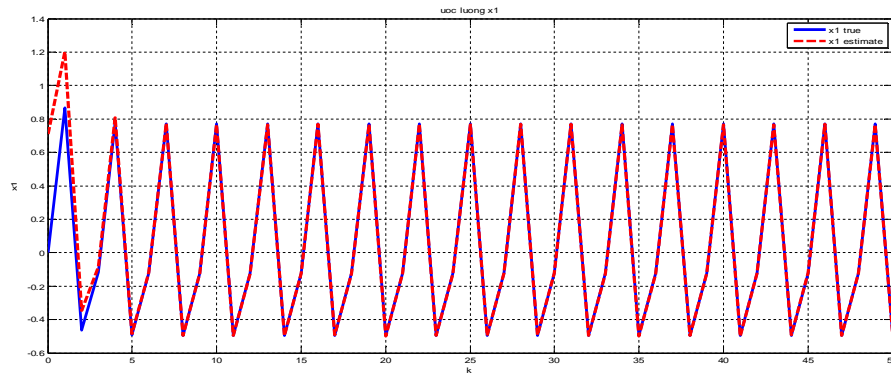
Hình 2.1: *Trạng thái quan sát được và trạng thái thực của đối tượng khi có nhiễu đầu vào và nhiễu đầu ra có giá trị kỳ vọng $\mu = 0$*

Ví dụ 2.2: *Minh họa bộ quan sát Kalman từng đoạn*

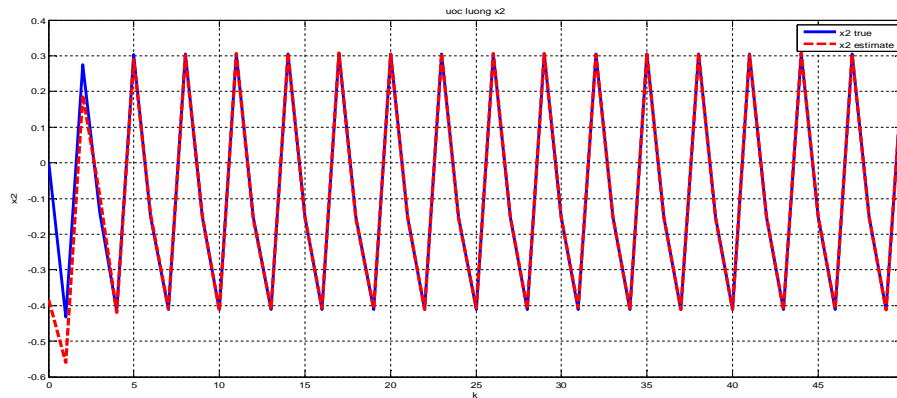
Tương tự, ta có thể xét đối tượng song tuyến:

$$\begin{cases} \mathbf{x}_{k+1} = \begin{pmatrix} 0,25 & u_k \\ u_k & -0,5 \end{pmatrix} \mathbf{x}_k + \begin{pmatrix} 2 \\ -1 \end{pmatrix} u_k \\ y_k = [1 \quad 1] \mathbf{x}_k \end{cases} \quad (2.15)$$

Chọn cửa sổ quan sát $M = 10$, ma trận trọng số $\Lambda = 0.05 * I(M)$, nhiễu đầu vào và nhiễu đầu ra là nhiễu Gauss có kỳ vọng bằng 0. Tiếp tục thực hiện các bước của Thuật toán 2.1, ta có được kết quả quan sát trạng thái của hệ như mô tả ở các Hình 2.2 và Hình 2.3. Kết quả mô phỏng cho thấy, sau một vài bước tính đầu tiên thì trạng thái quan sát được đã bám sát theo giá trị trạng thái thực của đối tượng và sau đó là hoàn toàn trùng khớp với các giá trị trạng thái thực này trong quá trình quan sát. Các kết quả mô phỏng đã khẳng định thêm tính đúng đắn của thuật toán quan sát Kalman từng đoạn được luận án đề xuất.



Hình 2.2: Biến trạng thái $x_1[k]$ khi có nhiễu đầu vào và nhiễu đầu ra có giá trị kỳ vọng $\mu = 0$



Hình 2.3: Biến trạng thái $x_2[k]$ khi có nhiễu đầu vào và nhiễu đầu ra có giá trị kỳ vọng $\mu = 0$

2.1.2 Thiết kế bộ quan sát Kalman từng đoạn cho hệ phi tuyến

Quan sát trạng thái hệ phi tuyến khi đã biết trạng thái đầu

Cũng từ tư tưởng chung đã được trình bày, sau đây ta sẽ xây dựng bộ quan sát Kalman từng đoạn cho hệ phi tuyến tổng quát mô tả bởi mô hình (2.1), trong đó các vector hàm $\underline{f}(x_k, u_k)$, $\underline{g}(x_k, u_k)$ được giả thiết là trơn theo cả x_k và u_k . Các tín hiệu nhiễu $\underline{\xi}_k$, $\underline{\zeta}_k$ cũng được giả thiết là nhiễu Gauss, phân bố chuẩn, có kỳ vọng bằng 0, và thỏa mãn (1.25).

Ta phân tích vector hàm $\underline{f}(\underline{x}_k, \underline{u}_k)$ thành chuỗi Taylor tại điểm quá khứ $\underline{\bar{x}}_k, \underline{u}_{k-1}$ như sau:

$$\begin{aligned}\underline{f}(\underline{x}_k, \underline{u}_k) &\approx \underline{f}(\underline{\bar{x}}_k, \underline{u}_{k-1}) + \underline{A}_k (\underline{x}_k - \underline{\bar{x}}_k) + \underline{B}_k (\underline{u}_k - \underline{u}_{k-1}) + \underline{g}_k \\ &= \underline{A}_k \underline{e}_k + \underline{B}_k \underline{v}_k + \underline{f}(\underline{\bar{x}}_k, \underline{u}_{k-1}) + \underline{g}_k = \underline{A}_k \underline{e}_k + \underline{B}_k \underline{v}_k + \underline{\bar{x}}_k + \underline{g}_k\end{aligned}\quad (2.16)$$

trong đó:

$$\underline{A}_k = \left. \frac{\partial \underline{f}}{\partial \underline{x}} \right|_{\underline{\bar{x}}_k, \underline{u}_{k-1}}, \underline{B}_k = \left. \frac{\partial \underline{f}}{\partial \underline{u}} \right|_{\underline{\bar{x}}_k, \underline{u}_{k-1}} \quad \text{và} \quad \underline{e}_k = \underline{x}_k - \underline{\bar{x}}_k, \underline{v}_k = \underline{u}_k - \underline{u}_{k-1}$$

và \underline{g}_k là sai số sinh ra từ việc bỏ qua các thành phần đạo hàm bậc cao trong chuỗi Taylor. Hiển nhiên \underline{v}_k là đại lượng đã biết, vì ở thời điểm k hiện tại ta đã có các giá trị đầu vào đo được $\underline{u}_{k-1}, \underline{u}_k$. Ngược lại \underline{g}_k là một đại lượng bất định.

Thay (2.16) vào phương trình thứ nhất trong mô hình phi tuyến ban đầu (2.1) của hệ, ta sẽ được:

$$\underline{e}_{k+1} = \underline{A}_k \underline{e}_k + \underline{B}_k \underline{v}_k + \underline{\xi}_{\underline{x}_k} + \underline{g}_k = \underline{A}_k \underline{e}_k + \underline{B}_k \underline{v}_k + \underline{v}_k \quad (2.17)$$

trong đó $\underline{e}_{k+1} = \underline{x}_{k+1} - \underline{\bar{x}}_k; \underline{v}_k = \underline{\xi}_{\underline{x}_k} + \underline{g}_k$.

Hoàn toàn tương tự với phương trình thứ hai của (2.1) ta có:

$$\begin{aligned}\underline{g}(\underline{x}_k, \underline{u}_k) &\approx \underline{g}(\underline{\bar{x}}_k, \underline{u}_{k-1}) + \underline{C}_k (\underline{x}_k - \underline{\bar{x}}_k) + \underline{D}_k (\underline{u}_k - \underline{u}_{k-1}) + \underline{\gamma}_k \\ &= \underline{C}_k \underline{e}_k + \underline{D}_k \underline{v}_k + \underline{\hat{y}}_{k-1} + \underline{\gamma}_k\end{aligned}\quad (2.18)$$

với:

$$\underline{C}_k = \left. \frac{\partial \underline{g}}{\partial \underline{x}} \right|_{\underline{\bar{x}}_k, \underline{u}_{k-1}}, \underline{D}_k = \left. \frac{\partial \underline{g}}{\partial \underline{u}} \right|_{\underline{\bar{x}}_k, \underline{u}_{k-1}} \quad \text{và} \quad \underline{\hat{y}}_{k-1} = \underline{g}(\underline{\bar{x}}_k, \underline{u}_{k-1})$$

là những đại lượng xác định được từ các giá trị đã có ở thời điểm trước đó là $k-1$. $\underline{\gamma}_k$ là sai số sinh ra từ việc bỏ qua các thành phần đạo hàm bậc cao trong chuỗi Taylor. Suy ra:

$$\Delta \underline{y}_{\underline{x}_k} = \underline{y}_k - \underline{\hat{y}}_{k-1} = \underline{C}_k \underline{e}_k + \underline{D}_k \underline{v}_k + \underline{\xi}_{\underline{x}_k} + \underline{\gamma}_k = \underline{C}_k \underline{e}_k + \underline{D}_k \underline{v}_k + \underline{\mu}_k \quad (2.19)$$

Ghép chung (2.17) và (2.19) lại với nhau ta được mô hình xấp xỉ tuyến tính của (2.1) trong đoạn $(k-1)T_a \leq t < (k+1)T_a$ như sau:

$$\begin{cases} \underline{e}_{k+1} = \underline{A}_k \underline{e}_k + \underline{B}_k \underline{v}_k + \underline{u}_k \\ \Delta \underline{y}_k = \underline{y}_k - \underline{\hat{y}}_{k-1} = \underline{C}_k \underline{e}_k + \underline{D}_k \underline{v}_k + \underline{\mu}_k \end{cases} \quad (2.20)$$

Trong mô hình tuyến tính trên, hai thành phần bất định $\underline{v}_k, \underline{\mu}_k$ là tổng của nhiều Gauss $\underline{\xi}_k, \underline{\zeta}_k$ (nhiều phân bố chuẩn, có kỳ vọng bằng 0) với sai lệch xấp xỉ $\underline{g}_k, \underline{\gamma}_k$ sinh ra bởi đã bỏ qua các thành phần bậc cao trong chuỗi Taylor. Do các sai lệch $\underline{g}_k, \underline{\gamma}_k$ này là tiên định nên tổng $\underline{v}_k, \underline{\mu}_k$ của chúng cũng là nhiều phân bố chuẩn, song có thể có kỳ vọng khác 0. Bởi vậy khi có giả thiết thêm rằng kỳ vọng của $\underline{g}_k, \underline{\gamma}_k$ là rất nhỏ để có thể bỏ qua được thì với mô hình tuyến tính (2.20) này, giá trị trạng thái $\underline{\hat{e}}_k \approx \underline{e}_k$ hoàn toàn được xác định từ tín hiệu đầu ra $\Delta \underline{y}_k$ và giá trị trạng thái đầu \underline{e}_{k-1} nhờ thuật toán KF (Thuật toán 1.5) của hệ tuyến tính. Giả thiết bổ sung rằng kỳ vọng của $\underline{g}_k, \underline{\gamma}_k$ rất nhỏ là chấp nhận được, vì các phân tích Taylor luôn được thực hiện trong lân cận nhỏ của $\underline{\hat{x}}_k^-, \underline{u}_{k-1}$.

Vậy khi đã biết trạng thái đầu $\underline{\hat{x}}_0 = \underline{x}_0$ của hệ, ta sẽ có được các giá trị trạng thái $\underline{x}_k, k = 1, 2, \dots$ tiếp theo của hệ (2.1) nhờ thuật toán sau:

Thuật toán 2.2a: Xác định trạng thái hệ phi tuyến khi đã có trạng thái đầu \underline{x}_0 .

1) Đo $\underline{u}_0, \underline{y}_0$. Gán $\underline{\hat{x}}_0 = \underline{x}_0, \underline{\hat{y}}_0 = \underline{y}_0, \underline{e}_0(+) = \underline{0}$ và $k = 1$. Tùy chọn $P_0(+)$.

2) Đo $\underline{u}_{k-1}, \underline{u}_k, \underline{y}_k$. Tính:

$$\underline{A}_k = \left. \frac{\partial f}{\partial \underline{x}} \right|_{\underline{\hat{x}}_{k-1}, \underline{u}_{k-1}}, \underline{B}_k = \left. \frac{\partial f}{\partial \underline{u}} \right|_{\underline{\hat{x}}_{k-1}, \underline{u}_{k-1}}, \underline{C}_k = \left. \frac{\partial g}{\partial \underline{x}} \right|_{\underline{\hat{x}}_{k-1}, \underline{u}_{k-1}}, \underline{D}_k = \left. \frac{\partial g}{\partial \underline{u}} \right|_{\underline{\hat{x}}_{k-1}, \underline{u}_{k-1}}, \Delta \underline{y}_k = \underline{y}_k - \underline{\hat{y}}_{k-1}, \underline{v}_k = \underline{u}_k - \underline{u}_{k-1}$$

3) Tính:

$$\underline{e}_k(-) = \underline{A}_k \underline{e}_{k-1}(+) + \underline{B}_k \underline{v}_{k-1}$$

$$P_k(-) = \underline{A}_k P_{k-1}(+) \underline{A}_k^T + \Psi_{k-1}$$

$$K_k = P_k(-) \underline{C}_k^T (\underline{C}_k P_k(-) \underline{C}_k^T + \Phi_k)^{-1}$$

$$P_k(+) = (\underline{I} - K_k \underline{C}_k) P_k(-)$$

$$\underline{e}_k(+) = \underline{e}_k(-) + K_k (\Delta \underline{y}_k - \underline{C}_k \underline{e}_k(-) - \underline{D}_k \underline{v}_k)$$

4) Tính $\underline{\hat{x}}_k = \underline{e}_k(+) + \underline{\hat{x}}_{k-1}$ và xuất ra làm giá trị trạng thái quan sát được của hệ.

5) Tính $\underline{y}_k = \underline{g}(\underline{x}_k, \underline{u}_k)$. Gán $k := k+1$ và quay về 2).

Xác định xấp xỉ trạng thái đầu theo tiêu chuẩn tối ưu

Như các giả thiết đã nêu trong quá trình xây dựng thì thuật toán 2.2a vừa đề xuất ở trên chỉ làm việc tốt khi đã có trạng thái đầu \underline{x}_0 . Để xác định được \underline{x}_0 ta sẽ áp dụng nguyên lý tối ưu như đã thực hiện cho trường hợp hệ song tuyến (2.5), nhưng bây giờ là với hệ phi tuyến (2.1).

Để xác định được \underline{x}_0 , tất nhiên ta cần một khoảng thời gian quan sát ký hiệu là MT_a , tức là để có được \underline{x}_0 ta cần có được $M+1$ các giá trị vào ra đo được. Ta sẽ ký hiệu các giá trị này là $\underline{u}_i, \underline{y}_i, i = 0, 1, \dots, M$. Với các ký hiệu đó, thì từ mô hình phi tuyến tổng quát của hệ là (2.1) và khi xem sự ảnh hưởng của của nhiễu là nguyên nhân gây ra sự sai lệch ở đầu ra, ta sẽ có:

$$\underline{\bar{x}}_i = \underline{f}(\underline{\bar{x}}_{i-1}, \underline{u}_{i-1}) \quad (2.21)$$

trong đó $\underline{\bar{x}}_i$ là trạng thái của hệ khi không có nhiễu $\underline{\xi}_{i-1}$ tác động vào hệ thống. Tất nhiên ở thực tế giữa $\underline{\bar{x}}_i$ và trạng thái thực \underline{x}_i tồn tại một sai lệch. Sai lệch này sẽ được xử lý nhờ nguyên lý hiệu chỉnh tối ưu sau này.

Từ mô hình (2.21) ta được:

$$\begin{aligned} \underline{\bar{x}}_i &= \underline{f}(\underline{\bar{x}}_{i-1}, \underline{u}_{i-1}) = \underline{f}(\underline{f}(\underline{\bar{x}}_{i-2}, \underline{u}_{i-2}), \underline{u}_{i-1}) \\ &= \underline{f} \circ \underline{f}(\underline{\bar{x}}_{i-2}, \underline{u}_{i-2}, \underline{u}_{i-1}) = \underline{f}^2(\underline{\bar{x}}_{i-2}, \underline{u}_{i-2}, \underline{u}_{i-1}) \end{aligned}$$

trong đó ký hiệu $\underline{f}^2(\cdot)$ là ký hiệu của hàm hợp $\underline{f} \circ \underline{f}(\cdot)$. Cứ tiếp tục như vậy ta có:

$$\begin{aligned} \underline{\bar{x}}_i &= \underline{f}^2(\underline{\bar{x}}_{i-2}, \underline{u}_{i-2}, \underline{u}_{i-1}) \\ &= \underline{f}^3(\underline{\bar{x}}_{i-3}, \underline{u}_{i-3}, \underline{u}_{i-2}, \underline{u}_{i-1}) \\ &\vdots \\ &= \underline{f}^i(\underline{\bar{x}}_0, \underline{u}_0, \underline{u}_1, \dots, \underline{u}_{i-1}) \end{aligned} \quad (2.22)$$

với $\underline{f}^i(\cdot)$ là tích của i các hàm $\underline{f}(\cdot)$. Tất nhiên hàm này chỉ có một biến số là $\underline{\bar{x}}_0$ vì $\underline{u}_0, \underline{u}_1, \dots, \underline{u}_{i-1}$ là đã biết nên chỉ được xem như tham số của hàm.

Thay (2.22) vào phương trình đầu ra của (2.1), ta có:

$$\begin{aligned}\underline{y}_i &= \underline{g}(\underline{x}_i, \underline{u}_i) = \underline{g}\left(\underline{f}^i(\underline{x}_0, \underline{u}_0, \underline{u}_1, \dots, \underline{u}_{i-1}), \underline{u}_i\right) \\ &= \underline{g} \circ \underline{f}^i(\underline{x}_0, \underline{u}_0, \underline{u}_1, \dots, \underline{u}_{i-1}, \underline{u}_i)\end{aligned}\quad (2.23)$$

và hàm này cũng chỉ có một biến số là \underline{x}_0 . Suy ra hàm mô tả sai lệch quan sát:

$$\underline{e}_i = \underline{y}_i - \underline{y}_i = \underline{y}_i - \underline{g} \circ \underline{f}^i(\underline{x}_0, \underline{u}_0, \underline{u}_1, \dots, \underline{u}_{i-1}, \underline{u}_i) \quad (2.24)$$

cũng chỉ phụ thuộc \underline{x}_0 . Nhiệm vụ tiếp theo là xác định \underline{x}_0 sao cho tổng bình phương các sai lệch quan sát trong toàn bộ cửa sổ quan sát $i = 0, 1, \dots, M$ là nhỏ nhất.

Viết chung lại tất cả các hàm mô tả sai lệch quan sát trong cửa sổ quan sát thành vector sai lệch:

$$\mathbf{e}(\underline{x}_0) = \text{col}(\underline{e}_0, \underline{e}_1, \dots, \underline{e}_M) \quad (2.25)$$

thì nhiệm vụ xác định \underline{x}_0 theo tiêu chuẩn tối ưu nêu trên sẽ chính là:

$$\underline{x}_0^* = \arg \min_{\underline{x}_0} J(\underline{x}_0) \quad \text{với } J(\underline{x}_0) = \mathbf{e}(\underline{x}_0)^T \Lambda \mathbf{e}(\underline{x}_0) \quad (2.26)$$

trong đó Λ là ma trận đối xứng xác định dương tùy chọn, được gọi là ma trận trọng số. Vì hàm mục tiêu $J(\underline{x}_0)$ cho bởi công thức (2.26) ở trên có các vector hàm $\mathbf{e}(\underline{x}_0)$ định nghĩa theo (2.25), (2.24) và (2.23) là phi tuyến, nên để tìm được nghiệm tối ưu \underline{x}_0^* của nó ta cần phải có các phương pháp quy hoạch phi tuyến. Những phương pháp quy hoạch phi tuyến, từ cơ bản đến nâng cao, đã được trình bày chi tiết trong nhiều tài liệu tối ưu hóa, chẳng hạn như [12,53].

Vậy thuật toán xác định giá trị trạng thái đầu \underline{x}_0 cho hệ phi tuyến (2.1) từ $M+1$ các giá trị vào ra đo được $\underline{u}_i, \underline{y}_i, i = 0, 1, \dots, M$ sẽ có cấu trúc như sau:

Thuật toán 2.2b: Xác định trạng thái đầu \underline{x}_0 .

1) Chọn cửa sổ quan sát $M \geq 2$.

Đo $M+1$ các giá trị vào ra $\underline{u}_i, \underline{y}_i, i = 0, 1, \dots, M$.

2) Xây dựng các vector hàm hợp $\underline{f}^i(\cdot), \underline{g} \circ \underline{f}^i(\cdot), i = 1, \dots, M$ theo (2.22) và (2.23).

3) Lập hàm mục tiêu $J(\underline{x}_0)$ theo (2.24), (2.25) và (2.26).

- 4) Tìm nghiệm tối ưu \bar{x}_0^* của bài toán (2.26) nhờ phương pháp quy hoạch phi tuyến.

Thuật toán quan sát từng đoạn trạng thái cho hệ phi tuyến

Nghiệm \bar{x}_0^* tìm được nhờ thuật toán 2.2b chính là giá trị xấp xỉ tối ưu của trạng thái đầu x_0 của hệ. Nhưng vì để có \bar{x}_0^* ta phải đo được tất cả $M+1$ các giá trị vào ra $\underline{u}_i, \underline{y}_i, i=0,1, \dots, M$ của hệ, nên lúc này \bar{x}_0^* không còn là giá trị trạng thái tức thời của hệ. Giá trị trạng thái tức thời bây giờ phải là \underline{x}_M và giá trị này hoàn toàn có thể xác định được từ \bar{x}_0^* theo công thức lặp (2.22) như sau:

$$\bar{x}_M = \underline{f}^M(\bar{x}_0^*, \underline{u}_0, \underline{u}_1, \dots, \underline{u}_{M-1}) \quad (2.27)$$

Ghép chung hai thuật toán 2.2a và thuật toán 2.2b cùng với công thức (2.27) ta có được thuật toán quan sát từng đoạn cho hệ phi tuyến (2.1) gồm các bước như sau:

Thuật toán 2.2: Quan sát trạng thái hệ phi tuyến.

- 1) Chọn cửa sổ quan sát $M \geq 2$.
 - a) Đo $M+1$ các giá trị vào ra $\underline{u}_i, \underline{y}_i, i=0,1, \dots, M$.
 - b) Xây dựng các vector hàm hợp $\underline{f}^i(\cdot), \underline{g} \circ \underline{f}^i(\cdot), i=1, \dots, M$ theo (2.22) và (2.23). Từ đó lập hàm mục tiêu $J(\bar{x}_0)$ theo (2.24), (2.25) và (2.26).
 - c) Tìm nghiệm tối ưu \bar{x}_0^* của bài toán (2.26) nhờ phương pháp quy hoạch phi tuyến.
 - d) Xác định các giá trị trạng thái $\bar{x}_i, i=1,2, \dots, M$ từ \bar{x}_0^* theo (2.27).
- 2) Gán $\underline{e}_M(+)=\underline{0}, k=M+1$.
 Tùy chọn $P_M(+)$ và tính $\bar{y}_M = \underline{g}(\bar{x}_M, \underline{u}_M)$.
- 3) Đo $\underline{u}_{k-1}, \underline{u}_k, \underline{y}_k$. Tính:

$$A_k = \frac{\partial \underline{f}}{\partial \underline{x}} \Big|_{\bar{x}_{k-1}, \underline{u}_{k-1}}, B_k = \frac{\partial \underline{f}}{\partial \underline{u}} \Big|_{\bar{x}_{k-1}, \underline{u}_{k-1}}, C_k = \frac{\partial \underline{g}}{\partial \underline{x}} \Big|_{\bar{x}_{k-1}, \underline{u}_{k-1}}, D_k = \frac{\partial \underline{g}}{\partial \underline{u}} \Big|_{\bar{x}_{k-1}, \underline{u}_{k-1}} \quad \text{và } \Delta \underline{y}_k = \underline{y}_k - \bar{y}_{k-1}$$

4) Tính:

$$e_k(-) = A_k e_{k-1}(+) + B_k v_{k-1}$$

$$P_k(-) = A_k P_{k-1}(+) A_k^T + \Psi_{k-1}$$

$$K_k = P_k(-) C_k^T (C_k P_k(-) C_k^T + \Phi_k)^{-1}$$

$$P_k(+) = (I - K_k C_k) P_k(-)$$

$$e_k(+) = e_k(-) + K_k (\Delta y_k - C_k e_k(-) - D_k v_k)$$

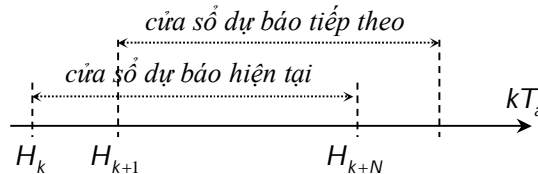
5) Tính $\bar{x}_k = e_k(+) + \bar{x}_{k-1}$ và xuất ra làm giá trị trạng thái quan sát được của hệ.

6) Tính $\bar{y}_k = g(\bar{x}_k, u_k)$. Gán $k := k+1$ và quay về 3).

Tuy nhiên, cũng giống như Thuật toán 1.6 về bộ điều khiển dự báo phản hồi trạng thái hệ phi tuyến, bộ quan sát trạng thái với Thuật toán 2.2 đề xuất trên đây sẽ khó cài đặt được do phải xác định các vector hàm hợp (2.23) ở bước đầu tiên. Bởi vậy, chỉ nên sử dụng thuật toán quan sát này khi có thể xấp xỉ được gần đúng giá trị trạng thái \bar{x}_0 đầu tiên. Trong trường hợp như vậy Thuật toán 2.2 sẽ trở thành thuật toán 2.2a.

2.2 Điều khiển dự báo phản hồi trạng thái hệ phi tuyến trên cơ sở sử dụng mô hình dự báo tuyến tính

Trước đây, ở mục 1.2.4 thuộc phần tổng quan, luận án đã trình bày nguyên tắc của điều khiển dự báo phản hồi đầu ra theo nguyên lý tách là ghép chung bộ điều khiển dự báo phi tuyến, được xây dựng trên nền Thuật toán 1.6, và bộ quan sát trạng thái, mà ở đây, trong khuôn khổ luận án, sẽ chính là các bộ lọc Kalman. Nguyên tắc điều khiển này cũng đã được minh họa ở Hình 1.2.



Hình 2.4: Nguyên lý điều khiển dự báo phản hồi trạng thái phi tuyến trên cơ sở sử dụng mô hình dự báo tuyến tính

Tuy nhiên, như đã phân tích ở mục 1.2.1, việc cài đặt trực tiếp Thuật toán 1.6 cho bộ điều khiển phản hồi trạng thái phi tuyến một cách tổng quát là hoàn toàn không khả thi do ở đó ta phải xác định được tường minh các công thức hàm hợp:

$$\underline{f}_{-m}^g(\underline{x}_k, \underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+m-1}) = (\underline{g} \circ \underbrace{\underline{f} \circ \underline{f} \circ \dots \circ \underline{f}}_m)(\underline{x}_k, \underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+m})$$

với $m = 1, 2, \dots, N$. Bởi vậy, sau đây luận án sẽ giới thiệu một phương pháp điều khiển dự báo khác cho hệ phi tuyến mang tính ứng dụng cao hơn. Phương pháp này được xây dựng trên nền tuyến tính hóa từng đoạn hệ phi tuyến dọc theo trục thời gian và sẽ được thực hiện đồng thời cùng với bước dịch chuyển của cửa sổ dự báo có độ rộng hữu hạn như minh họa ở Hình 2.4, trong đó H_k là các mô hình tuyến tính được sử dụng để dự báo tín hiệu đầu ra \underline{y}_{k+i} cho hệ phi tuyến (2.1) thuộc cửa sổ dự báo hiện tại $[k, k+N]$, tức là với $0 \leq i \leq N$.

2.2.1 Điều khiển hệ song tuyến

Điều khiển hệ hợp thức không chặt

Xét hệ song tuyến theo các biến trạng thái (2.2):

$$\begin{cases} \underline{x}_{k+1} = A(\underline{x}_k, k)\underline{x}_k + B(\underline{x}_k, k)\underline{u}_k \\ \underline{y}_k = C(\underline{x}_k, k)\underline{x}_k + D(\underline{x}_k, k)\underline{u}_k \end{cases} \quad (2.28)$$

Hệ trên có tín hiệu ra \underline{y}_k phụ thuộc tín hiệu vào \underline{u}_k ở cùng thời điểm nên được gọi là hệ hợp thức không chặt. Tên gọi này được bắt nguồn từ hệ (2.28) khi ở dạng tuyến tính tham số hằng sẽ có ma trận hàm truyền với các phần tử là thực-hữu tỷ có bậc đa thức tử số bằng bậc đa thức mẫu số, được gọi là hàm hợp thức không chặt (not strictly proper).

Với bài toán điều khiển phản hồi trạng thái, tức là với giả thiết trạng thái \underline{x}_k ở thời điểm $t = kT_a$ là đã biết, hệ cũng ở thời điểm đó sẽ trở thành:

$$H_k : \begin{cases} \underline{x}_{k+1} = A_k \underline{x}_k + B_k \underline{u}_k \\ \underline{y}_k = C_k \underline{x}_k + D_k \underline{u}_k \end{cases} \quad (2.29)$$

trong đó:

$$A_k = A(\underline{x}_k, k), \quad B_k = B(\underline{x}_k, k), \quad C_k = C(\underline{x}_k, k), \quad D_k = D(\underline{x}_k, k) \quad (2.30)$$

lúc này trở thành các ma trận hằng. Ta sẽ sử dụng mô hình tuyến tính (2.29) này để dự báo các tín hiệu ra \underline{y}_{k+i} thuộc cửa sổ dự báo hiện tại. Ý tưởng sử dụng mô hình dự báo tuyến tính (2.29) để dự báo tín hiệu đầu ra \underline{y}_{k+i} cho hệ song tuyến (2.28) đã được đề xuất bởi tài liệu [1,57]. Tuy nhiên khác với phương pháp điều khiển giới thiệu ở đó với việc sử dụng *cửa sổ dự báo vô hạn*, trong luận án này tác giả sẽ xây dựng bộ điều khiển dự báo với *cửa sổ dự báo hữu hạn* như mô tả trong Hình 2.4, tức là chỉ với các đầu ra dự báo \underline{y}_{k+i} , $0 \leq i \leq N$. Việc này sẽ kéo theo khối lượng tính toán trở nên nhiều hơn, song bù lại ta lại có thể dễ dàng xử lý được các điều kiện ràng buộc $\underline{u}_k \in U$ nếu có, cũng như trực tiếp điều khiển để hệ bám theo được tín hiệu ra mẫu $\{\underline{w}_k\}$ cho trước, thay vì chỉ điều khiển ổn định.

Từ mô hình dự báo tuyến tính (2.29) ta suy ra được:

$$\begin{aligned} \underline{x}_{k+i} &= A_k \underline{x}_{k+i-1} + B_k \underline{u}_{k+i-1} = A_k (A_k \underline{x}_{k+i-2} + B_k \underline{u}_{k+i-2}) + B_k \underline{u}_{k+i-1} \\ &= A_k^2 \underline{x}_{k+i-2} + A_k B_k \underline{u}_{k+i-2} + B_k \underline{u}_{k+i-1} \\ &= A_k^2 (A_k \underline{x}_{k+i-3} + B_k \underline{u}_{k+i-3}) + A_k B_k \underline{u}_{k+i-2} + B_k \underline{u}_{k+i-1} \\ &= A_k^3 \underline{x}_{k+i-3} + A_k^2 B_k \underline{u}_{k+i-3} + A_k B_k \underline{u}_{k+i-2} + B_k \underline{u}_{k+i-1} \\ &\vdots \\ &= A_k^i \underline{x}_k + A_k^{i-1} B_k \underline{u}_k + \dots + A_k B_k \underline{u}_{k+i-2} + B_k \underline{u}_{k+i-1} \end{aligned}$$

và từ đó là:

$$\begin{aligned} \underline{y}_{k+i} &= C_k \underline{x}_{k+i} + D_k \underline{u}_{k+i} \\ &= C_k (A_k^i \underline{x}_k + A_k^{i-1} B_k \underline{u}_k + \dots + A_k B_k \underline{u}_{k+i-2} + B_k \underline{u}_{k+i-1}) + D_k \underline{u}_{k+i} \\ &= C_k A_k^i \underline{x}_k + C_k A_k^{i-1} B_k \underline{u}_k + \dots + C_k A_k B_k \underline{u}_{k+i-2} + C_k B_k \underline{u}_{k+i-1} + D_k \underline{u}_{k+i} \end{aligned}$$

Bởi vậy, nếu sử dụng ký hiệu:

$$\underline{p} = \text{col}(\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N}) \quad (2.31)$$

thì toàn bộ giá trị đầu ra dự báo \underline{y}_{k+i} , $0 \leq i \leq N$ sẽ được viết chung lại thành:

$$\mathbf{y} = \begin{pmatrix} \underline{y}_k \\ \underline{y}_{k+1} \\ \vdots \\ \underline{y}_{k+N} \end{pmatrix} = \begin{pmatrix} C_k \\ C_k A_k \\ \vdots \\ C_k A_k^N \end{pmatrix} \underline{x}_k + \begin{pmatrix} D_k & \Theta & \dots & \Theta & \Theta \\ C_k B_k & D_k & \dots & \Theta & \Theta \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_k A_k^{N-1} B_k & C_k A_k^{N-2} B_k & \dots & C_k B_k & D_k \end{pmatrix} \underline{p}$$

$$= E \underline{x}_k + F \underline{p} \quad (2.32)$$

trong đó Θ là ký hiệu của ma trận có tất cả các phần tử bằng 0 và

$$E = \begin{pmatrix} C_k \\ C_k A_k \\ \vdots \\ C_k A_k^N \end{pmatrix}, \quad F = \begin{pmatrix} D_k & \Theta & \dots & \Theta & \Theta \\ C_k B_k & D_k & \dots & \Theta & \Theta \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_k A_k^{N-1} B_k & C_k A_k^{N-2} B_k & \dots & C_k B_k & D_k \end{pmatrix} \quad (2.33)$$

Do \underline{x}_k là đã biết nên \mathbf{y} trong (2.32) chỉ còn là hàm phụ thuộc \underline{p} trong đó chứa \underline{u}_k . Để xác định tín hiệu đầu vào \underline{u}_k ở thời điểm hiện tại $t = kT_a$ sao cho hệ bám theo được tín hiệu ra mẫu $\{\underline{w}_k\}$, ta xây dựng hàm sai lệch bám trong toàn bộ cửa sổ dự báo hiện tại như sau:

$$\mathbf{e} = \begin{pmatrix} \underline{w}_k \\ \underline{w}_{k+1} \\ \vdots \\ \underline{w}_{k+N} \end{pmatrix} - \mathbf{y} = \begin{pmatrix} \underline{w}_k \\ \underline{w}_{k+1} \\ \vdots \\ \underline{w}_{k+N} \end{pmatrix} - E \underline{x}_k - F \underline{p} = \mathbf{z} - F \underline{p} \quad \text{với } \mathbf{z} = \begin{pmatrix} \underline{w}_k \\ \underline{w}_{k+1} \\ \vdots \\ \underline{w}_{k+N} \end{pmatrix} - E \underline{x}_k \quad (2.34)$$

là một vector đã biết.

Từ đây, với mục đích cực tiểu hóa hàm mục tiêu về sai lệch dự báo:

$$\begin{aligned} J(\underline{p}) &= \mathbf{e}^T \Lambda_k \mathbf{e} + \underline{p}^T L_k \underline{p} = (\mathbf{z} - F \underline{p})^T \Lambda_k (\mathbf{z} - F \underline{p}) + \underline{p}^T L_k \underline{p} \\ &= \mathbf{z}^T \Lambda_k \mathbf{z} - 2 \mathbf{z}^T \Lambda_k F \underline{p} + \underline{p}^T (F^T \Lambda_k F + L_k) \underline{p} \xrightarrow{\underline{p}} \min \end{aligned}$$

$$\Leftrightarrow J'(\underline{p}) = -2 \mathbf{z}^T \Lambda_k F \underline{p} + \underline{p}^T (F^T \Lambda_k F + L_k) \underline{p} \xrightarrow{\underline{p}} \min$$

có hai ma trận trọng số Λ_k , L_k đối xứng xác định dương là tùy chọn, ta sẽ có được nghiệm tối ưu cho trường hợp không có ràng buộc $\underline{u}_k \in U$ như sau [12]:

$$\underline{p}^* = (F^T \Lambda_k F + L_k)^{-1} F^T \Lambda_k \mathbf{z}$$

Suy ra tín hiệu điều khiển \underline{u}_k cần tìm ở thời điểm hiện tại $t = kT_a$ sẽ là:

$$\underline{u}_k = (I, \Theta, \dots, \Theta) \underline{p}^* = (I, \Theta, \dots, \Theta) (F^T \Lambda_k F + L_k)^{-1} F^T \Lambda_k \mathbf{z}. \quad (2.35)$$

Vậy thuật toán điều khiển dự báo phản hồi trạng thái cho hệ song tuyến không liên tục (2.28) để tìm hiệu đầu ra của nó bám theo được dãy tín hiệu mẫu $\{\underline{w}_k\}$ sẽ gồm các bước như sau:

Thuật toán 2.3: Điều khiển bám tín hiệu đầu ra mẫu cho hệ song tuyến (2.28) bằng bộ điều khiển dự báo phản hồi trạng thái.

- 1) Gán $k = 0$. Chọn độ rộng cửa sổ dự báo $N > 2$.
- 2) Đo trạng thái \underline{x}_k (hoặc quan sát) và từ đó xác định các ma trận A_k, B_k, C_k, D_k theo (2.30), E, F theo (2.33), vector \mathbf{z} theo (2.34). Chọn hai ma trận trọng số Λ_k, L_k đối xứng xác định dương.
- 3) Tính \underline{u}_k theo (2.35) rồi đưa vào điều khiển hệ (2.28) trong khoảng thời gian đúng bằng chu kỳ trích mẫu T_a .
- 4) Gán $k := k + 1$ rồi quay về bước 2).

Trong thuật toán trên, nếu Λ_k, L_k được chọn không phụ thuộc vào k thì chúng sẽ là hai ma trận hằng và việc chọn $\Lambda = \Lambda_k, L = L_k$ sẽ được thực hiện ngay từ bước 1. Ngoài ra, do kết quả \underline{u}_k tại bước thứ k chỉ là phần tử đầu tiên của nghiệm tối ưu \underline{p}^* có liên quan tới thành phần $\underline{e}_k = \underline{w}_k - \underline{y}_k$ mà ở đó \underline{u}_k có tác dụng, nên khi chọn Λ_k , ta có thể chọn nó dạng ma trận đường chéo $\Lambda_k = \text{diag}(\lambda_j^k)$ với các phần tử trên đường chéo thỏa mãn $\lambda_1^k > \lambda_j^k > 0, j = 2, 3, \dots, M$, trong đó M là số phần tử của vector sai lệch dự báo \mathbf{e} , tức là $M = \dim(\mathbf{e})$.

Hơn nữa, khi bài toán có thêm điều kiện ràng buộc $\underline{u}_k \in U$ thì thay vì sử dụng trực tiếp công thức xác định nghiệm tối ưu (2.35) ta sẽ sử dụng phương pháp quy hoạch phi tuyến QP (khi U được mô tả bằng các bất phương trình tuyến tính), hoặc SQP [12,53] để tìm \underline{p}^* và từ đó là $\underline{u}_k \in U$.

Điều khiển hệ hợp thức chặt

Tiếp theo, ta xét trường hợp đặc biệt, khi mà đối tượng điều khiển (2.28) có tín hiệu ra \underline{y}_k không phụ thuộc trực tiếp tín hiệu vào, mô tả bởi:

$$\begin{cases} \underline{x}_{k+1} = A(\underline{x}_k, k)\underline{x}_k + B(\underline{x}_k, k)\underline{u}_k \\ \underline{y}_k = C(\underline{x}_k, k)\underline{x}_k \end{cases} \quad (2.36)$$

Lớp hệ trên được gọi là hệ hợp thức chặt. Tên gọi này được suy ra từ hệ tuyến tính tương ứng mà ở đó có ma trận hàm truyền với các phân tử bậc dạng thực-hữu tỷ có bậc đa thức tử số luôn nhỏ hơn bậc đa thức mẫu số (strickly proper).

Khi đó, tương ứng mô hình dự báo tuyến tính tương ứng (2.29) của cửa sổ dự báo hiện tại $[k, k+N)$ cũng sẽ trở thành:

$$H_k : \begin{cases} \underline{x}_{k+1} = A_k \underline{x}_k + B_k \underline{u}_k \\ \underline{y}_k = C_k \underline{x}_k \end{cases}$$

và hai ma trận E, F trong (2.33) sẽ được rút gọn lại thành:

$$E = \begin{pmatrix} C_k A_k \\ \vdots \\ C_k A_k^N \end{pmatrix}, \quad F = \begin{pmatrix} C_k B_k & \Theta & \dots & \Theta \\ \vdots & \vdots & \ddots & \vdots \\ C_k A_k^{N-1} B_k & C_k A_k^{N-2} B_k & \dots & C_k B_k \end{pmatrix} \quad (2.37)$$

cũng như vector \underline{p} định nghĩa trong (2.31) bây giờ sẽ là:

$$\underline{p} = \text{col}(\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N-1}).$$

Về vector sai lệch \mathbf{e} cho bởi (2.34), nhưng khi chỉ tính riêng trong cửa sổ dự báo hiện tại \underline{y}_{k+i} , $1 \leq i \leq N$, sẽ là:

$$\mathbf{e} = \begin{pmatrix} \underline{w}_{k+1} \\ \vdots \\ \underline{w}_{k+N} \end{pmatrix} - \mathbf{y} = \begin{pmatrix} \underline{w}_{k+1} \\ \vdots \\ \underline{w}_{k+N} \end{pmatrix} - E \underline{x}_k - F \underline{p} = \mathbf{z} - F \underline{p} \quad \text{với} \quad \mathbf{z} = \begin{pmatrix} \underline{w}_{k+1} \\ \vdots \\ \underline{w}_{k+N} \end{pmatrix} - E \underline{x}_k \quad (2.38)$$

Vậy thuật toán điều khiển dự báo phản hồi trạng thái cho hệ song tuyến mang tính quán tính thuần túy (2.36) để tín hiệu đầu ra của nó bám theo được dãy tín hiệu mẫu $\{\underline{w}_k\}$ sẽ được chỉnh sửa lại với các bước như sau:

Thuật toán 2.4: Điều khiển bám tín hiệu đầu ra mẫu cho hệ song tuyến (2.36) bằng bộ điều khiển dự báo phản hồi trạng thái.

- 1) Gán $k = 0$. Chọn độ rộng cửa sổ dự báo $N > 2$.
- 2) Đo trạng thái \underline{x}_k (hoặc quan sát) và từ đó xác định các ma trận A_k, B_k, C_k theo (2.30), E, F (2.37), vector \mathbf{z} theo (2.38). Chọn hai ma trận trọng số Λ_k, L_k đối xứng xác định dương.
- 3) Tính \underline{u}_k theo (2.35) rồi đưa vào điều khiển hệ (2.36) trong khoảng thời gian đúng bằng chu kỳ trích mẫu T_a .
- 4) Gán $k := k + 1$ rồi quay về bước 2).

Tương tự như ở trường hợp hệ hợp thức không chặt, trong thuật toán trên, nếu Λ_k, L_k được chọn không phụ thuộc vào k thì chúng sẽ là hai ma trận hằng và việc chọn $\Lambda = \Lambda_k, L = L_k$ sẽ được thực hiện ngay từ bước 1. Cũng như vậy, do kết quả \underline{u}_k tại bước thứ k chỉ là phần tử đầu tiên của nghiệm tối ưu \underline{p}^* có liên quan tới thành phần $\underline{e}_k = \underline{w}_k - \underline{y}_k$ mà ở đó \underline{u}_k có tác dụng, nên khi chọn Λ_k , ta có thể chọn nó dạng ma trận đường chéo $\Lambda_k = \text{diag}(\lambda_j^k)$ với các phần tử trên đường chéo thỏa mãn $\lambda_1^k > \lambda_j^k > 0, j = 2, 3, \dots, M$, trong đó $M = \dim(\mathbf{e})$.

Ngoài ra, khi bài toán có thêm điều kiện ràng buộc $\underline{u}_k \in U$ thì thay vì sử dụng trực tiếp công thức xác định nghiệm tối ưu (2.35) ta sẽ sử dụng phương pháp quy hoạch phi tuyến QP (trong trường hợp U được mô tả bằng các bất phương trình tuyến tính), hoặc SQP [12,53], trong trường hợp U mô tả bởi các bất phương trình phi tuyến, để tìm \underline{p}^* và từ đó là $\underline{u}_k \in U$.

2.2.2 Điều khiển hệ phi tuyến

Điều khiển hệ hợp thức không chặt

Cũng theo tư tưởng sử dụng mô hình dự báo tuyến tính để điều khiển bám cho hệ song tuyến như ở mục trên, sau đây luận án sẽ mở rộng sang cho cả hệ phi tuyến hợp thức không chặt, mô tả bởi:

$$\begin{cases} \underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{u}_k) \\ \underline{y}_k = C(\underline{x}_k, \underline{u}_{k-1})\underline{x}_k + D(\underline{x}_k, \underline{u}_{k-1})\underline{u}_k \end{cases} \quad (2.39)$$

trong đó $C(\underline{x}_k, \underline{u}_{k-1})$, $D(\underline{x}_k, \underline{u}_{k-1})$ là các ma trận phụ thuộc trạng thái ở thời điểm hiện tại \underline{x}_k và tín hiệu đầu vào ở thời điểm trước đó \underline{u}_{k-1} , cũng như $\underline{f}(\underline{x}_k, \underline{u}_k)$ là vector hàm tron, khả vi hai lần. Cũng giống như ở hệ song tuyến trước đây, tên gọi hợp thức không chặt được bắt nguồn từ hệ tuyến tính có ma trận hàm truyền với các phân tử là hàm phức, thực-hữu tỷ có bậc đa thức tử số bằng bậc đa thức mẫu số.

Để xây dựng mô hình dự báo tín hiệu ra dạng tuyến tính cho hệ phi tuyến (2.39) ở thời điểm k hiện tại trước tiên ta phân tích vế phải của thành phần thứ nhất trong (2.39) thành chuỗi Taylor tại điểm quá khứ $\underline{x}_{k-1}, \underline{u}_{k-1}$ ta được:

$$\begin{aligned} \underline{f}(\underline{x}_k, \underline{u}_k) &\approx \underline{f}(\underline{x}_{k-1}, \underline{u}_{k-1}) + \underline{A}_k (\underline{x}_k - \underline{x}_{k-1}) + \underline{B}_k (\underline{u}_k - \underline{u}_{k-1}) \\ &= \underline{A}_k \underline{x}_k + \underline{B}_k \underline{u}_k + \underline{d}_k \end{aligned}$$

trong đó:

$$\underline{A}_k = \left. \frac{\partial \underline{f}}{\partial \underline{x}} \right|_{\underline{x}_{k-1}, \underline{u}_{k-1}}, \quad \underline{B}_k = \left. \frac{\partial \underline{f}}{\partial \underline{u}} \right|_{\underline{x}_{k-1}, \underline{u}_{k-1}}, \quad \underline{d}_k = \underline{f}(\underline{x}_{k-1}, \underline{u}_{k-1}) - \underline{A}_k \underline{x}_{k-1} - \underline{B}_k \underline{u}_{k-1} \quad (2.40)$$

Hiển nhiên ở thời điểm k hiện tại thì các giá trị quá khứ \underline{x}_{k-1} , \underline{u}_{k-1} là đã biết nên hai ma trận \underline{A}_k , \underline{B}_k và vector \underline{d}_k là những đại lượng rõ (xác định được).

Suy ra thành phần thứ nhất trong mô hình phi tuyến là:

$$\underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{u}_k)$$

sẽ thay thế được bởi: $\underline{x}_{k+1} = \underline{A}_k \underline{x}_k + \underline{B}_k \underline{u}_k + \underline{d}_k$

và mô hình dự báo đầu ra dạng tuyến tính ở thời điểm k hiện tại cho hệ phi tuyến (2.39) bây giờ sẽ là:

$$\begin{cases} \underline{x}_{k+1} = A_k \underline{x}_k + B_k \underline{u}_k + \underline{d}_k \\ \underline{y}_k = C_k \underline{x}_k + D_k \underline{u}_k \end{cases} \quad (2.41)$$

$$\text{trong đó: } C_k = C(\underline{x}_k, \underline{u}_{k-1}), \quad D_k = D(\underline{x}_k, \underline{u}_{k-1}) \quad (2.42)$$

Với mô hình dự báo trên ta có được các giá trị trạng thái dự báo:

$$\underline{x}_{k+i}, \quad i=1, 2, \dots, N$$

thuộc cửa sổ dự báo hiện tại sẽ là:

$$\begin{aligned} \underline{x}_{k+i} &= A_k \underline{x}_{k+i-1} + B_k \underline{u}_{k+i-1} + \underline{d}_k \\ &= A_k (A_k \underline{x}_{k+i-2} + B_k \underline{u}_{k+i-2} + \underline{d}_k) + B_k \underline{u}_{k+i-1} + \underline{d}_k \\ &= A_k^2 \underline{x}_{k+i-2} + A_k B_k \underline{u}_{k+i-2} + \dots + B_k \underline{u}_{k+i-1} + A_k \underline{d}_k + \underline{d}_k \\ &= A_k^3 \underline{x}_{k+i-3} + A_k^2 B_k \underline{u}_{k+i-3} + \dots + B_k \underline{u}_{k+i-1} + A_k^2 \underline{d}_k + A_k \underline{d}_k + \underline{d}_k \\ &\vdots \\ &= A_k^i \underline{x}_k + A_k^{i-1} B_k \underline{u}_k + \dots + B_k \underline{u}_{k+i-1} + A_k^{i-1} \underline{d}_k + \dots + A_k \underline{d}_k + \underline{d}_k \end{aligned}$$

Suy ra đầu ra dự báo tương ứng \underline{y}_{k+i} , $i=1, 2, \dots, N$ là

$$\begin{aligned} \underline{y}_{k+i} &= C_k A_k^i \underline{x}_k + C_k A_k^{i-1} B_k \underline{u}_k + \dots + C_k B_k \underline{u}_{k+i-1} + D_k \underline{u}_{k+i} + C_k (A_k^{i-1} \underline{d}_k + \dots + A_k \underline{d}_k + \underline{d}_k) \\ &= C_k A_k^i \underline{x}_k + C_k A_k^{i-1} B_k \underline{u}_k + \dots + C_k B_k \underline{u}_{k+i-1} + D_k \underline{u}_{k+i} + \underline{d}_k^i \end{aligned}$$

trong đó:

$$\underline{d}_k^i = C_k (A_k^{i-1} \underline{d}_k + \dots + A_k \underline{d}_k + \underline{d}_k)$$

Giống như (2.31), vẫn ký hiệu:

$$\underline{p} = \text{col}(\underline{u}_k, \underline{u}_{k+1}, \dots, \underline{u}_{k+N}) \quad (2.43)$$

thì toàn bộ các tín hiệu ra dự báo \underline{y}_{k+i} , $i=1, 2, \dots, N$ thuộc cửa sổ dự báo hiện tại được viết chung lại thành:

$$\begin{pmatrix} \underline{y}_k \\ \underline{y}_{k+1} \\ \vdots \\ \underline{y}_{k+N} \end{pmatrix} = \begin{pmatrix} C_k \\ C_k A_k \\ \vdots \\ C_k A_k^N \end{pmatrix} \underline{x}_k + \begin{pmatrix} D_k & \Theta & \dots & \Theta & \Theta \\ C_k B_k & D_k & \dots & \Theta & \Theta \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_k A_k^{N-1} B_k & C_k A_k^{N-2} B_k & \dots & C_k B_k & D_k \end{pmatrix} \underline{p} + \begin{pmatrix} \underline{d}_k^0 \\ \underline{d}_k^1 \\ \vdots \\ \underline{d}_k^N \end{pmatrix} \quad (2.44)$$

với $\underline{d}_k^0 = \underline{d}_k$.

Giống như (2.33), lại ký hiệu tiếp:

$$\mathbf{y} = \begin{pmatrix} \underline{y}_k \\ \underline{y}_{k+1} \\ \vdots \\ \underline{y}_{k+N} \end{pmatrix}, E = \begin{pmatrix} C_k \\ C_k A_k \\ \vdots \\ C_k A_k^N \end{pmatrix}, F = \begin{pmatrix} D_k & \Theta & \cdots & \Theta & \Theta \\ C_k B_k & D_k & \cdots & \Theta & \Theta \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ C_k A_k^{N-1} B_k & C_k A_k^{N-2} B_k & \cdots & C_k B_k & D_k \end{pmatrix}, \mathbf{d} = \begin{pmatrix} \underline{d}_k^0 \\ \underline{d}_k^1 \\ \vdots \\ \underline{d}_k^N \end{pmatrix} \quad (2.45)$$

thì công thức dự báo trên sẽ được viết ngắn gọn lại thành:

$$\mathbf{y} = E \underline{x}_k + F \underline{p} + \mathbf{d} \quad (2.46)$$

Với các tín hiệu đầu ra dự báo như trên, ta sẽ có được sai lệch đầu ra dự báo:

$$\mathbf{e} = \begin{pmatrix} \underline{w}_k \\ \underline{w}_{k+1} \\ \vdots \\ \underline{w}_{k+N} \end{pmatrix} - \mathbf{y} = \begin{pmatrix} \underline{w}_k \\ \underline{w}_{k+1} \\ \vdots \\ \underline{w}_{k+N} \end{pmatrix} - E \underline{x}_k - F \underline{p} - \mathbf{d} = \begin{pmatrix} \underline{w}_k \\ \underline{w}_{k+1} \\ \vdots \\ \underline{w}_{k+N} \end{pmatrix} - E \underline{x}_k - \mathbf{d} - F \underline{p} = \mathbf{z} - F \underline{p} \quad (2.47)$$

trong đó:

$$\mathbf{z} = \begin{pmatrix} \underline{w}_k \\ \underline{w}_{k+1} \\ \vdots \\ \underline{w}_{k+N} \end{pmatrix} - E \underline{x}_k - \mathbf{d} \quad (2.48)$$

Đến đây thì hoàn toàn tương tự như ở trường hợp hệ song tuyến, ta cũng sẽ có được tín hiệu điều khiển \underline{u}_k tính theo công thức (2.35) với các tham số \mathbf{z} cho bởi (2.48) và E, F cho bởi (2.45).

Vậy thuật toán điều khiển dự báo phản hồi trạng thái để đầu ra \underline{y} của hệ phi tuyến (2.39) bám theo được dãy tín hiệu mẫu $\{\underline{w}_k\}$ cho trước sẽ có dạng như sau:

Thuật toán 2.5: Điều khiển bám tín hiệu đầu ra mẫu cho hệ phi tuyến (2.39) bằng bộ điều khiển dự báo phản hồi trạng thái.

- 1) Gán $\underline{u}_{-1} = \underline{0}$, $\underline{x}_{-1} = \underline{0}$, $k = 0$. Chọn độ rộng cửa sổ dự báo $N > 2$.
- 2) Đo trạng thái \underline{x}_k (hoặc quan sát) và từ đó xác định các ma trận A_k, B_k theo (2.40), C_k, D_k theo (2.42), E, F theo (2.45) và vector \mathbf{z} theo (2.48). Chọn hai ma trận trọng số Λ_k, L_k đối xứng xác định dương.
- 3) Tính \underline{u}_k theo (2.35) rồi đưa vào điều khiển hệ (2.39) trong khoảng thời gian đúng bằng chu kỳ trích mẫu T_a .
- 4) Gán $k := k + 1$ rồi quay về bước 2).

Cũng giống như ở hệ song tuyến (Thuật toán 2.3 và Thuật toán 2.4), trong Thuật toán 2.5 nêu trên, nếu Λ_k, L_k được chọn không phụ thuộc vào k thì việc chọn $\Lambda = \Lambda_k, L = L_k$ sẽ được thực hiện ngay từ bước 1. Ngoài ra, do kết quả \underline{u}_k tại bước thứ k chỉ là phần tử đầu tiên của nghiệm tối ưu \underline{p}^* có liên quan tới thành phần $\underline{e}_k = \underline{w}_k - \underline{y}_k$ mà ở đó \underline{u}_k có tác dụng, nên khi chọn Λ_k , ta có thể chọn nó dạng ma trận đường chéo $\Lambda_k = \text{diag}(\lambda_j^k)$ có các phần tử $\lambda_1^k > \lambda_j^k > 0, j = 2, 3, \dots, M$.

Điều khiển hệ hợp thức chặt

Hoàn toàn tương tự như, ở trường hợp đối tượng điều khiển (2.39) có đầu ra không phụ thuộc trực tiếp tín hiệu vào (tức là hệ tương ứng ở dạng tuyến tính tham số hằng sẽ có ma trận hàm truyền với các phần tử là hàm thực-hữu tỷ hợp thức chặt):

$$\begin{cases} \underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{u}_k) \\ \underline{y}_k = C(\underline{x}_k, \underline{u}_{k-1})\underline{x}_k \end{cases} \quad (2.49)$$

thì công thức (2.45) để tính E, F sẽ được thay bởi công thức giống như (2.37):

$$\mathbf{y} = \begin{pmatrix} \underline{y}_k \\ \vdots \\ \underline{y}_{k+N-1} \end{pmatrix}, E = \begin{pmatrix} C_k A_k \\ \vdots \\ C_k A_k^N \end{pmatrix}, F = \begin{pmatrix} C_k B_k & \Theta & \cdots & \Theta \\ \vdots & \vdots & \ddots & \vdots \\ C_k A_k^{N-1} B_k & C_k A_k^{N-2} B_k & \cdots & C_k B_k \end{pmatrix}, \mathbf{d} = \begin{pmatrix} \underline{d}_k^1 \\ \vdots \\ \underline{d}_k^N \end{pmatrix} \quad (2.50)$$

và công thức (2.48) để xác định \mathbf{z} bây giờ sẽ là:

$$\mathbf{z} = \begin{pmatrix} \underline{w}_{k+1} \\ \vdots \\ \underline{w}_{k+N} \end{pmatrix} - E \underline{x}_k - \mathbf{d} \quad (2.51)$$

Khi đó thuật toán Thuật toán 2.5 để điều khiển dự báo phản hồi trạng thái để đầu ra \underline{y} của hệ phi tuyến (2.39) khi áp dụng cho hệ (2.49) bám theo được dãy tín hiệu mẫu $\{\underline{w}_k\}$ cho trước sẽ được chỉnh sửa lại như sau:

Thuật toán 2.6: Điều khiển bám tín hiệu đầu ra mẫu cho hệ phi tuyến (2.49) bằng bộ điều khiển dự báo phản hồi trạng thái.

1) Gán $\underline{u}_{-1} = \underline{0}, \underline{x}_{-1} = \underline{0}, k = 0$. Chọn độ rộng cửa sổ dự báo $N > 2$.

- 2) Đo trạng thái \underline{x}_k (hoặc quan sát) và từ đó xác định các ma trận A_k, B_k theo (2.40), C_k theo (2.42), E, F theo (2.50) và vector \mathbf{z} theo (2.51). Chọn hai ma trận trọng số Λ_k, L_k đối xứng xác định dương.
- 3) Tính \underline{u}_k theo (2.35) rồi đưa vào điều khiển hệ (2.49) trong khoảng thời gian đúng bằng chu kỳ trích mẫu T_a .
- 4) Gán $k := k+1$ rồi quay về bước 2).

Cũng giống như ở Thuật toán 2.5, nếu Λ_k, L_k được chọn không phụ thuộc vào k thì việc chọn $\Lambda = \Lambda_k, L = L_k$ sẽ được thực hiện ngay từ bước 1.

Ngoài ra, do kết quả \underline{u}_k tại bước thứ k chỉ là phần tử đầu tiên của nghiệm tối ưu \underline{p}^* có liên quan tới thành phần $\underline{e}_k = \underline{w}_k - \underline{y}_k$ mà ở đó \underline{u}_k có tác dụng, nên khi chọn Λ_k , ta có thể chọn nó dạng ma trận đường chéo:

$$\Lambda_k = \text{diag}(\lambda_j^k)$$

có các phần tử:

$$\lambda_1^k > \lambda_j^k > 0, j = 2, 3, \dots, M$$

với $M = \dim(\mathbf{e})$.

2.3 Điều khiển dự báo phản hồi đầu ra hệ phi tuyến với các bộ lọc Kalman mở rộng

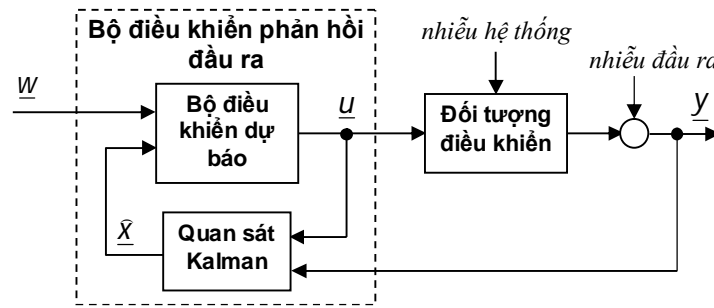
2.3.1 Thuật toán điều khiển

Cấu trúc điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ phi tuyến được mô tả ở Hình 2.5. Như vậy sau khi đã có các phương pháp quan sát trạng thái bằng bộ lọc Kalman và bộ điều khiển dự báo phản hồi trạng thái thì bộ điều khiển dự báo phản hồi đầu ra sẽ chính là sự mắc nối tiếp của hai bộ điều khiển này (bộ quan sát trạng thái và bộ điều khiển phản hồi trạng thái).

Về quan sát trạng thái Kalman luận án đã trình bày tổng cộng tất cả sáu thuật toán, trong đó có bốn thuật toán cơ bản gồm ba thuật toán EKF mở rộng (Thuật toán

1.7 - Thuật toán 1.9), một thuật toán UKF (Thuật toán 1.10) ở chương 1 (chương tổng quan) và hai thuật toán đề xuất thêm được xây dựng trên nền tuyến tính hóa từng đoạn ở chương 2 là Thuật toán 2.1 cho hệ song tuyến và Thuật toán 2.2 cho hệ phi tuyến.

Về bộ điều khiển dự báo phản hồi trạng thái luận án cũng đã đề xuất ở chương 2 hai bộ điều khiển được xây dựng trên nền tuyến tính hóa từng đoạn mô hình dự báo phi tuyến. Đó là bộ điều khiển ở Thuật toán 2.3 hoặc Thuật toán 2.4 cho các hệ song tuyến (2.28), (2.36) và bộ điều khiển ở Thuật toán 2.5 hoặc Thuật toán 2.6 cho hệ phi tuyến (2.39), (2.49).



Hình 2.5: Cấu trúc hệ điều khiển phản hồi đầu ra theo nguyên lý tách

Như vậy, về nguyên tắc ta sẽ có tổng cộng bốn bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ song tuyến (2.28) và bốn bộ điều khiển cho hệ phi tuyến (2.39).

Để minh họa, thuật toán sau đây sẽ mô tả các bước làm việc của bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách được xây dựng trên cơ sở ghép chung bộ quan sát trạng thái EKF loại 3 (Thuật toán 1.9) và bộ điều khiển dự báo phản hồi trạng thái phi tuyến (Thuật toán 2.5 hoặc Thuật toán 2.6) cho hệ (2.39) có đầu ra không phụ thuộc \underline{u}_{k-1} , tức là:

$$\begin{cases} \underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{u}_k) + \underline{\xi}_k \\ \underline{y}_k = C(\underline{x}_k)\underline{x}_k + D(\underline{x}_k)\underline{u}_k + \underline{\zeta}_k \end{cases} \quad (2.52)$$

hoặc:

$$\begin{cases} \underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{u}_k) + \underline{\xi}_k \\ \underline{y}_k = C(\underline{x}_k)\underline{x}_k + \underline{\zeta}_k \end{cases} \quad (2.53)$$

Thuật toán 2.7: Điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ phi tuyến (2.52) hoặc (2.53) với bộ lọc Kalman EKF loại 3.

- 1) Tùy chọn điều kiện đầu $\underline{x}_0 = \underline{x}_0(+)$ và $P_0(+)$. Đo $\underline{u}_0, \underline{y}_0$. Gán $k=1$.
- 2) Đo \underline{y}_k . Chọn hai ma trận trọng số Λ_k, L_k đối xứng xác định dương.
- 3) Tính:

$\underline{x}_k(-) = \underline{f}_{-k}(\underline{x}_{k-1}(+), \underline{u}_{k-1})$, G_k theo (1.47), trong đó $\underline{g}(\underline{x}_k, \underline{u}_k) = C(\underline{x}_k)\underline{x}_k + D(\underline{x}_k)\underline{u}_k$ khi đối tượng điều khiển là hệ (2.52) hoặc $\underline{g}(\underline{x}_k, \underline{u}_k) = C(\underline{x}_k)\underline{x}_k$ khi đối tượng điều khiển là (2.53). Tính F_{k-1} theo (1.45) và:

$$\begin{aligned} P_k(-) &= F_{k-1}P_{k-1}(+)F_{k-1}^T + \Psi_{k-1} \\ K_k &= P_k(-)G_k^T(G_kP_k(-)G_k^T + \Phi_k)^{-1} \\ P_k(+) &= (I - K_kG_k)P_k(-) \\ \underline{x}_k(+) &= \underline{x}_k(-) + K_k(\underline{y}_k - G_k\underline{x}_k(-)) \end{aligned}$$

- 4) Gán $\underline{x}_k = \underline{x}_k(+)$ và xác định các ma trận A_k, B_k theo (2.40), C_k, D_k theo công thức tương tự như (2.42) nhưng bây giờ được sửa đổi thành $C_k = C(\underline{x}_k)$, $D_k = D(\underline{x}_k)$, hai ma trận E, F theo (2.45) nếu đối tượng điều khiển là (2.52) hoặc theo (2.50) nếu đối tượng điều khiển là (2.53) và vector \mathbf{z} theo (2.48) hoặc theo (2.51).
- 5) Gán $k := k+1$.

Tính \underline{u}_k theo (2.35) rồi đưa vào điều khiển hệ (2.52) hoặc (2.53) trong khoảng thời gian đúng bằng chu kỳ trích mẫu T_a rồi quay về bước 2).

Thuật toán điều khiển dự báo phản hồi đầu ra theo nguyên lý tách ở trên hoàn toàn có thể chuyển đổi được một cách tương tự cho hệ song tuyến (2.28) và (2.36) bằng cách thay vì sử dụng Thuật toán 2.5 hoặc Thuật toán 2.6 tại bước 4) và 5) ta sử dụng Thuật toán 2.3 hoặc Thuật toán 2.4.

2.3.2 Tính ổn định ISS của bộ điều khiển phản hồi đầu ra

Tất cả các bộ điều khiển dự báo phản hồi trạng thái xây dựng trên nền quy hoạch phi tuyến trong luận án đều sử dụng mô hình dự báo không liên tục tuyến tính hóa từng đoạn (2.29), (2.41) có cửa sổ dự báo là hữu hạn để dự báo tín hiệu ra cho hệ phi tuyến. Đó cũng là điểm khác biệt so với phương pháp đã được trình bày trong các tài liệu [12,1,57]. Ngoài ra, một điểm khác biệt nữa của bộ điều khiển dự báo phản hồi đầu ra của luận án so với bộ điều khiển đã giới thiệu ở tài liệu [1] là luận án sử dụng bộ lọc Kalman phi tuyến để quan sát trạng thái thay vì bộ quan sát RHO tối ưu (optimal receding horizon observer).

Tiếp theo, luận án sẽ khẳng định tính ổn định của bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách đã được đề xuất ở Thuật toán 2.7.

Trước tiên ta ký hiệu:

$$\underline{u}_k^* = \mathcal{K}^{MPC}(\underline{x}_k) \quad (2.54)$$

là bộ điều khiển dự báo phản hồi trạng thái làm hệ (1.1) ổn định tiệm cận toàn cục theo nghĩa Lyapunov, tức là với nó tồn tại hàm hợp thức:

$$\gamma_1(|\underline{x}_k|) \leq V(\underline{x}_k, k) \leq \gamma_2(|\underline{x}_k|) \quad (2.55)$$

thỏa mãn:

$$\begin{aligned} \Delta V &= V(\underline{x}_{k+1}, k+1) - V(\underline{x}_k, k) = V\left(\underline{f}(\underline{x}_k, \underline{u}_k^*), k+1\right) - V(\underline{x}_k, k) \\ &= V\left(\underline{f}(\underline{x}_k, \mathcal{K}^{MPC}(\underline{x}_k)), k+1\right) - V(\underline{x}_k, k) \leq -\gamma_2(|\underline{x}_k|) \end{aligned} \quad (2.56)$$

trong đó $\gamma_i(r)$, $i = 1, 2, 3$ là ký hiệu của những hàm thuộc lớp \mathcal{K} , tức là những hàm thực của biến thực $r \geq 0$, đơn điệu tăng và $\gamma_i(0) = 0$, thì bài toán xét tính ổn định hệ phản hồi đầu ra theo nguyên lý tách ở đây tương đương với bài toán xét tính ổn định của hệ:

$$\underline{x}_{k+1} = \underline{f}\left(\underline{x}_k, \mathcal{K}^{MPC}(\underline{x}_k)\right) = \underline{f}\left(\underline{x}_k, \mathcal{K}^{MPC}(\underline{x}_k + \underline{e}_k)\right) \quad (2.57)$$

trong đó $\underline{e}_k = \underline{x}_k - \hat{\underline{x}}_k$ là ký hiệu của sai lệch quan sát tại thời điểm k .

Về bài toán vừa phát biểu, các tài liệu [31,60] cũng đã chứng minh được rằng với các hệ (2.1) có $\underline{f}(\underline{x}_k, \underline{u}_k)$ là vector hàm Lipschitz, chẳng hạn như hệ tuyến tính, thì quỹ đạo trạng thái $\{\underline{x}_k\}$ của hệ kín (2.57) sẽ luôn thỏa mãn:

$$|\underline{x}_k| \leq \beta(|\underline{x}_0|, k) + \gamma(|\underline{e}_k|)$$

trong đó $\gamma(|\underline{e}_k|) \in \mathcal{K}$ và $\beta(|\underline{x}_0|, k) \in \mathcal{KL}$, tức là hàm thực đơn điệu tăng theo $|\underline{x}_0|$ và đơn điệu giảm theo k . Mở rộng hơn nữa, tài liệu [1] cũng đã chỉ ra rằng nếu bộ điều khiển phản hồi trạng thái làm hệ ổn định tiệm cận và bộ quan sát luôn có sai lệch quan sát hữu hạn thì bộ điều khiển đầu ra tương ứng được xây dựng theo nguyên lý tách sẽ ổn định ISS (*input to state stable*), tức là dãy giá trị $\{\underline{x}_k\}$ sẽ luôn tiến về một lân cận nhỏ xung quanh gốc tọa độ. Khái niệm ổn định này còn được biết đến dưới tên gọi là ổn định thực tế (*practical stable*).

Tương tự, sau đây luận án cũng sẽ chứng minh tính ổn định ISS của bộ điều khiển phản hồi đầu ra cho bởi Thuật toán 2.7, tức là khẳng định tính ổn định ISS của hệ khi sử dụng bộ điều khiển phản hồi đầu ra $\underline{u}_k^* = \mathcal{K}^{MPC}(\bar{\underline{x}}_k)$, trong đó $\bar{\underline{x}}_k$ là giá trị trạng thái quan sát được của hệ nhờ các bộ lọc EKF/UKF.

Định lý: *Nếu các bộ điều khiển dự báo phản hồi trạng thái cho ở Thuật toán 2.3 - Thuật toán 2.6, ký hiệu bởi (2.54), là hàm Lipschitz, hệ phi tuyến (2.1) có vector hàm $\underline{f}(\underline{x}_k, \underline{u}_k)$ cũng là hàm Lipschitz, thì bộ điều khiển dự báo phản hồi đầu ra của luận án (Thuật toán 2.7) sẽ làm hệ ổn định ISS.*

Chứng minh:

Do có giả thiết rằng bộ điều khiển phản hồi trạng thái (2.54) thỏa mãn điều kiện Lipschitz nên ta luôn có:

$$|\mathcal{K}^{MPC}(\bar{\underline{x}}_k) - \mathcal{K}^{MPC}(\underline{x}_k)| \leq L_1 |\underline{e}_k|$$

trong đó $\bar{\underline{x}}_k$ là giá trị trạng thái quan sát được, $\underline{e}_k = \bar{\underline{x}}_k - \underline{x}_k$ là sai lệch quan sát và L_1 là hằng số Lipschitz của $\underline{u}_k^* = \mathcal{K}^{MPC}(\underline{x}_k)$.

Ngoài ra, do bộ điều khiển (2.54) làm hệ ổn định tiệm cận, nên phải có:

$$\left| \underline{f}(\underline{x}_k, \mathcal{K}^{MPC}(\underline{x}_k)) \right| \leq \beta(|\underline{x}_0|, k)$$

với $\beta \in \mathcal{KL}$ là hàm không âm, đơn điệu tăng theo $|\underline{x}_0|$ và đơn điệu giảm theo k .

Vậy, khi vector hàm $\underline{f}(\underline{x}_k, \underline{u}_k)$ cũng thỏa mãn điều kiện Lipschitz, tức là:

$$\left| \underline{f}(\underline{x}_k, \underline{u}_k) - \underline{f}(\underline{x}_k, \underline{v}_k) \right| \leq L_2 |\underline{u}_k - \underline{v}_k|$$

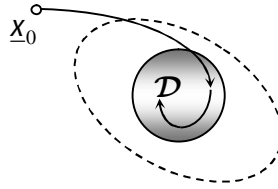
trong đó L_2 là hằng số Lipschitz tương ứng của $\underline{f}(\underline{x}_k, \underline{u}_k)$, thì quỹ đạo trạng thái của hệ kín được điều khiển bởi bộ điều khiển dự báo phản hồi đầu ra (Thuật toán 2.7) sẽ phải thỏa mãn:

$$\begin{aligned} \left| \underline{f}(\underline{x}_k, \mathcal{K}^{MPC}(\bar{\underline{x}}_k)) \right| &= \left| \underline{f}(\underline{x}_k, \mathcal{K}^{MPC}(\bar{\underline{x}}_k)) - \underline{f}(\underline{x}_k, \mathcal{K}^{MPC}(\underline{x}_k)) + \underline{f}(\underline{x}_k, \mathcal{K}^{MPC}(\underline{x}_k)) \right| \\ &\leq \left| \underline{f}(\underline{x}_k, \mathcal{K}^{MPC}(\bar{\underline{x}}_k)) - \underline{f}(\underline{x}_k, \mathcal{K}^{MPC}(\underline{x}_k)) \right| + \left| \underline{f}(\underline{x}_k, \mathcal{K}^{MPC}(\underline{x}_k)) \right| \\ &\leq L_2 \left| \mathcal{K}^{MPC}(\bar{\underline{x}}_k) - \mathcal{K}^{MPC}(\underline{x}_k) \right| + \left| \underline{f}(\underline{x}_k, \mathcal{K}^{MPC}(\underline{x}_k)) \right| \\ &\leq L_2 L_1 |\underline{e}_k| + \beta(|\underline{x}_0|, k) \end{aligned}$$

Điều này nói rằng với bộ quan sát có $\underline{e}_k = \bar{\underline{x}}_k - \underline{x}_k$ bị chặn thì quỹ đạo trạng thái $\{\underline{x}_k\}$ của hệ kín sẽ luôn tiến về một miền \mathcal{D} bao gốc và có đường kính d không lớn hơn:

$$d \leq L_2 L_1 \sup_k |\underline{e}_k| \quad (2.58)$$

Ngoài ra, nếu dãy $|\underline{e}_k|$ là đơn điệu giảm thì bán kính của \mathcal{D} cũng sẽ nhỏ dần như được minh họa như ở Hình 2.6 và đó là điều phải chứng minh. ■



Hình 2.6: Tính ổn định ISS của hệ kín phản hồi đầu ra theo nguyên lý tách

Như vậy, với định lý vừa chứng minh, tính ổn định ISS của hệ kín phản hồi đầu ra đã được khẳng định, ngay cả khi không có được $\bar{\underline{x}}_k \rightarrow \underline{x}_k$ trong một chu kỳ dịch chuyển của cửa sổ dự báo.

Ta còn có thể nhận thấy rằng kết quả trên (định lý) không những chỉ đúng với bộ quan sát Kalman (không liên tục), là nội dung nghiên cứu của luận án, mà còn đúng với mọi bộ quan sát khác, kể cả quan sát phi tuyến. Hơn thế nữa, khi bộ quan sát thỏa mãn thêm đơn điệu giảm của sai lệch $e_k = \bar{x}_k - \underline{x}_k$, miền hấp dẫn cho bởi (2.58) cũng sẽ có kích thước đơn điệu giảm tương ứng. Trong trường hợp tiệm cận $\lim_{k \rightarrow \infty} |e_k| = 0$ được thỏa mãn thì hệ kín phản hồi đầu ra sẽ còn là ổn định tiệm cận bán toàn cục (*semiglobal asymptotic stable*), tức là mọi quỹ đạo trạng thái của hệ xuất phát từ điểm trạng thái đầu \underline{x}_0 thuộc một miền compact (đóng và bị chặn) sẽ luôn tiệm cận được về gốc tọa độ.

2.4 Kết luận chương 2

Trong chương 2 luận án đã trình bày các nội dung sau:

- 1) Mở rộng lọc Kalman tuyến tính (KF) để áp dụng cho hệ phi tuyến trên cơ sở tuyến tính hóa từng đoạn mô hình phi tuyến của hệ dọc theo trục thời gian và được dịch chuyển trên trục thời gian cùng với cửa sổ dự báo của bộ điều khiển dự báo. Cụ thể, luận án đã xây dựng được:
 - a) Thuật toán 2.1 để quan sát trạng thái hệ song tuyến.
 - b) Thuật toán 2.2 để quan sát trạng thái hệ phi tuyến.
 Khả năng áp dụng của hai thuật toán trên cũng đã được luận án mô phỏng với đối tượng song tuyến theo tín hiệu vào (2.14), (2.15) ở ví dụ 2.1 và ví dụ 2.2.
- 2) Phương pháp điều khiển dự báo phản hồi trạng thái hệ phi tuyến trên cơ sở sử dụng mô hình dự báo tuyến tính từng đoạn mà cụ thể là hai thuật toán:
 - a) Thuật toán 2.3 và Thuật toán 2.4 để điều khiển phản hồi trạng thái hệ song tuyến.
 - b) Thuật toán 2.5 hoặc Thuật toán 2.6 để điều khiển phản hồi trạng thái hệ phi tuyến.
- 3) Bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách trên cơ sở ghép chung bộ quan sát trạng thái Kalman và bộ điều khiển dự báo phản hồi trạng thái do

luận án đề xuất. Chi tiết các bước làm việc của bộ điều khiển này đã được luận án thể hiện ở Thuật toán 2.7.

- 4) Một điều kiện đủ để bộ điều khiển phản hồi đầu ra là Thuật toán 2.7 làm hệ ổn định ISS (ổn định thực tế).

CHƯƠNG 3 THỰC NGHIỆM KIỂM CHỨNG CHẤT LƯỢNG CỦA BỘ ĐIỀU KHIỂN DẪ ĐỀ XUẤT

Trong chương này, các thuật toán điều khiển dự báo được luận án đề xuất sẽ được ứng dụng vào thiết kế điều khiển các đối tượng phi tuyến. Trong thời gian gần đây thì các nghiên cứu ứng dụng MPC để áp dụng cho lớp các đối tượng động học biến đổi nhanh ngày càng trở nên phổ biến nhờ sự phát triển của nền tảng điều khiển số. Với sự đa dạng của lớp các đối tượng này, luận án lựa chọn trình bày chi tiết phần thiết kế điều khiển cho đối tượng con lắc ngược (là đối tượng có mô hình phi tuyến thường được sử dụng để kiểm chứng các thuật toán điều khiển mới) và phần thí nghiệm kiểm chứng trên mô hình thí nghiệm hệ thống con lắc ngược quay. Ngoài đối tượng con lắc ngược, các đối tượng công nghiệp khác (van công nghiệp, động cơ không đồng bộ 3 pha) cũng đã được thiết kế điều khiển thành công bằng các bộ điều khiển dự báo đề xuất và được tác giả thể hiện trong các công trình đã công bố (từ công trình số 3 đến công trình số 7) trên các tạp chí khoa học kỹ thuật.

3.1 Điều khiển đối tượng con lắc ngược

Xét hệ con lắc ngược có mô hình liên tục trong không gian trạng thái, theo tài liệu [11], như sau:

$$\frac{dx}{dt} = \begin{pmatrix} x_2 \\ \frac{(g \cos x_3 - l x_4^2) m_h \sin x_3}{m_c + m_h \sin^2 x_3} \\ x_4 \\ \frac{(g - l x_4^2 \cos x_3) m_h \sin x_3 + g m_c \sin x_3}{l m_c + l m_h \sin^2 x_3} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ -\frac{\cos x_3}{l m_c + l m_h \sin^2 x_3} \end{pmatrix} u \quad (3.1)$$

trong đó

$$\underline{x} = (x_1, x_2, x_3, x_4)^T, \quad x_1 = y_c, \quad x_2 = \frac{dy_c}{dt}, \quad x_3 = \varphi, \quad x_4 = \frac{d\varphi}{dt}$$

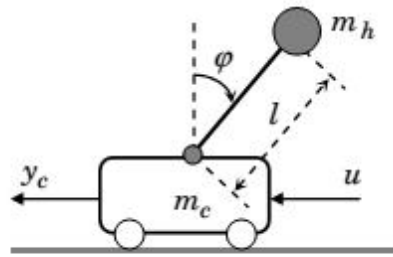
là các biến trạng thái, y_c là quãng đường dịch chuyển của xe dưới tác động lực đẩy u ở đầu vào, m_c là khối lượng của xe, m_h là khối lượng của quả cầu gắn với xe qua một thanh đỡ có chiều dài l . Thanh đỡ này xoay quanh được trục gá vào xe. Khi xe

dịch chuyển không đều, thanh đỡ quả cầu bị lệch so với phương thẳng đứng một góc lắc là φ . Tùy theo từng nhiệm vụ điều khiển được đặt ra khác nhau mà tín hiệu đầu ra sẽ được quy ước khác nhau. Chẳng hạn nếu nhiệm vụ điều khiển chỉ là ổn định góc lắc φ của vật thì tín hiệu đầu ra sẽ là:

$$\underline{y} = x_3 = (0, 0, 1, 0) \underline{x} \quad (3.2)$$

hoặc nếu nhiệm vụ điều khiển là vừa ổn định góc lắc φ , vừa ổn định vị trí y_c , thì tín hiệu đầu ra sẽ là:

$$\underline{y} = (y_1, y_2)^T = (x_1, x_3)^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \underline{x} \quad (3.3)$$



Hình 3.1: Cấu trúc vật lý hệ con lắc ngược.

Nhiệm vụ điều khiển được đặt ra bây giờ là xây dựng bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách sao cho tín hiệu ra của hệ bám theo được tín hiệu mẫu cho trước. Hình 3.1 mô tả cấu trúc vật lý hệ con lắc ngược.

Kết quả mô phỏng sau đây được thực hiện với các thông số cụ thể của mô hình hệ con lắc ngược như sau:

- Chiều dài thanh đỡ: $l = 0,27[m]$
- Khối lượng con lắc: $m_h = 0,1[kg]$
- Khối lượng của xe: $m_c = 1,2[kg]$
- Gia tốc trọng trường: $g = 9,8[m/s^2]$

3.1.1 Điều khiển dự báo phản hồi trạng thái

Chỉ điều khiển bám ổn định góc lắc

Khi nhiệm vụ điều khiển chỉ là bám ổn định góc lắc, thì tín hiệu đầu ra y của hệ sẽ được xác định theo công thức (3.2).

Mô hình trạng thái liên tục của hệ con lắc ngược (3.1) là một hệ phi tuyến affine SISO.

Đối với mô hình phi tuyến này của con lắc ngược, ta có thể thiết kế bộ điều khiển phản hồi trạng thái theo các bước của Thuật toán 2.6 hoặc tìm cách viết lại mô hình con lắc ngược sang dạng song tuyến theo trạng thái để sử dụng Thuật toán 2.4. Ở đây, luận án sẽ thực hiện lần lượt theo cả hai thuật toán để so sánh chất lượng, từ đó lựa chọn bộ điều khiển phản hồi trạng thái cho chất lượng điều khiển mong muốn và dễ cài đặt hơn.

Sử dụng trực tiếp mô hình phi tuyến để thiết kế bộ điều khiển dự báo phản hồi trạng thái đối tượng con lắc ngược theo Thuật toán 2.6

Để thiết kế điều khiển theo Thuật toán 2.6, ta sẽ phải sử dụng phép phân tích Taylor đối với mô hình (3.1) và các công thức tính đạo hàm riêng để xác định các ma trận A_k, B_k theo công thức (2.40).

Sử dụng các ký hiệu:

$$T_1 = (-g \cos x_3 + lx_4^2)m_h \sin x_3 = T_{11} \cdot T_{12}$$

$$T_{11} = -g \cos x_3 + lx_4^2$$

$$T_{12} = m_h \sin x_3$$

$$M_1 = m_c + m_h \sin^2 x_3$$

$$T_2 = (g - lx_4^2 \cos x_3)m_h \sin x_3 + gm_c \sin x_3 = T_{21} \cdot T_{22} + T_{23}$$

$$T_{21} = (g - lx_4^2 \cos x_3)$$

$$T_{22} = m_h \sin x_3$$

$$T_{23} = gm_c \sin x_3$$

$$M_2 = lm_c + lm_h \sin^2 x_3$$

Mô hình con lắc ngược (3.1),(3.2) viết lại thành:

$$\frac{d\underline{x}}{dt} = \begin{pmatrix} x_2 \\ \frac{T_1}{M_1} \\ x_4 \\ \frac{T_2}{M_2} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{M_1} \\ 0 \\ -\frac{\cos x_3}{M_2} \end{pmatrix} \underline{u} = \underline{f}(\underline{x}, \underline{u}) \quad (3.4)$$

$y = x_3$

Lượng tử hóa mô hình liên tục trên theo thời gian với chu kỳ trích mẫu T_a chọn trước, ta sẽ được mô hình không liên tục tương ứng dạng (2.49):

$$\begin{cases} \underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{u}_k) \\ \underline{y}_k = C(\underline{x}_k, \underline{u}_{k-1})\underline{x}_k \end{cases} \quad (3.5)$$

Phân tích vế phải của thành phần thứ nhất trong (3.5) thành chuỗi Taylor tại điểm quá khứ $\underline{x}_{k-1}, \underline{u}_{k-1}$ ta được:

$$\begin{aligned} \underline{f}(\underline{x}_k, \underline{u}_k) &\approx \underline{f}(\underline{x}_{k-1}, \underline{u}_{k-1}) + \underline{A}_k (\underline{x}_k - \underline{x}_{k-1}) + \underline{B}_k (\underline{u}_k - \underline{u}_{k-1}) \\ &= \underline{A}_k \underline{x}_k + \underline{B}_k \underline{u}_k + \underline{d}_k \end{aligned}$$

trong đó

$$\underline{A}_k = \left. \frac{\partial \underline{f}}{\partial \underline{x}} \right|_{\underline{x}_{k-1}, \underline{u}_{k-1}}, \quad \underline{B}_k = \left. \frac{\partial \underline{f}}{\partial \underline{u}} \right|_{\underline{x}_{k-1}, \underline{u}_{k-1}}$$

Và $\underline{d}_k = \underline{f}(\underline{x}_{k-1}, \underline{u}_{k-1}) - \underline{A}_k \underline{x}_{k-1} - \underline{B}_k \underline{u}_{k-1}$

Lấy đạo hàm riêng của $f(\underline{x}_k, \underline{u}_k)$ theo \underline{x}_k :

$$\frac{\partial \underline{f}}{\partial \underline{x}_k} = \begin{pmatrix} (0, 1, 0, 0) \\ \frac{M_1 \frac{\partial T_1}{\partial x_k} - T_1 \frac{\partial M_1}{\partial x_k}}{M_1 \cdot M_1} \\ (0, 0, 0, 1) \\ \frac{M_2 \frac{\partial T_2}{\partial x_k} - T_2 \frac{\partial M_2}{\partial x_k}}{M_2 \cdot M_2} \end{pmatrix} + \begin{pmatrix} 0 \\ -\frac{\frac{\partial M_1}{\partial x_k}}{M_1 \cdot M_1} \\ 0 \\ -\frac{M_2 (-\sin x_{3(k)}) (0, 0, 1, 0) - \cos x_{3(k)} \frac{\partial M_2}{\partial x_k}}{M_2 \cdot M_2} \end{pmatrix} \underline{u}$$

Suy ra:

$$\frac{\partial f}{\partial \underline{x}_k} = \begin{pmatrix} (0, 1, 0, 0) \\ \frac{M_1 \frac{\partial T_1}{\partial \underline{x}_k} - T_1 \frac{\partial M_1}{\partial \underline{x}_k} - \frac{\partial M_1}{\partial \underline{x}_k} u}{M_1 \cdot M_1} \\ (0, 0, 0, 1) \\ \frac{M_2 \frac{\partial T_2}{\partial \underline{x}_k} - T_2 \frac{\partial M_2}{\partial \underline{x}_k} - M_2 (-\sin x_{3(k)} (0, 0, 1, 0)) - \cos x_{3(k)} \frac{\partial M_2}{\partial \underline{x}_k}}{M_2 \cdot M_2} \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{pmatrix}$$

Vói:

$$\frac{\partial T_1}{\partial \underline{x}_k} = \frac{\partial(T_{11} T_{12})}{\partial \underline{x}_k} = \frac{\partial T_{11}}{\partial \underline{x}_k} T_{12} + T_{11} \frac{\partial T_{12}}{\partial \underline{x}_k}$$

$$\begin{aligned} \frac{\partial T_{11}}{\partial \underline{x}_k} &= \frac{\partial(-g \cos x_{4(k)} + l x_{4(k)}^2)}{\partial \underline{x}_k} \\ &= g \sin x_{4(k)} (0, 0, 1, 0) + 2l x_{4(k)} (0, 0, 1, 0) \end{aligned}$$

$$\frac{\partial T_{12}}{\partial \underline{x}_k} = \frac{\partial(m_h \sin x_{3(k)})}{\partial \underline{x}_k} = m_h \cos(x_{3(k)}) (0, 0, 1, 0)$$

$$\frac{\partial M_1}{\partial \underline{x}_k} = \frac{\partial(m_c + m_h \sin x_{3(k)} \sin x_{3(k)})}{\partial \underline{x}_k} = 2m_h \cos(x_{3(k)}) (0, 0, 1, 0)$$

$$\frac{\partial T_2}{\partial \underline{x}_k} = \frac{\partial(g - l x_{4(k)}^2 \cos(x_{3(k)}) m_h \sin(x_{3(k)}) + g m_c \sin(x_{3(k)}))}{\partial \underline{x}_k}$$

$$= \frac{\partial(T_{21} T_{22})}{\partial \underline{x}_k} + \frac{\partial T_{23}}{\partial \underline{x}_k} = \frac{\partial T_{21}}{\partial \underline{x}_k} T_{22} + T_{21} \frac{\partial T_{22}}{\partial \underline{x}_k} + \frac{\partial T_{23}}{\partial \underline{x}_k}$$

$$\begin{aligned} \frac{\partial T_{21}}{\partial \underline{x}_k} &= \frac{\partial(g - l(x_{4(k)}^2 \cos(x_{3(k)}))}{\partial \underline{x}_k} \\ &= -l(2x_{4(k)} \cos(x_{3(k)})) (0, 0, 0, 1) + x_{4(k)}^2 (-\sin x_{3(k)}) (0, 0, 1, 0) \end{aligned}$$

$$\frac{\partial T_{22}}{\partial \underline{x}_k} = \frac{\partial(m_h \sin(x_{3(k)}))}{\partial \underline{x}_k} = m_h \cos(x_{3(k)}) (0, 0, 1, 0)$$

$$\frac{\partial T_{23}}{\partial \underline{x}_k} = \frac{\partial(g m_c \sin(x_{3(k)}))}{\partial \underline{x}_k} = g m_c \cos(x_{3(k)}) (0, 0, 1, 0)$$

$$\frac{\partial M_2}{\partial \underline{x}_k} = \frac{\partial(l m_c + l m_h \sin(x_{3(k)}) \sin(x_{3(k)}))}{\partial \underline{x}_k} = 2l m_h \sin(x_{3(k)}) \cos(x_{3(k)}) (0, 0, 1, 0)$$

$$S_1 = (0, 1, 0, 0)$$

$$S_2 = \frac{M_1 \frac{\partial T_1}{\partial x_k} - T_1 \frac{\partial M_1}{\partial x_k} - \frac{\partial M_1}{\partial x_k}}{M_1 \cdot M_1} + \frac{\partial M_1}{\partial x_k} u_k$$

$$S_3 = (0, 0, 0, 1)$$

$$S_4 = \frac{M_2 \frac{\partial T_2}{\partial x_k} - T_2 \frac{\partial M_2}{\partial x_k} - M_2 \sin(x_{3(k)}) (0, 0, 1, 0) - \cos(x_{3(k)}) \frac{\partial M_2}{\partial x_k}}{M_2 \cdot M_2} u_k$$

Lấy đạo hàm riêng của $f(x_k, u_k)$ theo u_k : $\frac{\partial f}{\partial u_k} = \begin{pmatrix} 0 \\ 1 \\ M_1 \\ 0 \\ \cos(x_{3(k)}) \\ -M_2 \end{pmatrix}$

Suy ra, tại thời điểm $t = kT_a$ hiện tại, mô hình con lắc ngược biểu diễn bởi hệ phi tuyến không liên tục (3.5) là xấp xỉ được bởi :

$$\begin{cases} \underline{x}_{k+1} = A_k \underline{x}_k + B_k \underline{u}_k + \underline{d}_k \\ \underline{y}_k = C_k \underline{x}_k \end{cases} \quad (3.6)$$

trong đó:

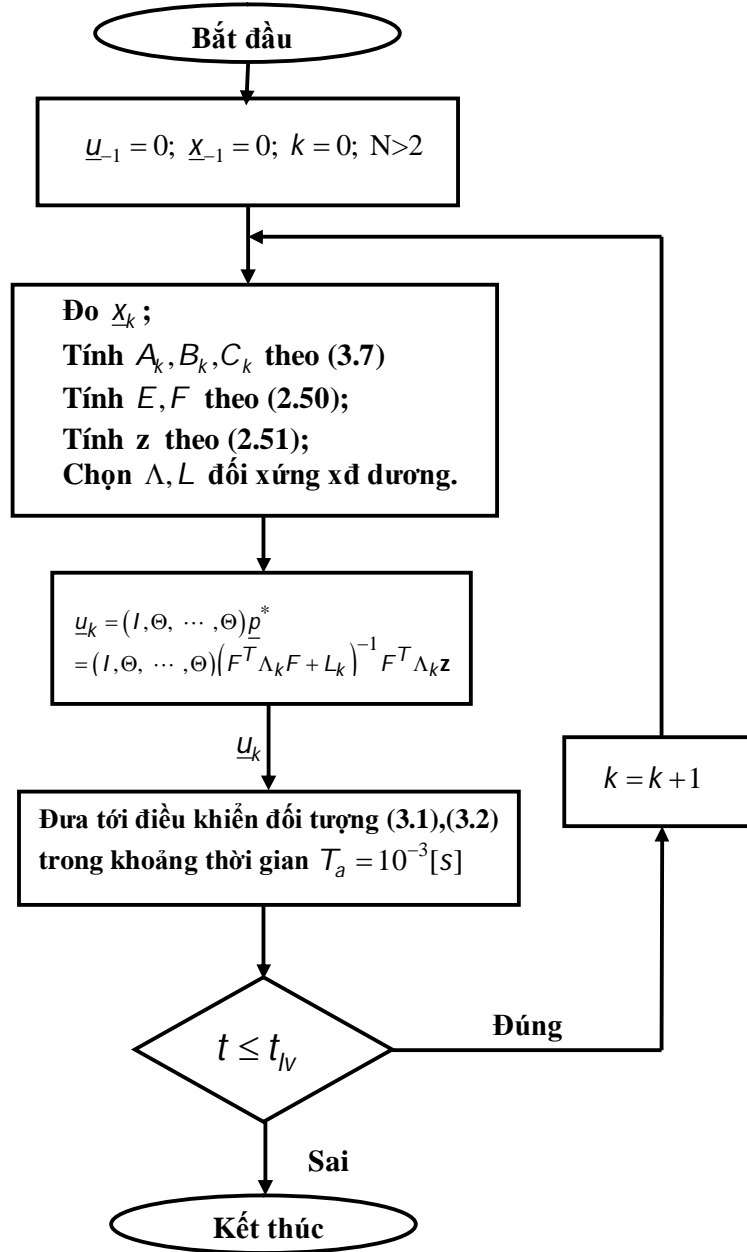
$$A_k = \left. \frac{\partial f}{\partial \underline{x}} \right|_{\underline{x}_{k-1}, \underline{u}_{k-1}} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{pmatrix} \bigg|_{\underline{x}_{k-1}, \underline{u}_{k-1}}, \quad B_k = \left. \frac{\partial f}{\partial \underline{u}} \right|_{\underline{x}_{k-1}, \underline{u}_{k-1}} = \begin{pmatrix} 0 \\ 1 \\ M_1 \\ 0 \\ \cos(x_{3(k)}) \\ -M_2 \end{pmatrix} \bigg|_{\underline{x}_{k-1}, \underline{u}_{k-1}} \quad (3.7)$$

$$\underline{x}_k = \underline{f}(\underline{x}_{k-1}, \underline{u}_{k-1}), C_k = C(\underline{x}_k, \underline{u}_{k-1}) = (0, 0, 1, 0)$$

$$\underline{d}_k = \underline{f}(\underline{x}_{k-1}, \underline{u}_{k-1}) - A_k \underline{x}_{k-1} - B_k \underline{u}_{k-1}$$

Có thể thấy, việc tuyến tính hóa từng đoạn của mô hình phi tuyến mà tại đó, giữa hai thời điểm chu kỳ trích mẫu các giá trị x_1, x_3 là hằng số, tức là giữa hai chu

kỳ trích mẫu luôn có $x_2 = \dot{x}_1 = 0; x_4 = \dot{x}_3 = 0$. Vì vậy, tại thời điểm giữa hai chu kỳ trích mẫu, $S_1 = S_2 = (0, 0, 0, 0)$.

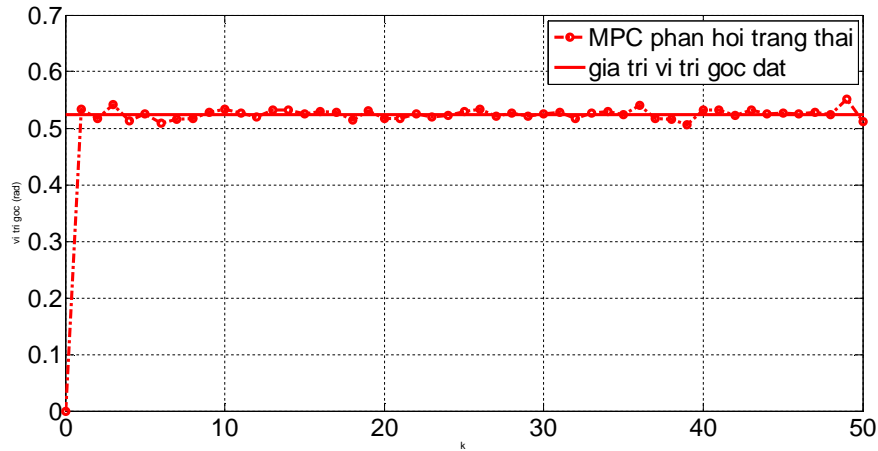


Hình 3.2: Lưu đồ thuật toán thiết kế bộ điều khiển DBPHTT bám theo giá trị đầu ra đặt cho con lắc ngược theo Thuật toán 2.6 với $t \leq t_{IV}$ là thời gian mô phỏng.

Sử dụng mô hình (3.6) để thiết kế điều khiển cho con lắc ngược theo các bước của Thuật toán 2.6. Chọn các giá trị đầu: $\underline{u}_{-1} = 0, \underline{x}_{-1} = 0, k = 0, N = 10$, đo trạng thái \underline{x}_k và từ đó xác định các ma trận A_k, B_k, C_k theo (2.40), (2.42). E, F theo (2.50), vector \mathbf{z} theo (2.51). Chọn các ma trận trọng số

$$\Lambda = \begin{bmatrix} N & 0 & \dots & 0 \\ 0 & N-1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}, L = I_{N \times N} \quad (3.8)$$

Các nhiễu ồn trắng được đưa tới đầu vào và đầu ra. Lưu đồ thuật toán thiết kế bộ điều khiển phản hồi trạng thái cho con lắc ngược theo Thuật toán 2.6 được trình bày như ở Hình 3.2. Thực hiện lần lượt các bước tính toán để xác định các ma trận A_k, B_k, C_k theo trạng thái của đối tượng trong mỗi bước tính rồi đưa tín hiệu điều khiển xác định được vào điều khiển đối tượng liên tục trong khoảng thời gian trích mẫu $T_a = 10^{-3}$ [s], giá trị vị trí góc đầu ra đặt giả sử là $\varphi = \pi/6$, ta thu được kết quả mô phỏng như trên Hình 3.3.



Hình 3.3: So sánh góc lắc thực có với góc lắc đặt trước khi sử dụng trực tiếp mô hình phi tuyến để thiết kế bộ điều khiển dự báo phản hồi trạng thái theo Thuật toán 2.6

Kết quả mô phỏng cho thấy giá trị góc lắc thực bám ổn định theo được giá trị đặt, ngoại trừ hiện tượng dao động nhỏ xung quanh giá trị đặt. Hiện tượng này sinh ra bởi góc đặt trước $\varphi = \pi/6$ không phải là một điểm cân bằng của hệ, nên để giữ hệ tại điểm đặt trước đó, tín hiệu điều khiển là khác không. Nó luôn thay đổi tạo ra hiện tượng dao động tương ứng xung quanh giá trị đặt. Biên độ dao động rất nhỏ, khoảng $0 - 0,02(rad) \approx (1-2)^0$.

Sử dụng mô hình song tuyến để thiết kế bộ điều khiển dự báo phản hồi trạng thái cho đối tượng con lắc ngược theo Thuật toán 2.4

Tiếp theo, với mô hình phi tuyến của con lắc ngược, luận án sẽ biến đổi sang dạng song tuyến theo trạng thái để có thể thiết kế điều khiển theo Thuật toán 2.4 .

Viết lại mô hình (3.1) về dạng song tuyến như sau:

$$\begin{aligned} \frac{d\underline{x}}{dt} &= \hat{A}(\underline{x})\underline{x} + \hat{B}(\underline{x})u \\ y &= \hat{C}(\underline{x})\underline{x} \end{aligned} \quad (3.9)$$

ta sẽ có nhiều phương án xác định $\hat{A}(\underline{x})$, $\hat{B}(\underline{x})$, $\hat{C}(\underline{x})$ từ (3.1), chẳng hạn như:

$$\hat{A}(\underline{x}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{(g \cos x_3 - lx_4^2)m_h \sin x_4}{(m_c + m_h \sin^2 x_3)x_4} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{[(g - lx_4^2 \cos x_3)m_h + gm_c] \sin x_3}{(lm_c + lm_h \sin^2 x_3)x_3} & 0 \end{pmatrix} \quad (3.10)$$

$$\hat{B}(\underline{x}) = \begin{pmatrix} 0 \\ 1 \\ \frac{1}{m_c + m_h \sin^2 x_3} \\ 0 \\ \frac{\cos x_3}{lm_c + lm_h \sin^2 x_3} \end{pmatrix} \quad \text{và} \quad \hat{C}(\underline{x}) = (0, 0, 1, 0) \quad (3.11)$$

là một phương án.

Lượng tử hóa mô hình liên tục trên theo thời gian với chu kỳ trích mẫu T_a chọn trước, ta sẽ được mô hình không liên tục tương ứng:

$$\begin{cases} \underline{x}_{k+1} = A(\underline{x}_k)\underline{x}_k + B(\underline{x}_k)\underline{u}_k \\ y = C(\underline{x}_k)\underline{x}_k \end{cases} \quad (3.12)$$

trong đó

$$A(\underline{x}_k) = I + T_a \widehat{A}(\underline{x}_k), \quad B(\underline{x}_k, \underline{u}_k) = T_a \widehat{B}(\underline{x}_k), \quad C(\underline{x}_k) = \widehat{C}(\underline{x}_k). \quad (3.13)$$

và đây cũng là mô hình rời rạc của hệ con lắc ngược được ta sử dụng cho việc thiết kế bộ điều khiển dự báo phản hồi trạng thái theo Thuật toán 2.4 sao cho đầu ra của nó là vị trí góc $y = \varphi$ bám theo giá trị đặt trước.

Tại thời điểm $t = kT_a$ hiện tại, mô hình con lắc ngược biểu diễn bởi hệ song tuyến không liên tục (3.12) là xấp xỉ được bởi :

$$H_k : \begin{cases} \underline{x}_{k+1} = A_k \underline{x}_k + B_k \underline{u}_k \\ \underline{y}_k = C_k \underline{x}_k \end{cases} \quad (3.14)$$

$$\text{với: } A_k = A(\underline{x}_k, k), \quad B_k = B(\underline{x}_k, k), \quad C_k = C(\underline{x}_k, k), \quad D_k = D(\underline{x}_k, k) \quad (3.15)$$

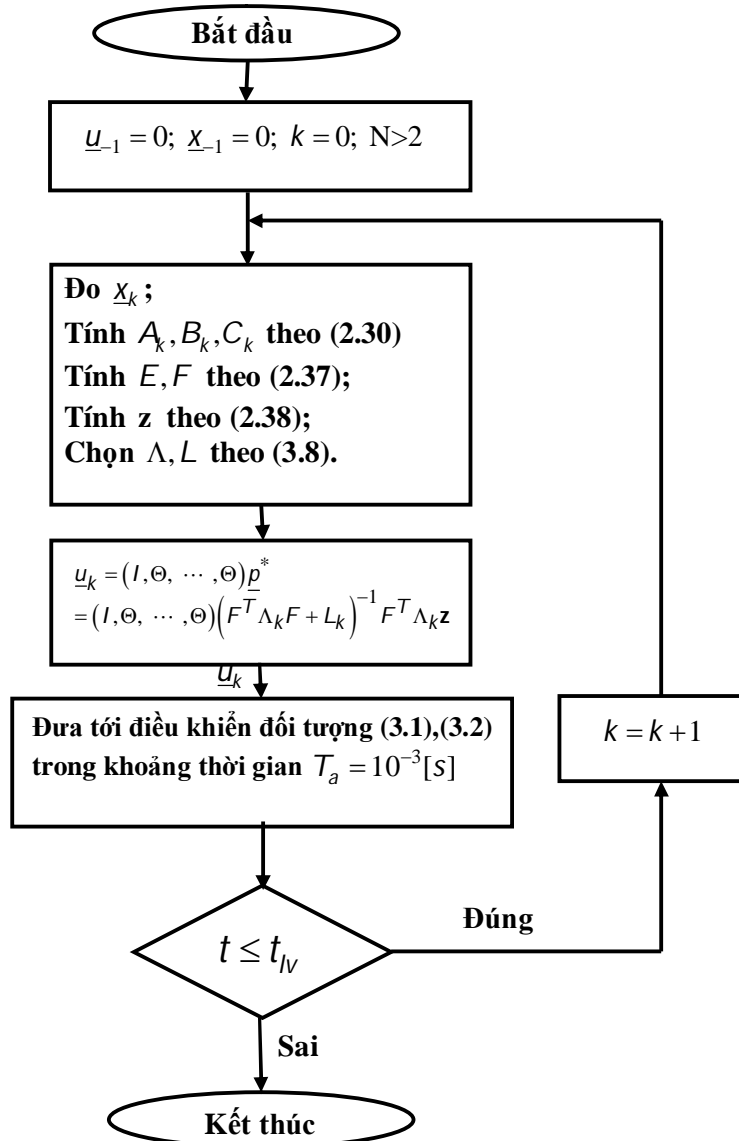
Tương tự, ta cũng có lưu đồ thuật toán thiết kế bộ điều khiển dự báo phản hồi trạng thái theo Thuật toán 2.4 cho đối tượng con lắc ngược như trên

Hình 3.4. Thực hiện lần lượt các bước tính toán để xác định các ma trận A_k , B_k , C_k theo trạng thái của đối tượng trong mỗi bước tính rồi đưa tín hiệu điều khiển xác định được vào điều khiển đối tượng liên tục trong khoảng thời gian trích mẫu $T_a = 10^{-3}$ [s], giá trị vị trí góc đầu ra đặt giả sử là $\varphi = \pi/6$, ta thu được kết quả mô phỏng như trên Hình 3.5.

Kết quả trên cho thấy giá trị góc lắc thực có chưa bám theo được giá trị đặt. Phân tích nguyên nhân có thể nằm ở công thức xấp xỉ mô hình song tuyến (3.12) với các giá trị ước lượng (3.13), nên ta có thể chỉnh sửa lại mô hình song tuyến (3.12) bằng cách thay (3.10) bởi:

$$\hat{A}(\underline{x}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(g \cos x_3 - l x_4^2) m_h \sin x_4}{(m_c + m_h \sin^2 x_3) x_4} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{[(g - l x_4^2 \cos x_3) m_h + g m_c] \sin x_3}{(l m_c + l m_h \sin^2 x_3) x_3} & 0 \end{pmatrix} \quad (3.16)$$

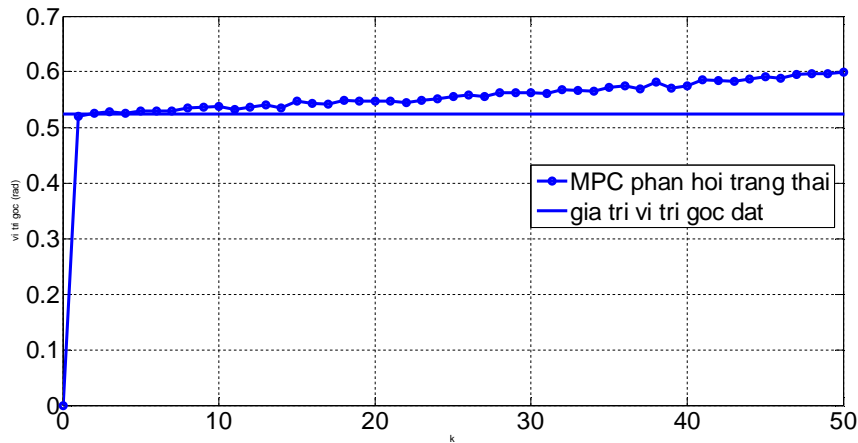
Việc chỉnh sửa này cũng xuất phát từ phân tích rằng do tuyến tính hóa từng đoạn của mô hình song tuyến mà tại đó, giữa hai thời điểm chu kỳ trích mẫu các giá trị x_1, x_3 là hằng số, tức là giữa hai chu kỳ trích mẫu luôn có $\dot{x}_1 = \dot{x}_3 = 0$.



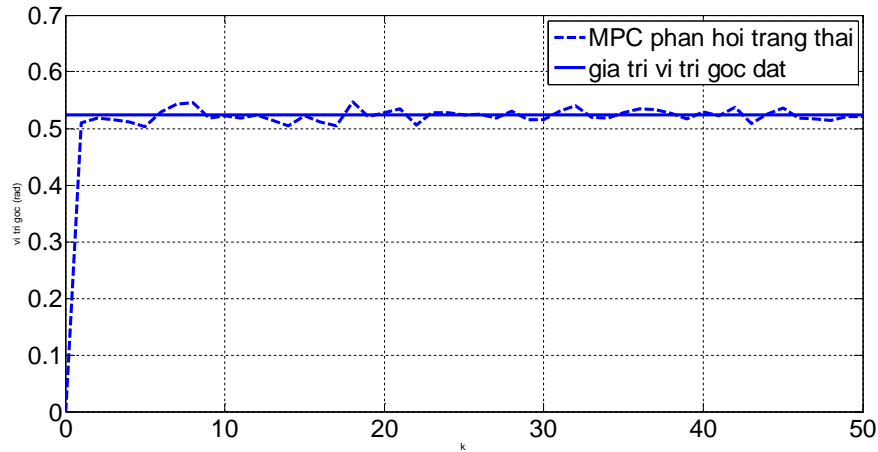
Hình 3.4: Lưu đồ thuật toán thiết kế bộ điều khiển DBPHTT bám theo giá trị đầu ra đặt cho con lắc ngược theo Thuật toán 2.4 với $t \leq t_N$ là thời gian mô phỏng

Hình 3.6 là kết quả mô phỏng thu được với ma trận $\hat{A}(x)$ lấy từ (3.16), các ma trận $\hat{B}(x)$, $\hat{C}(x)$ được giữ nguyên theo công thức (3.11). Nó cho thấy giá trị góc lắc thực bây giờ bám ổn định theo được giá trị đặt, ngoại trừ hiện tượng dao động nhỏ xung quanh giá trị đặt. Hiện tượng này sinh ra bởi góc đặt trước $\varphi = \pi/6$ không phải là một điểm cân bằng của hệ, nên để giữ hệ tại điểm đặt trước đó, tín hiệu điều khiển là khác không. Nó luôn thay đổi tạo ra hiện tượng dao động tương ứng xung quanh giá trị đặt. Đặc biệt hiện tượng này sẽ còn nhận biết được rõ hơn nữa khi giá trị đặt trước mà hệ cần bám theo vừa là góc, vừa là vị trí của xe như được mô tả ở Hình 3.8 và Hình 3.9.

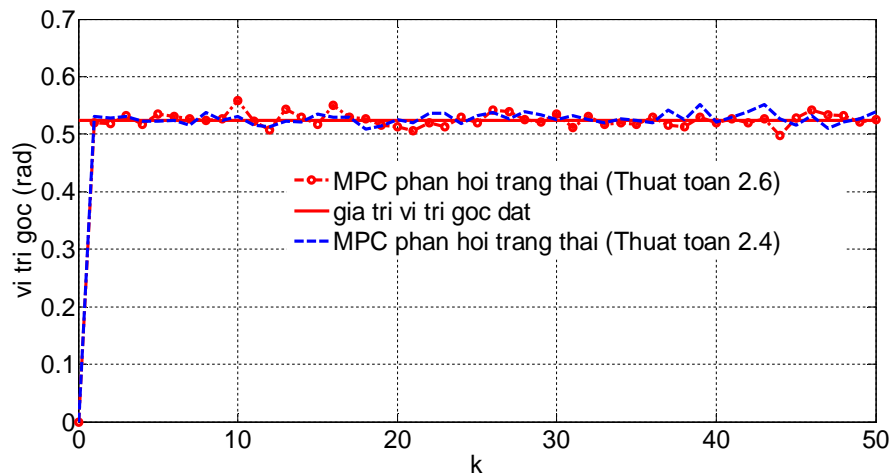
Trên Hình 3.7 là so sánh kết quả điều khiển vị trí góc theo giá trị đặt khi sử dụng 2 bộ điều khiển theo Thuật toán 2.4 (sử dụng mô hình song tuyến) và Thuật toán 2.6 (sử dụng trực tiếp mô hình phi tuyến). Đường nét đứt là vị trí góc khi sử dụng bộ điều khiển theo Thuật toán 2.4, đường nét đứt có chấm tròn là vị trí góc khi sử dụng bộ điều khiển theo Thuật toán 2.6 .



Hình 3.5: So sánh góc lắc thực có với góc lắc đặt trước khi sử dụng mô hình song tuyến (3.10).



Hình 3.6: So sánh góc lắc thực có với góc lắc đặt trước khi sử dụng mô hình song tuyến (3.16).



Hình 3.7: So sánh kết quả điều khiển vị trí góc theo giá trị đặt khi sử dụng 2 bộ điều khiển theo Thuật toán 2.4 và Thuật toán 2.6

So sánh cho thấy, trong cả hai trường hợp, góc lắc đều bám theo giá trị đặt đáp ứng mục tiêu điều khiển, biên độ sai lệch trong cả hai trường hợp là rất nhỏ ($(1-2)^0$). Tại một số thời điểm thì biên độ sai lệch của góc lắc khi sử dụng bộ điều khiển theo Thuật toán 2.6 có lớn hơn một chút so với bộ điều khiển theo Thuật toán 2.4. Nguyên nhân có thể được giải thích rằng, khi ta sử dụng trực tiếp mô hình phi

tuyến theo Thuật toán 2.6 thì trong bước tuyến tính hóa từng đoạn đã có sử dụng các công thức xấp xỉ, vì vậy dẫn đến sai lệch mô hình, nhưng sai lệch này cũng là rất nhỏ, không đáng kể.

Nhận xét:

Qua việc thiết kế điều khiển và so sánh chất lượng điều khiển với 2 bộ điều khiển theo Thuật toán 2.4 (sử dụng mô hình song tuyến) và Thuật toán 2.6 (sử dụng mô hình phi tuyến), có thể thấy ngay rằng, với các đối tượng phi tuyến có mô hình có thể chuyển đổi về dạng song tuyến thì ta nên sử dụng Thuật toán 2.4 để thiết kế điều khiển do việc cài đặt theo Thuật toán 2.4 là đơn giản hơn và khối lượng tính toán ít hơn so với Thuật toán 2.6, đồng thời chất lượng điều khiển cũng tốt hơn so với việc sử dụng Thuật toán 2.6, do trong Thuật toán 2.6 có sử dụng công thức xấp xỉ có thể dẫn đến sai lệch mô hình. Chính vì vậy, với những đối tượng phi tuyến có mô hình biến đổi được về dạng song tuyến thì ta nên sử dụng mô hình song tuyến để thiết kế điều khiển theo Thuật toán 2.4.

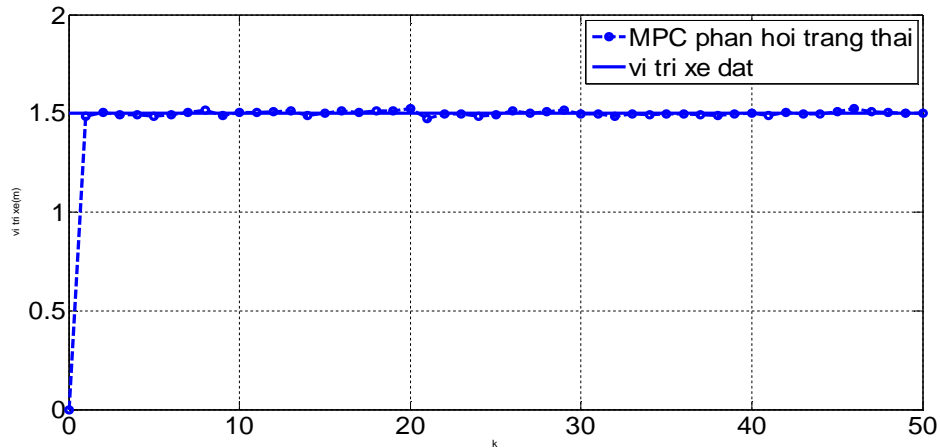
Toàn bộ chi tiết phần mã nguồn chương trình cài đặt trên MatLab để điều khiển dự báo phản hồi trạng thái hệ con lắc ngược theo Thuật toán 2.4 và Thuật toán 2.6 được trình bày ở phần phụ lục **P1**.

Điều khiển bám ổn định góc lắc và vị trí

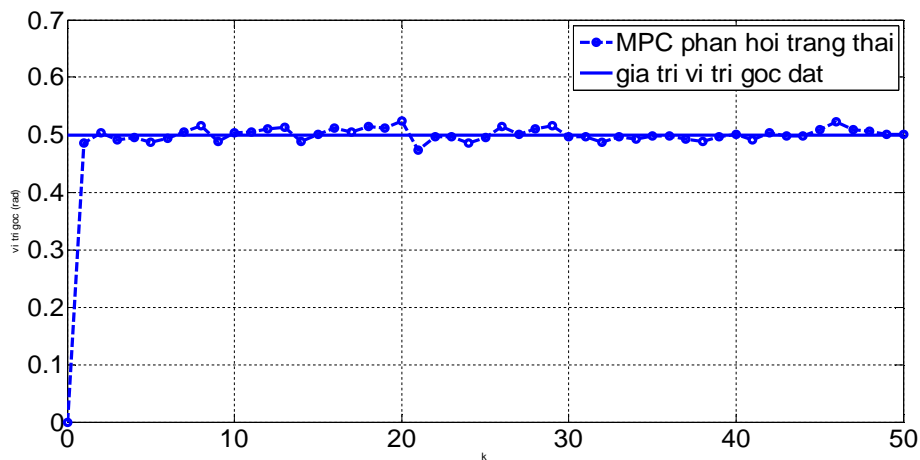
Khi nhiệm vụ điều khiển vừa là bám ổn định góc lắc, vừa là bám ổn định vị trí thì hai tín hiệu đầu ra y của hệ sẽ được xác định theo công thức (3.3).

Lại sử dụng mô hình song tuyến (3.9) nhưng bây giờ với hai đầu ra (3.3) và ma trận $\hat{A}(x)$ theo (3.16), hai ma trận $\hat{B}(x)$, $\hat{C}(x)$ còn lại theo (3.11), cùng với chu kỳ trích mẫu $T_a = 10^{-3}s$ giống như ở trường hợp một đầu ra, ta thu được kết quả cho ở Hình 3.8 và Hình 3.9. Hai hình biểu diễn kết quả mô phỏng này cho thấy chất lượng bám ổn định tốt của bộ điều khiển dự báo phản hồi trạng thái (Thuật toán 2.4), phù hợp với nhận định lý thuyết.

Toàn bộ chi tiết phần mã nguồn chương trình cài đặt trên MatLab để điều khiển dự báo phản hồi trạng thái hệ con lắc ngược theo Thuật toán 2.4 được trình bày ở phần phụ lục **P2**.



Hình 3.8: So sánh vị trí thực có với vị trí đặt trước



Hình 3.9: So sánh góc lắc thực có với góc lắc đặt trước

3.1.2 Quan sát trạng thái với lọc Kalman mở rộng

Tiếp theo, ta sẽ thiết kế bộ quan sát Kalman mở rộng loại 3 (EKF loại 3) theo các bước của Thuật toán 1.9 cho đối tượng con lắc ngược có mô hình trạng thái liên tục (3.1).

Vẫn tiếp tục chuyển mô hình (3.1) về dạng song tuyến theo trạng thái (3.9) và dạng không liên tục là (3.12), với hai đầu ra (3.3) và:

$$\hat{A}(x) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{(g \cos x_3 - lx_4^2)m_h \sin x_4}{(m_c + m_h \sin^2 x_3)x_2} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{(g - lx_4^2 \cos x_3)m_h \sin x_3 + gm_c \sin x_3}{(lm_c + lm_h \sin^2 x_3)x_4} \end{pmatrix}$$

để sử dụng vào việc thiết kế bộ quan sát trạng thái theo Thuật toán 1.9.

Với mô hình dạng không liên tục này, ta thực hiện tính G_k và F_{k-1} lần lượt theo các công thức (1.45) và (1.47) của EKF loại 3 là:

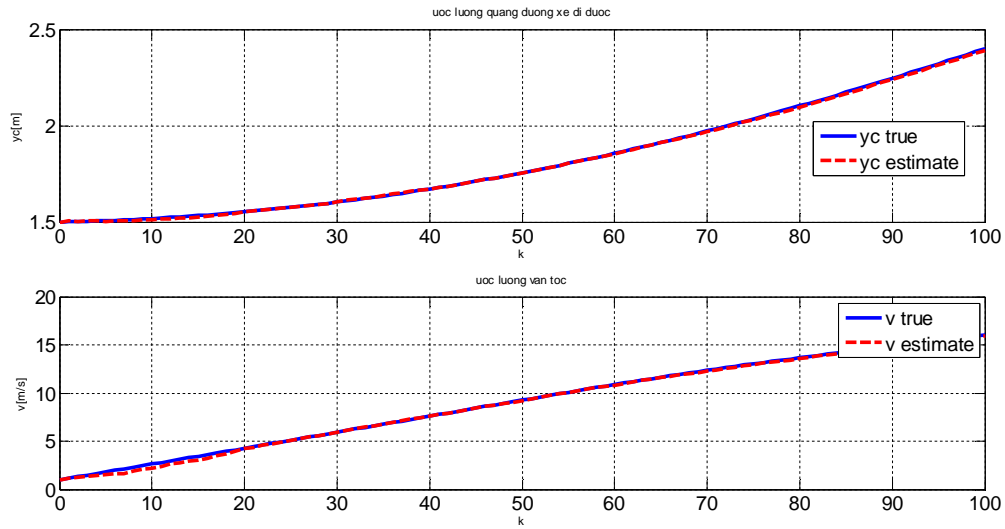
$$G_k = \hat{C}(x_k) \text{ và } F_{k-1} = \hat{A}(x_{k-1}) \quad (3.17)$$

và trạng thái ước lượng ban đầu giả sử được chọn là:

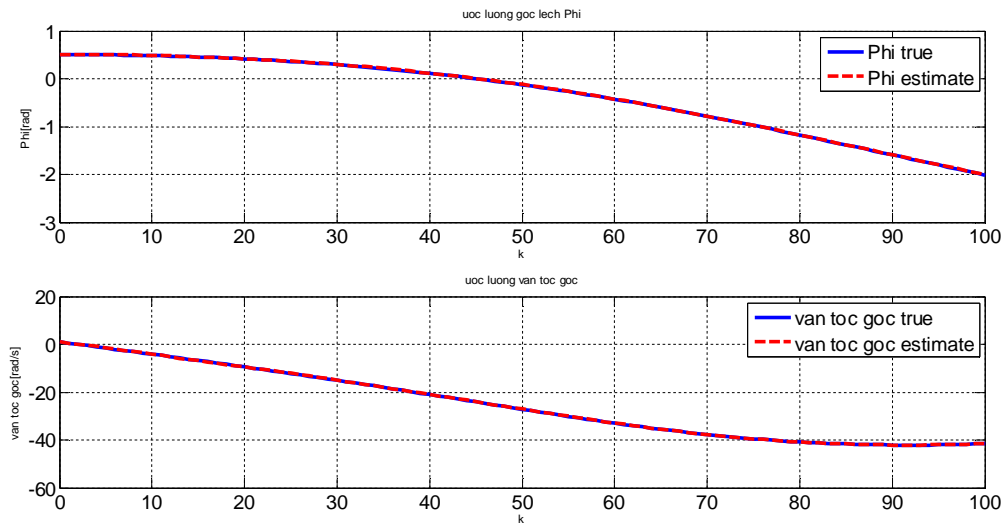
$$x_{es0} = (1, 2, 2\pi, 2)^T$$

Lần lượt trên các Hình 3.10 và Hình 3.11 là kết quả mô phỏng thu được, so sánh giữa các biến trạng thái quan sát được bằng EKF loại 3 (đường nét đứt) và biến trạng thái thực (đường nét liền) của hệ con lắc ngược (3.1). Kết quả cho thấy, dưới sự tác động của nhiễu ồn trắng cả bên trong hệ thống và đầu ra thì các biến trạng thái quan sát được gồm: $x_1 = y_c$, $x_2 = \frac{dy_c}{dt}$, $x_3 = \varphi$, $x_4 = \frac{d\varphi}{dt}$ đều trùng khớp với các giá trị của biến trạng thái thực. Điều này cho thấy bộ lọc Kalman mở rộng loại 3 làm việc tốt để quan sát trạng thái cho đối tượng con lắc ngược có mô hình phi tuyến.

Toàn bộ chi tiết phần mã nguồn chương trình cài đặt trên MatLab để quan sát trạng thái hệ con lắc ngược theo Thuật toán 1.9 được trình bày ở phần phụ lục **P2**.

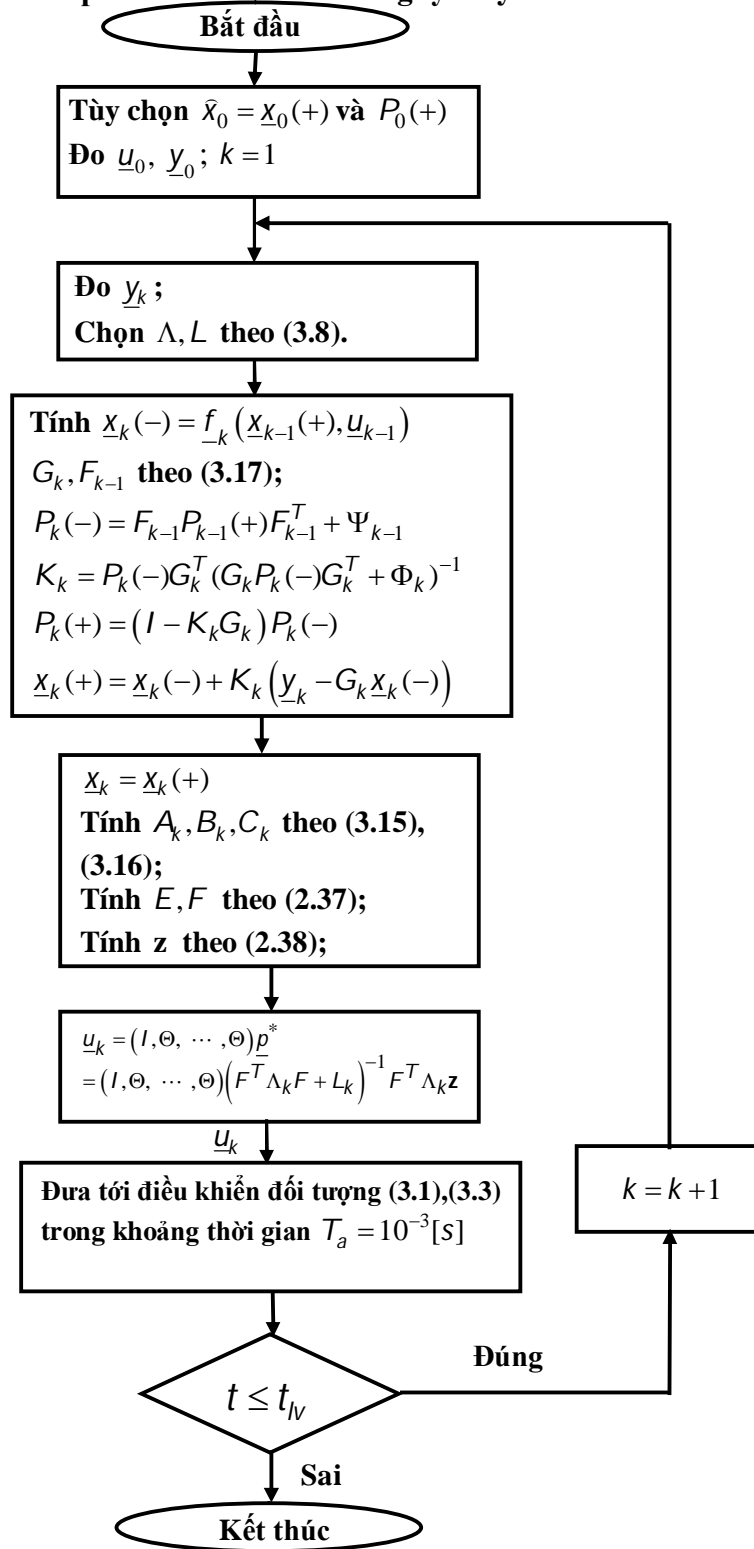


Hình 3.10: Giá trị trạng thái x_3, x_4 quan sát được so sánh với giá trị thực khi có nhiễu hệ thống và nhiễu đầu ra là ồn trắng



Hình 3.11: Giá trị trạng thái x_1, x_2 quan sát được so sánh với giá trị thực khi có nhiễu hệ thống và nhiễu đầu ra là ồn trắng

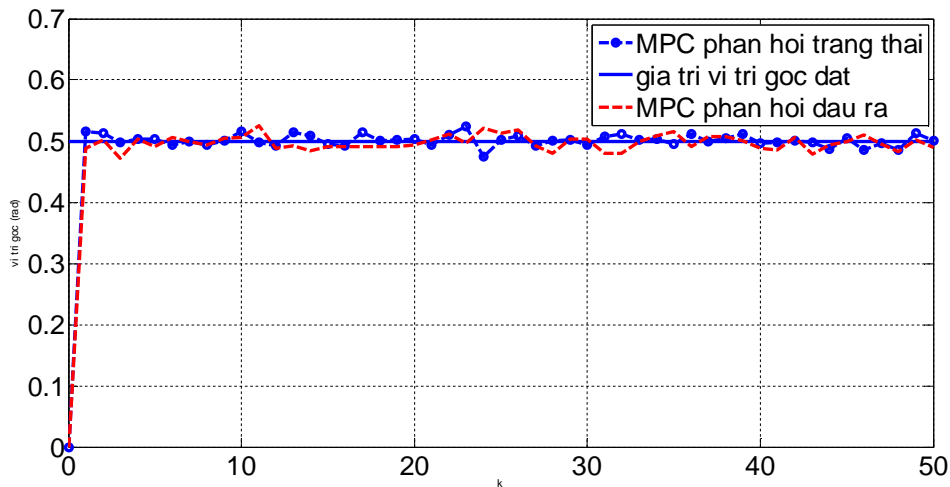
3.1.3 Điều khiển phản hồi đầu ra theo nguyên lý tách



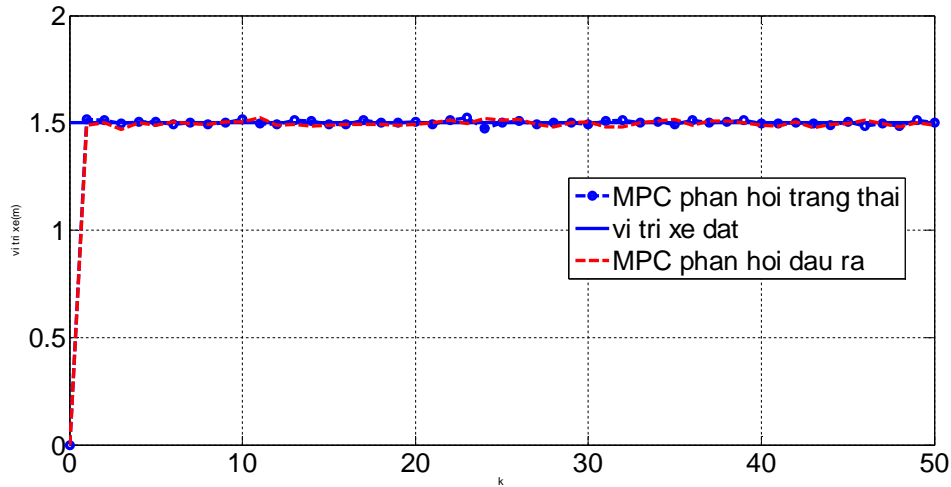
Hình 3.12: Lưu đồ thuật toán thiết kế bộ ĐKDB PHDR bám theo giá trị đầu ra đặt cho đối tượng con lắc ngược theo Thuật toán 2.7 với $t \leq t_{IV}$ là thời gian mô phỏng

Tiếp theo, ta sẽ thiết kế bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho đối tượng con lắc ngược có mô hình phi tuyến (3.1) bằng việc kết hợp giữa bộ lọc EKF loại 3 và bộ điều khiển dự báo phản hồi trạng thái cho đối tượng này theo phiên bản của Thuật toán 2.7 xây dựng cho hệ song tuyến.

Lựa chọn các điều kiện đầu, cửa sổ dự báo, bộ các thông số của hệ con lắc ngược, các tín hiệu nhiễu giống như đã mô phỏng với bộ điều khiển dự báo phản hồi trạng thái ở mục 3.1.1 nhưng tín hiệu trạng thái đưa vào để tính toán tín hiệu điều khiển là các trạng thái quan sát được bằng bộ quan sát EKF loại 3 như đã thiết kế ở mục 3.1.2. Trong lưu đồ thuật toán hình 3.12 lúc này có thêm bước tính toán ước lượng trạng thái để thiết kế bộ điều khiển. Kết quả so sánh tín hiệu đầu ra (là vị trí góc của con lắc và vị trí xe) giữa hai bộ điều khiển phản hồi trạng thái và phản hồi đầu ra thể hiện trên Hình 3.13 và Hình 3.14.



Hình 3.13: Góc lắc thực $y_2 = \varphi$ so sánh với góc lắc đặt thu được nhờ bộ điều khiển dự báo phản hồi đầu ra khi có nhiễu hệ thống và nhiễu đầu ra ở dạng ồn trắng



Hình 3.14: Vị trí xe thực có $y_1 = y_c$ so sánh với vị trí thu được nhờ bộ điều khiển dự báo phản hồi đầu ra khi có nhiễu hệ thống và nhiễu đầu ra ở dạng ồn trắng

Trên Hình 3.13, đường nét liền thể hiện giá trị đầu ra đặt (vị trí góc của con lắc ngược), đường nét đứt có chấm tròn là giá trị đầu ra khi điều khiển đối tượng bằng bộ điều khiển dự báo phản hồi trạng thái và đường nét đứt đậm là giá trị đầu ra khi điều khiển đối tượng bằng bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách. Tương tự, Hình 3.14 biểu diễn giá trị vị trí thực có của xe được so sánh với giá trị vị trí đặt trước.

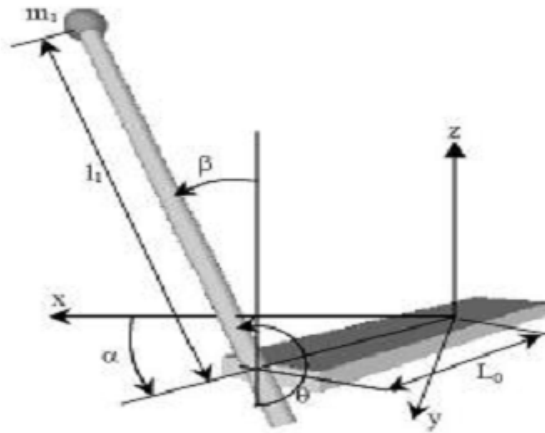
Có thể thấy, vị trí góc lắc thực có và vị trí xe thực có của đối tượng khi điều khiển bằng 2 bộ điều khiển đều luôn bám theo được với giá trị đầu ra đặt và ổn định trong suốt thời gian điều khiển. Khi ta tăng giá trị nhiễu ảnh hưởng đến đối tượng thì góc lắc và vị trí xe vẫn bám tốt theo giá trị góc lắc và vị trí xe đặt. Tín hiệu đầu ra khi sử dụng bộ điều khiển phản hồi đầu ra có sai lệch ít so với bộ điều khiển phản hồi trạng thái, song vẫn bám tốt theo giá trị đặt, ngoại trừ có hiện tượng dao động nhỏ xung quanh giá trị đặt sinh ra bởi những giá trị đặt trước này không phải là điểm cân bằng của hệ.

Toàn bộ chi tiết phần mã nguồn chương trình cài đặt trên MatLab để điều khiển dự báo phản hồi đầu ra hệ con lắc ngược theo nguyên lý tách với Thuật toán 2.7 được trình bày ở phần phụ lục **P3**.

3.2 Kiểm chứng chất lượng bộ điều khiển dự báo phản hồi đầu ra trên mô hình thí nghiệm con lắc ngược quay

Để kiểm chứng được chất lượng bám bền vững với nhiễu Gauss của bộ điều khiển dự báo phản hồi đầu ra (thuật toán 2.7) trong môi trường thực tế, sau đây luận án sẽ tiến hành thí nghiệm trên con lắc ngược quay, là mô hình thí nghiệm đã có sẵn tại phòng thí nghiệm Đo lường và Điều khiển, Khoa Điện Tử của Trường Đại học KTCN Thái Nguyên.

3.2.1 Mô hình toán của đối tượng con lắc ngược quay



Hình 3.15: Mô hình động học con lắc ngược quay

Hình 3.15 là mô hình động học của con lắc ngược quay. Hệ thống này bao gồm:

- Cánh tay (Arm): gắn vào trục động cơ một chiều (DC motor) quay quanh trục thẳng đứng.

- Con lắc (Pendulum): là khớp quay tự do gắn vào trục của cảm biến góc quay (Potentiometer) ở đầu cánh tay, tự do trong mặt phẳng vuông góc với cánh tay. Con lắc không ổn định, nó luôn đổ xuống, trừ khi có lực tác động thích hợp vào cánh tay. Bài toán đặt ra là điều khiển cánh tay để lắc con lắc lên (Swing up) và giữ con lắc cân bằng ở vị trí thẳng đứng.

Từ mô hình động học này, mô hình toán của hệ được xây dựng như sau [46,64]:

Các phương trình Lagrange mô tả chuyển động của con lắc ngược quay [41]:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\Theta}_i} \right) - \frac{\partial L}{\partial \Theta_i} = Q_i \quad (3.18)$$

$$L = T - V$$

trong đó:

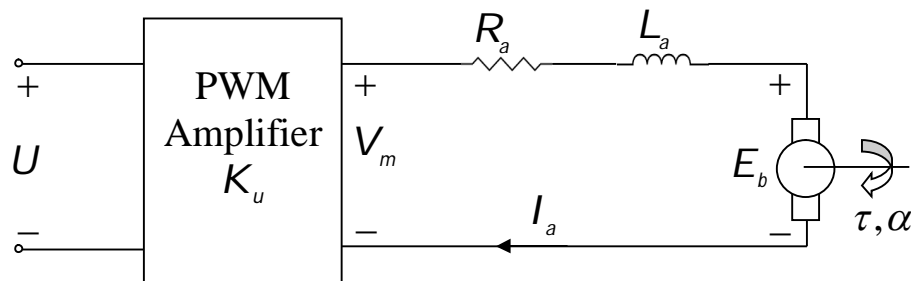
T : là tổng động năng của phần thân quay.

V : là tổng thế năng của phần thân quay.

Q_i : là tổng mômen xoắn theo trục i

Θ_i : là vị trí góc theo trục i

Các phương trình mô tả động cơ một chiều:



Hình 3.16: Sơ đồ tương đương động cơ điện 1 chiều và bộ điều chế độ rộng xung (bộ khuếch đại xung PWM)

Động cơ một chiều sử dụng trong hệ con lắc ngược quay để tạo ra mômen xoắn cho hệ thống. Mô men xoắn của động cơ theo [35] được tính toán như sau:

Điện áp đầu ra của bộ khuếch đại:

$$V_m = K_u \cdot u \quad (3.19)$$

với K_u là hệ số khuếch đại của bộ PWM.

Theo định luật Kirchoff về điện áp,

$$V_m = I_a R_a + L_a \frac{dI_a}{dt} + E_b \quad (3.20)$$

với:

- $I_a (A)$: là dòng điện cuộn dây phản ứng,
- $R_a (\Omega)$: là điện trở cuộn dây phản ứng,
- $L_a (H)$: là điện cảm cuộn dây phản ứng,
- $E_b (V)$: sức điện động cảm ứng của động cơ. E_b tỷ lệ thuận với sự thay đổi từ thông nó tỷ lệ với vận tốc góc của động cơ.

$$E_b = K_b \frac{d\alpha}{dt} = K_b \dot{\alpha} \quad (3.21)$$

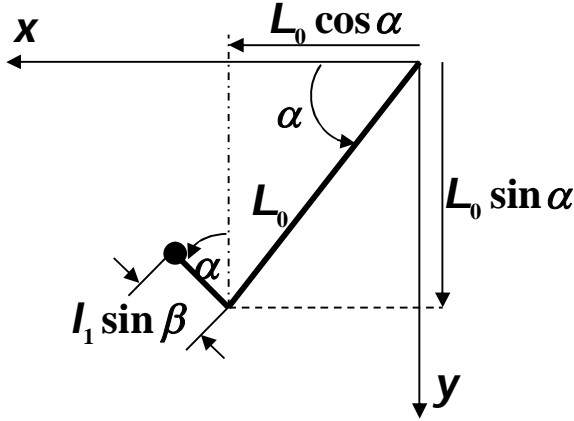
Mô men xoắn gây ra bởi động cơ, τ_m tỷ lệ với dòng điện phản ứng. Giả thiết rằng, ảnh hưởng của điện cảm cuộn dây, L_a , là không đáng kể, mô men xoắn có thể được tính:

$$\tau_m = K_t I_a = K_t \frac{(V_m - E_b)}{R_a} = \frac{K_t K_u}{R_a} u - \frac{K_t K_b}{R_a} \dot{\alpha} \quad (3.22)$$

Mô hình động học của con lắc ngược (khi con lắc ở vị trí thẳng đứng hướng lên trên)

Gắn hệ con lắc ngược lên hệ trục tọa độ như trong Hình 3.15, động năng của cánh tay, T_0 và thế năng V_0 có thể được viết như sau:

$$\begin{aligned} T_0 &= \frac{1}{2} J_0 \dot{\alpha}^2 \\ V_0 &= 0 \end{aligned} \quad (3.23)$$



Hình 3.17: Hình chiếu bằng của hệ con lắc ngược quay

Hình chiếu bằng của hệ con lắc ngược quay được thể hiện trên Hình 3.17. Đối với phần con lắc, được phân tích động học như sau:

$$\begin{aligned}x_1 &= L_0 \cos \alpha + l_1 \sin \alpha \sin \beta \\y_1 &= L_0 \sin \alpha - l_1 \cos \alpha \sin \beta \\z_1 &= l_1 \cos \beta\end{aligned}\quad (3.24)$$

Từ (3.24), sẽ có:

$$\begin{aligned}\dot{x}_1 &= -L_0 \dot{\alpha} \sin \alpha + l_1 \dot{\alpha} \cos \alpha \sin \beta + l_1 \dot{\beta} \sin \alpha \cos \beta \\ \dot{y}_1 &= L_0 \dot{\alpha} \cos \alpha + l_1 \dot{\alpha} \sin \alpha \sin \beta - l_1 \dot{\beta} \cos \alpha \cos \beta \\ \dot{z}_1 &= -l_1 \dot{\beta} \sin \beta\end{aligned}\quad (3.25)$$

và từ đó:

$$\begin{aligned}\dot{x}_1^2 &= L_0^2 \dot{\alpha}^2 \sin^2 \alpha + (l_1 \dot{\alpha} \cos \alpha \sin \beta + l_1 \dot{\beta} \sin \alpha \cos \beta)^2 \\ &\quad - 2L_0 \dot{\alpha} \sin \alpha (l_1 \dot{\alpha} \cos \alpha \sin \beta + l_1 \dot{\beta} \sin \alpha \cos \beta) \\ &= L_0^2 \dot{\alpha}^2 \sin^2 \alpha + l_1^2 \dot{\alpha}^2 \cos^2 \alpha \sin^2 \beta + l_1^2 \dot{\beta}^2 \sin^2 \alpha \cos^2 \beta \\ &\quad + 2l_1^2 \dot{\alpha} \dot{\beta} \sin \alpha \cos \alpha \sin \beta \cos \beta - 2L_0 l_1 \dot{\alpha}^2 \sin \alpha \cos \alpha \sin \beta - 2L_0 l_1 \dot{\alpha} \dot{\beta} \sin^2 \alpha \cos \beta\end{aligned}\quad (3.26)$$

$$\begin{aligned}\dot{y}_1^2 &= L_0^2 \dot{\alpha}^2 \cos^2 \alpha + (l_1 \dot{\alpha} \sin \alpha \sin \beta - l_1 \dot{\beta} \cos \alpha \cos \beta)^2 \\ &\quad + 2L_0 \dot{\alpha} \cos \alpha (l_1 \dot{\alpha} \sin \alpha \sin \beta - l_1 \dot{\beta} \cos \alpha \cos \beta) \\ &= L_0^2 \dot{\alpha}^2 \cos^2 \alpha + l_1^2 \dot{\alpha}^2 \sin^2 \alpha \sin^2 \beta + l_1^2 \dot{\beta}^2 \cos^2 \alpha \cos^2 \beta \\ &\quad - 2l_1^2 \dot{\alpha} \dot{\beta} \sin \alpha \cos \alpha \sin \beta \cos \beta + 2L_0 l_1 \dot{\alpha}^2 \sin \alpha \cos \alpha \sin \beta - 2L_0 l_1 \dot{\alpha} \dot{\beta} \cos^2 \alpha \cos \beta\end{aligned}\quad (3.27)$$

$$\dot{z}_1^2 = l_1^2 \dot{\beta}^2 \sin^2 \beta \quad (3.28)$$

Tổng bình phương độ lớn vận tốc của quả lắc sẽ là:

$$\begin{aligned} \dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2 &= L_0 \dot{\alpha}^2 + l_1^2 \dot{\alpha}^2 \sin^2 \beta + l_1^2 \dot{\beta}^2 \cos^2 \beta - 2L_0 l_1 \dot{\alpha} \dot{\beta} \cos \beta + l_1^2 \dot{\beta}^2 \sin^2 \beta \\ &= L_0^2 \dot{\alpha}^2 + l_1^2 \dot{\alpha}^2 \sin^2 \beta + l_1^2 \dot{\beta}^2 - 2L_0 l_1 \dot{\alpha} \dot{\beta} \cos \beta \end{aligned} \quad (3.29)$$

Động năng của quả lắc T_1 và thế năng V_1 được tính:

$$\begin{aligned} T_1 &= \frac{1}{2} J_1 \dot{\beta}^2 + \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2) \\ &= \frac{1}{2} J_1 \dot{\beta}^2 + \frac{1}{2} m_1 L_0^2 \dot{\alpha}^2 + \frac{1}{2} m_1 l_1^2 \dot{\alpha}^2 \sin^2 \beta + \frac{1}{2} m_1 l_1^2 \dot{\beta}^2 - m_1 L_0 l_1 \dot{\alpha} \dot{\beta} \cos \beta \\ &= \frac{1}{2} m_1 L_0^2 \dot{\alpha}^2 + \frac{1}{2} (J_1 + m_1 l_1^2) \dot{\beta}^2 + \frac{1}{2} m_1 l_1^2 \dot{\alpha}^2 \sin^2 \beta - m_1 L_0 l_1 \dot{\alpha} \dot{\beta} \cos \beta \end{aligned} \quad (3.30)$$

$$V_1 = m_1 g z_1 = m_1 g l_1 \cos \beta \quad (3.31)$$

Phương trình cân bằng năng lượng của hệ con lắc ngược quay đưa ra theo Lagrange, L được xây dựng như sau:

$$\begin{aligned} L_0 &= T_0 - V_0 \\ L_1 &= T_1 - V_1 \\ L &= L_0 + L_1 = T_0 + T_1 - V_0 - V_1 \\ &= \frac{1}{2} J_0 \dot{\alpha}^2 + \frac{1}{2} m_1 L_0^2 \dot{\alpha}^2 + \frac{1}{2} (J_1 + m_1 l_1^2) \dot{\beta}^2 + \frac{1}{2} m_1 l_1^2 \dot{\alpha}^2 \sin^2 \beta - m_1 L_0 l_1 \dot{\alpha} \dot{\beta} \cos \beta - m_1 g l_1 \cos \beta \\ &= \frac{1}{2} (J_0 + m_1 L_0^2) \dot{\alpha}^2 + \frac{1}{2} (J_1 + m_1 l_1^2) \dot{\beta}^2 + \frac{1}{2} m_1 l_1^2 \dot{\alpha}^2 \sin^2 \beta - m_1 L_0 l_1 \dot{\alpha} \dot{\beta} \cos \beta - m_1 g l_1 \cos \beta \end{aligned} \quad (3.32)$$

Với vị trí góc cánh tay theo trục quay của α , ta có:

$$\begin{aligned} \frac{\partial L}{\partial \alpha} &= 0 \\ \frac{\partial L}{\partial \dot{\alpha}} &= (J_0 + m_1 L_0^2) \dot{\alpha} + m_1 l_1^2 \dot{\alpha} \sin^2 \beta - m_1 L_0 l_1 \dot{\beta} \cos \beta \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\alpha}} \right) &= (J_0 + m_1 L_0^2) \ddot{\alpha} + m_1 l_1^2 \ddot{\alpha} \sin^2 \beta + 2m_1 l_1^2 \dot{\alpha} \dot{\beta} \sin \beta \cos \beta - m_1 L_0 l_1 \ddot{\beta} \cos \beta + m_1 L_0 l_1 \dot{\beta}^2 \sin \beta \\ &= (J_0 + m_1 L_0^2) \ddot{\alpha} + m_1 l_1^2 \ddot{\alpha} \sin^2 \beta - m_1 L_0 l_1 \ddot{\beta} \cos \beta + m_1 l_1^2 \dot{\alpha} \dot{\beta} \sin 2\beta + m_1 L_0 l_1 \dot{\beta}^2 \sin \beta \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\alpha}} \right) - \frac{\partial L}{\partial \alpha} &= \tau_\alpha \\ (J_0 + m_1 L_0^2) \ddot{\alpha} + m_1 l_1^2 \ddot{\alpha} \sin^2 \beta - m_1 L_0 l_1 \ddot{\beta} \cos \beta + m_1 l_1^2 \dot{\alpha} \dot{\beta} \sin 2\beta + m_1 L_0 l_1 \dot{\beta}^2 \sin \beta &= \tau_\alpha \end{aligned} \quad (3.33)$$

Tổng mô men xoắn trên trục theo hướng tăng α là τ_α . Nó bao gồm mô men xoắn gây ra bởi động cơ và mô men xoắn theo chiều ngược lại gây ra bởi ma sát.

$$\tau_\alpha = \tau_m - C_0 \dot{\alpha} \quad (3.34)$$

với: τ_m là mô men xoắn tác dụng gây ra bởi động cơ một chiều,

C_0 là ma sát nhớt tác dụng theo trục quay của α (hệ số ma sát của cánh tay).

Như vậy, ta có:

$$(J_0 + m_1 L_0^2) \ddot{\alpha} + m_1 l_1^2 \ddot{\alpha} \sin^2 \beta - m_1 L_0 l_1 \ddot{\beta} \cos \beta + m_1 l_1^2 \dot{\alpha} \dot{\beta} \sin 2\beta + m_1 L_0 l_1 \dot{\beta}^2 \sin \beta = \tau_m - C_0 \dot{\alpha} \quad (3.35)$$

$$(J_0 + m_1 L_0^2) \ddot{\alpha} + m_1 l_1^2 \ddot{\alpha} \sin^2 \beta - m_1 L_0 l_1 \ddot{\beta} \cos \beta + C_0 \dot{\alpha} + m_1 l_1^2 \dot{\alpha} \dot{\beta} \sin 2\beta + m_1 L_0 l_1 \dot{\beta}^2 \sin \beta = \tau_m \quad (3.36)$$

Thay (3.22) vào (3.36) ta được:

$$\begin{aligned} (J_0 + m_1 L_0^2) \ddot{\alpha} + m_1 l_1^2 \ddot{\alpha} \sin^2 \beta - m_1 L_0 l_1 \ddot{\beta} \cos \beta + (C_0 + \frac{K_t K_b}{R_a}) \dot{\alpha} \\ + m_1 l_1^2 \dot{\alpha} \dot{\beta} \sin 2\beta + m_1 L_0 l_1 \dot{\beta}^2 \sin \beta - \frac{K_t K_u}{R_a} u = 0 \end{aligned} \quad (3.37)$$

Tương tự, ta viết cho trục quay cho góc β :

$$\frac{\partial L}{\partial \beta} = m_1 l_1^2 \dot{\alpha}^2 \sin \beta \cos \beta + m_1 L_0 l_1 \dot{\alpha} \dot{\beta} \sin \beta + m_1 g l_1 \sin \beta$$

$$= \frac{1}{2} m_1 l_1^2 \dot{\alpha}^2 \sin 2\beta + m_1 L_0 l_1 \dot{\alpha} \dot{\beta} \sin \beta + m_1 g l_1 \sin \beta$$

$$\frac{\partial L}{\partial \dot{\beta}} = (J_1 + m_1 l_1^2) \dot{\beta} - m_1 L_0 l_1 \dot{\alpha} \cos \beta$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\beta}} \right) = (J_1 + m_1 l_1^2) \ddot{\beta} - m_1 L_0 l_1 \ddot{\alpha} \cos \beta + m_1 L_0 l_1 \dot{\alpha} \dot{\beta} \sin \beta$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\beta}} \right) - \frac{\partial L}{\partial \beta} = \tau_\beta$$

$$\begin{aligned} (J_1 + m_1 l_1^2) \ddot{\beta} - m_1 L_0 l_1 \ddot{\alpha} \cos \beta + m_1 L_0 l_1 \dot{\alpha} \dot{\beta} \sin \beta - \frac{1}{2} m_1 l_1^2 \dot{\alpha}^2 \sin 2\beta - m_1 L_0 l_1 \dot{\alpha} \dot{\beta} \sin \beta \\ - m_1 g l_1 \sin \beta = \tau_\beta \end{aligned}$$

$$-m_1 L_0 l_1 \ddot{\alpha} \cos \beta + (J_1 + m_1 l_1^2) \ddot{\beta} - \frac{1}{2} m_1 l_1^2 \dot{\alpha}^2 \sin 2\beta - m_1 g l_1 \sin \beta = \tau_\beta \quad (3.38)$$

Tổng mô men xoắn trên trục theo hướng tăng β là τ_β . Theo trục này thì mô men xoắn chỉ bao gồm thành phần gây ra bởi ma sát.

$$\tau_\beta = -C_1 \dot{\beta} \quad (3.39)$$

với C_1 là ma sát nhớt tác dụng theo trục quay của β (hệ số ma sát của con lắc).

Thay (3.39) vào (3.38) ta có:

$$\begin{aligned} -m_1 L_0 l_1 \ddot{\alpha} \cos \beta + (J_1 + m_1 l_1^2) \ddot{\beta} - \frac{1}{2} m_1 l_1^2 \dot{\alpha}^2 \sin 2\beta - m_1 g l_1 \sin \beta &= -C_1 \dot{\beta} \\ -m_1 L_0 l_1 \ddot{\alpha} \cos \beta + (J_1 + m_1 l_1^2) \ddot{\beta} - \frac{1}{2} m_1 l_1^2 \dot{\alpha}^2 \sin 2\beta + C_1 \dot{\beta} - m_1 g l_1 \sin \beta &= 0 \end{aligned} \quad (3.40)$$

Chuyển các phương trình mô tả toán học hệ con lắc ngược quay (3.22),(3.37), (3.40) về dạng ma trận, ta sẽ có mô hình động học hệ con lắc ngược quay khi con lắc ở vị trí thẳng đứng hướng lên trên có dạng như sau:

$$\begin{pmatrix} J_0 + m_1 L_0^2 + m_1 l_1^2 \sin^2 \beta & m_1 L_0 l_1 \cos \beta \\ -(m_1 L_0 l_1 \cos \beta) & J_1 + m_1 l_1^2 \end{pmatrix} \begin{pmatrix} \ddot{\alpha} \\ \ddot{\beta} \end{pmatrix} + \begin{pmatrix} C_0 + \frac{K_t K_b}{R_a} + \frac{1}{2} m_1 l_1^2 \dot{\beta} \sin 2\beta & m_1 L_0 l_1 \dot{\beta} \sin \beta + \frac{1}{2} m_1 l_1^2 \dot{\alpha} \sin 2\beta \\ -\left(\frac{1}{2} m_1 l_1^2 \dot{\alpha} \sin 2\beta\right) & C_1 \end{pmatrix} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \end{pmatrix} + \begin{pmatrix} 0 \\ -m_1 g l_1 \sin \beta \end{pmatrix} = \begin{pmatrix} \frac{K_t K_u}{R_a} \\ 0 \end{pmatrix} u \quad (3.41)$$

Các ký hiệu :

- $\alpha(rad)$: Góc giữa cánh tay và trục x
- $\beta(rad)$: Góc quay của con lắc hướng lên,
- $J_0(kg.m^2)$: Mô men kích thích của cánh tay,
- $J_1(kg.m^2)$: Mô men kích thích của con lắc,
- $C_0(kg.m^2 / s)$: Hệ số ma sát của cánh tay,
- $C_1(kg.m^2 / s)$: Hệ số ma sát của con lắc,
- $m_1(kg)$: Khối lượng hiệu dụng của con lắc,
- $l_1(m)$: Chiều dài hiệu dụng của con lắc,
- $L_0(m)$: Khoảng cách từ trục động cơ đến con lắc,
- $K_t(N.m / A)$: Hằng số mô men của động cơ,
- $K_b(V.s / rad)$: Hằng số sức điện động của động cơ,
- $K_u(V / count)$: Hệ số khuếch đại điều khiển động cơ,
- $R_a(\Omega)$: Điện trở phần ứng của động cơ,

- $g(m/s^2)$: Gia tốc trọng trường,
- $u(counts)$: Tín hiệu điều khiển động cơ (đưa tới bộ khuếch đại xung PWM).

Biến đổi lại mô hình (3.41) về dạng hệ phương trình vi phân:

$$\begin{cases} (J_0 + m_1 L_0^2 + m_1 l_1^2 \sin^2 \beta) \ddot{\alpha} - (m_1 L_0 l_1 \cos \beta) \ddot{\beta} + (C_0 + \frac{K_t K_b}{R_a} + \frac{1}{2} m_1 l_1^2 \dot{\beta} \sin 2\beta) \dot{\alpha} \\ + (m_1 L_0 l_1 \dot{\beta} \sin \beta + \frac{1}{2} m_1 l_1^2 \dot{\alpha} \sin 2\beta) \dot{\beta} - \frac{K_t K_u}{R_a} u = 0 \\ - (m_1 L_0 l_1 \cos \beta) \ddot{\alpha} + (J_1 + m_1 l_1^2) \ddot{\beta} - (\frac{1}{2} m_1 l_1^2 \dot{\alpha} \sin 2\beta) \dot{\alpha} + C_1 \dot{\beta} - m_1 g l_1 \sin \beta = 0 \end{cases} \quad (3.42)$$

Đặt biến trạng thái:

$$\underline{x} = [x_1, x_2, x_3, x_4]^T = [\alpha, \dot{\alpha}, \beta, \dot{\beta}]^T$$

thì:

$$\dot{\underline{x}} = [\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4]^T = [\dot{\alpha}, \ddot{\alpha}, \dot{\beta}, \ddot{\beta}]^T$$

Thay vào (3.42) ta có:

$$\begin{cases} \dot{x}_1 = x_2 \\ (J_0 + m_1 L_0^2 + m_1 l_1^2 \sin^2 x_3) \dot{x}_2 - (m_1 L_0 l_1 \cos x_3) \dot{x}_4 + (C_0 + \frac{K_t K_b}{R_a} + \frac{1}{2} m_1 l_1^2 x_4 \sin 2x_3) x_2 + \\ + (m_1 L_0 l_1 x_4 \sin x_3 + \frac{1}{2} m_1 l_1^2 x_2 \sin 2x_3) x_4 - \frac{K_t K_u}{R_a} u = 0 \\ \dot{x}_3 = x_4 \\ - (m_1 L_0 l_1 \cos x_3) \dot{x}_2 + (J_1 + m_1 l_1^2) \dot{x}_4 - (\frac{1}{2} m_1 l_1^2 x_2 \sin 2x_3) x_2 + C_1 x_4 - m_1 g l_1 \sin x_3 = 0 \end{cases} \quad (3.43)$$

Sử dụng các ký hiệu:

$$k_1 = (J_0 + m_1 L_0^2 + m_1 l_1^2 \sin^2 x_3), \quad k_2 = (m_1 L_0 l_1 \cos x_3),$$

$$k_3 = (C_0 + \frac{K_t K_b}{R_a} + \frac{1}{2} m_1 l_1^2 x_4 \sin 2x_3), \quad k_4 = (m_1 L_0 l_1 x_4 \sin x_3 + \frac{1}{2} m_1 l_1^2 x_2 \sin 2x_3),$$

$$k_5 = \frac{K_t K_u}{R_a}, \quad k_6 = (J_1 + m_1 l_1^2), \quad k_7 = (\frac{1}{2} m_1 l_1^2 x_2 \sin 2x_3),$$

$$k_8 = C_1, \quad k_9 = m_1 g l_1 \sin x_3$$

thì mô hình (3.43) sẽ được viết ngắn gọn lại thành:

$$\begin{cases} \dot{x}_1 = x_2 \\ k_1 \dot{x}_2 - k_2 \dot{x}_4 + k_3 x_2 + k_4 x_4 - k_5 u = 0 \\ \dot{x}_3 = x_4 \\ -k_2 \dot{x}_2 + k_6 \dot{x}_4 - k_7 x_2 + k_8 x_4 - k_9 = 0 \end{cases} \quad (3.44)$$

Tiếp theo, từ (3.44) ta có:

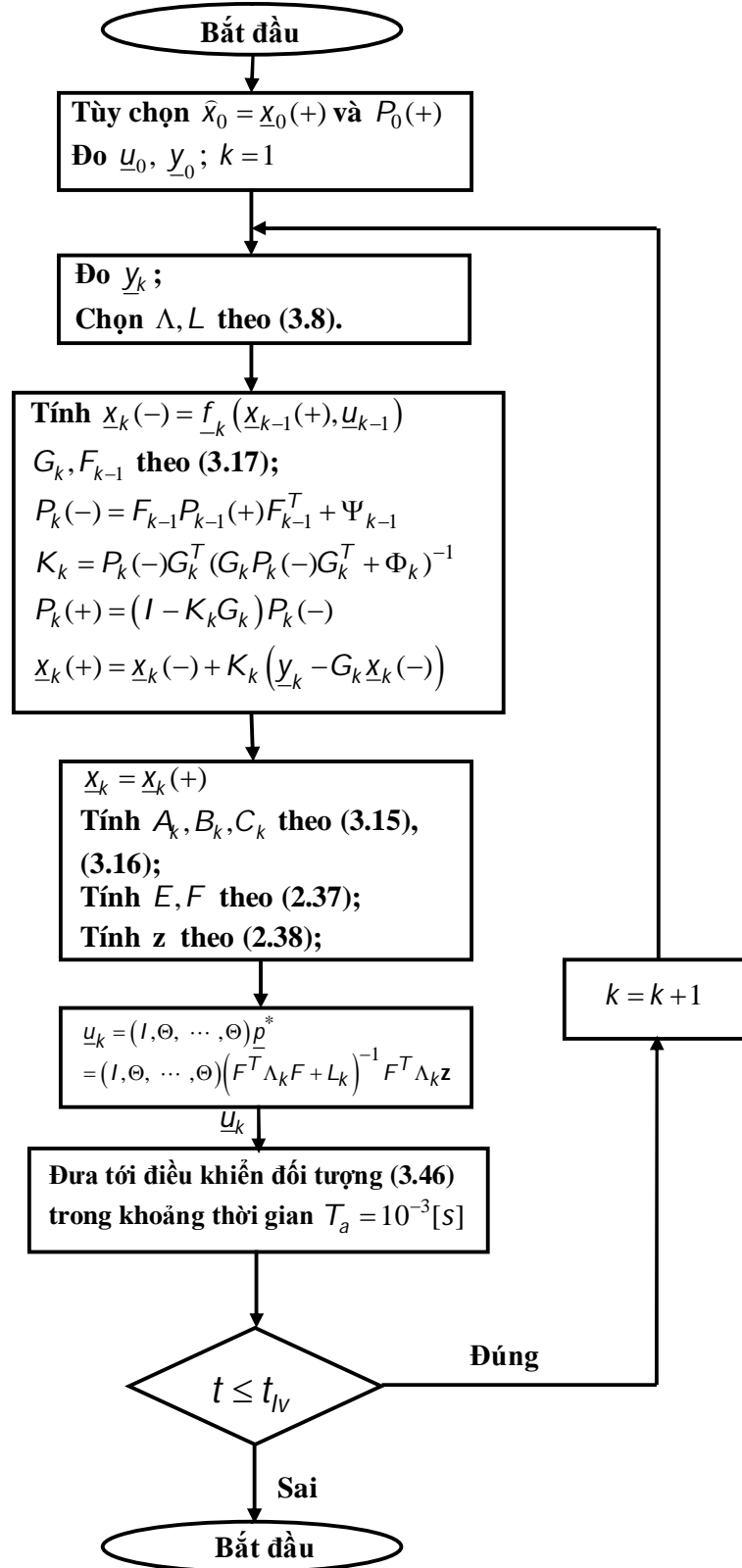
$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{k_2 k_7 - k_6 k_3}{(k_1 k_6 - k_2^2)} x_2 - \frac{k_2 k_8 + k_6 k_4}{(k_1 k_6 - k_2^2)} x_4 + \frac{k_2 k_9}{(k_1 k_6 - k_2^2)} + \frac{k_6 k_5}{(k_1 k_6 - k_2^2)} u \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{k_2 (k_2 k_7 - k_6 k_3) + k_7 (k_1 k_6 - k_2^2)}{k_6 (k_1 k_6 - k_2^2)} x_2 - \frac{k_2 (k_2 k_8 + k_6 k_4) + k_8 (k_1 k_6 - k_2^2)}{k_6 (k_1 k_6 - k_2^2)} x_4 \\ + \frac{k_2^2 k_9 + k_9 (k_1 k_6 - k_2^2)}{k_6 (k_1 k_6 - k_2^2)} + \frac{k_2 k_6 k_5}{k_6 (k_1 k_6 - k_2^2)} u \end{cases} \quad (3.45)$$

Ký hiệu các tham số:

$$\begin{aligned} K_1 &= \frac{k_2 k_7 - k_6 k_3}{(k_1 k_6 - k_2^2)}, \quad K_2 = \frac{k_2 k_8 + k_6 k_4}{(k_1 k_6 - k_2^2)}, \quad K_3 = \frac{k_2 k_9}{(k_1 k_6 - k_2^2)}, \quad K_4 = \frac{k_6 k_5}{(k_1 k_6 - k_2^2)} \\ K_5 &= \frac{k_2 (k_2 k_7 - k_6 k_3) + k_7 (k_1 k_6 - k_2^2)}{k_6 (k_1 k_6 - k_2^2)}, \quad K_6 = \frac{k_2 (k_2 k_8 + k_6 k_4) + k_8 (k_1 k_6 - k_2^2)}{k_6 (k_1 k_6 - k_2^2)} \\ K_7 &= \frac{k_2^2 k_9 + k_9 (k_1 k_6 - k_2^2)}{k_6 (k_1 k_6 - k_2^2)}, \quad K_8 = \frac{k_2 k_6 k_5}{k_6 (k_1 k_6 - k_2^2)} \end{aligned}$$

rồi thay vào (3.45), ta được:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = K_1 x_2 - K_2 x_4 + K_3 + K_4 u \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = K_5 x_2 - K_6 x_4 + K_7 + K_8 u \end{cases} \quad (3.46)$$



Hình 3.18: Lưu đồ thuật toán thiết kế bộ ĐKDBPHDR bám theo giá trị đầu ra đặt cho con lắc ngược quay theo Thuật toán 2.7

Bây giờ, để áp dụng bộ điều khiển phản hồi trạng thái cho bởi thuật toán 2.4, ta sẽ chuyển đổi mô hình (3.46) về dạng mô hình song tuyến theo trạng thái:

$$\begin{cases} \frac{d\underline{x}}{dt} = A(\underline{x})\underline{x} + B(\underline{x})u \\ \underline{y} = \underline{x}_3 = (0, 0, 1, 0)\underline{x} \end{cases} \quad (3.47)$$

$$\text{với: } A(\underline{x}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & K_1 + K_3/x_2 & 0 & -K_2 \\ 0 & 0 & 0 & 1 \\ 0 & K_5 + K_7/x_2 & 0 & -K_6 \end{pmatrix}, \quad B(\underline{x}) = \begin{pmatrix} 0 \\ K_4 \\ 0 \\ K_8 \end{pmatrix}$$

Đầu ra ở đây được lựa chọn là vị trí góc lắc β . Mô hình (3.47) của đối tượng con lắc ngược quay cũng vẫn có dạng giống như mô hình (3.9) của đối tượng con lắc ngược. Với mô hình toán đã được biến đổi về dạng song tuyến theo trạng thái như (3.47), tiếp theo, luận án sẽ tiến hành thiết kế bộ điều khiển phản hồi đầu ra cho đối tượng con lắc ngược quay tại phòng thí nghiệm.

3.2.2 Kết quả mô phỏng bộ điều khiển phản hồi đầu ra cho đối tượng con lắc ngược quay

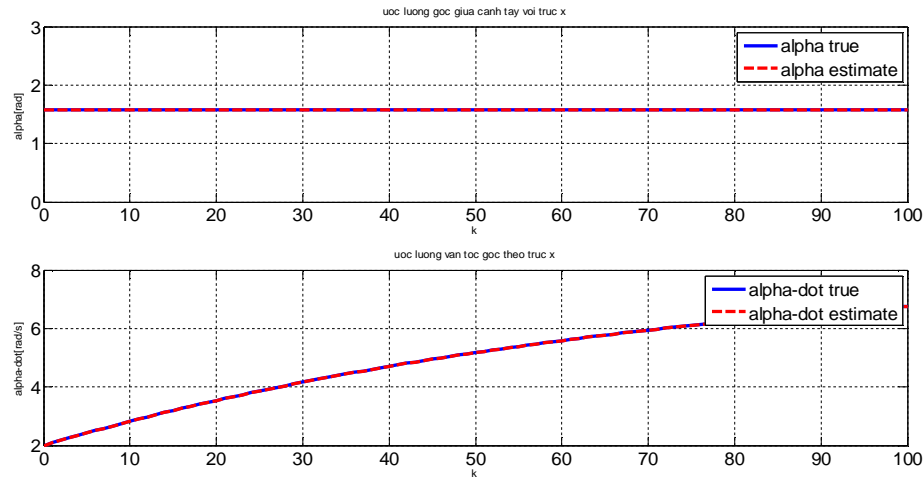
Sử dụng bộ các tham số vật lý sau cho hệ thống (Được xác định bằng thực nghiệm theo [64,3]) để thực hiện mô phỏng theo Thuật toán 2.7:

$$\begin{aligned} m_1 &= 0,0319(\text{kg}), \quad l_1 = 0,1572(\text{m}), \quad K_u = 0,0636(\text{V} / \text{count}), \quad K_t = 0,0706(\text{N.m} / \text{A}), \\ K_b &= 0,0707(\text{V.s} / \text{rad}), \quad R_a = 0,9(\Omega), \quad L_0 = 0,0312(\text{m}), \quad J_0 = 0,008591(\text{kg.m}^2), \\ C_0 &= 0,006408(\text{kg.m}^2 / \text{s}), \quad J_1 = 0,000217(\text{kg.m}^2), \quad C_1 = 0,000158(\text{kg.m}^2 / \text{s}) \end{aligned}$$

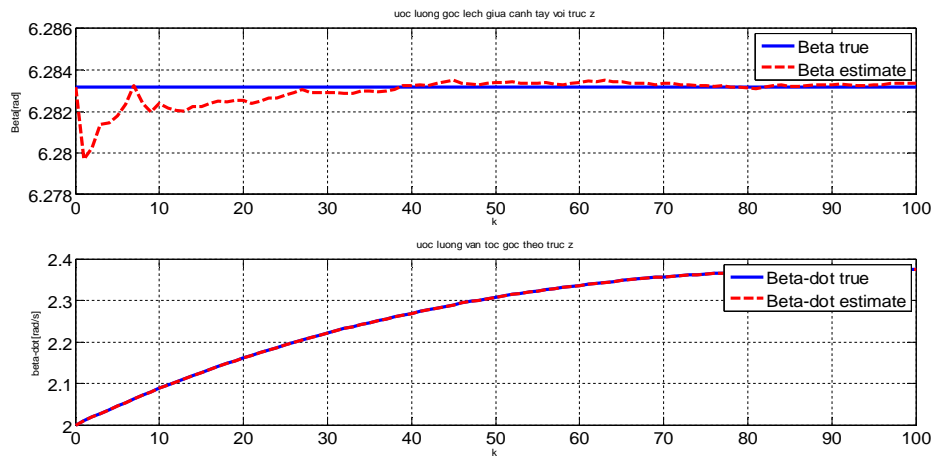
ta có được kết quả quan sát trạng thái theo EKF3 (Thuật toán 1.9) cho hệ con lắc ngược quay cho ở Hình 3.19 và Hình 3.20. Kết quả này cho thấy bộ quan sát kháng được tốt nhiễu ồn trắng có lẫn trong cả trạng thái và đầu ra của hệ, đúng như lý thuyết đã khẳng định.

Hình 3.21 biểu diễn kết quả mô phỏng bộ điều khiển phản hồi đầu ra theo Thuật toán 2.7 cho hệ con lắc ngược quay. Nó cho thấy việc sử dụng thêm bộ quan sát trạng thái EKF3 gần như không làm thay đổi chất lượng bám giá trị đặt của bộ

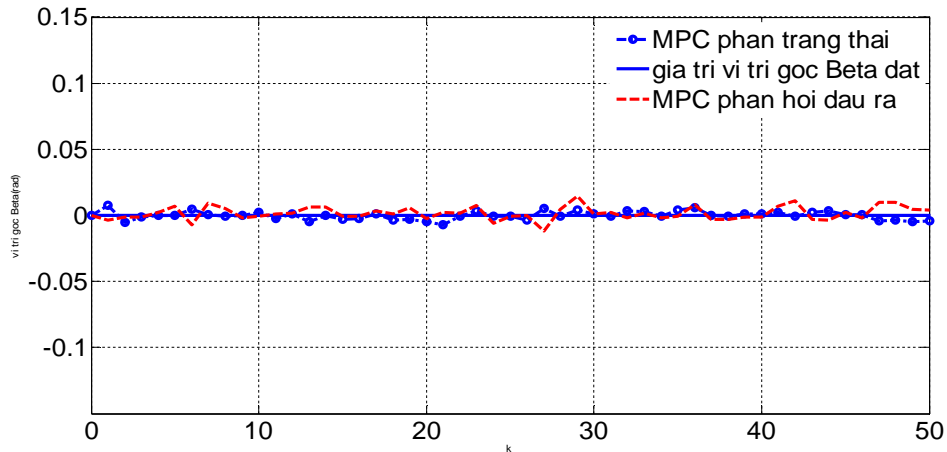
điều khiển phản hồi trạng thái, tức là bộ điều khiển dự báo phản hồi đầu ra cũng đã kháng được tốt nhiễu ồn trắng có trong hệ.



Hình 3.19: Kết quả quan sát trạng thái α và $\dot{\alpha}$ bằng EKF loại 3



Hình 3.20: Kết quả quan sát trạng thái β và $\dot{\beta}$ bằng EKF loại 3



Hình 3.21: Kết quả mô phỏng bộ điều khiển phản hồi đầu ra so sánh với bộ điều khiển MPC phản hồi trạng thái cho đối tượng con lắc ngược quay (đầu ra là vị trí góc con lắc theo trục z)

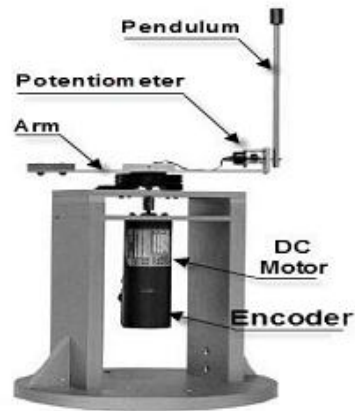
Toàn bộ phần chương trình mô phỏng cho đối tượng con lắc ngược quay được trình bày trong phần phụ lục **P4**.

3.2.3 Mô tả hệ thống và kết quả thí nghiệm

3.2.3.1 Mô tả hệ thống thí nghiệm

Hình 3.23 mô tả hệ thống thí nghiệm con lắc ngược quay Kri PP-300 tại phòng thí nghiệm Đo lường – Điều khiển, trường ĐH KTCN Thái Nguyên. Hệ thống này bao gồm:

- *Phần mềm hệ thống:* Cung cấp giao diện cho người dùng dễ dàng nghiên cứu và sử dụng hệ thống.
- *Hệ thống máy tính:* Công cụ cung cấp giao diện giao tiếp giữa người dùng và bo mạch điều khiển thông qua cổng COM.
- *Bo mạch điều khiển:* Sử dụng vi mạch Arduino Mega 2560, có thể sử dụng môi trường Matlab-Simulink để cài đặt trực tiếp các thuật toán điều khiển, phối hợp kết nối dễ dàng với bo mạch công suất điều khiển động cơ của hệ thống.



Hình 3.22: Mô hình con lắc ngược quay Kri-300



*Hình 3.23: Mô hình thí nghiệm con lắc ngược quay Kri PP-300 tại
Phòng thí nghiệm Đo lường – Điều khiển, Trường ĐH KTCN
Thái Nguyên*



Hình 3.24: Vi mạch Arduino Mega 2560

- *Bộ mạch công suất điều khiển động cơ (Motor Driver Board)* : Được thiết kế để kết nối điều khiển động cơ thông qua tín hiệu PWM.



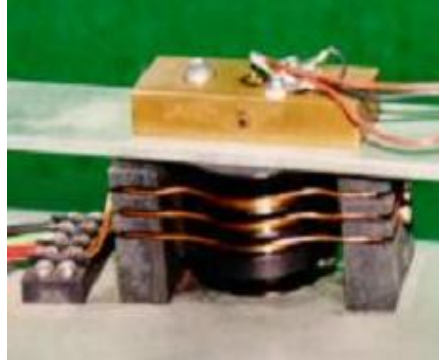
Hình 3.25: Bộ mạch công suất điều khiển động cơ

- *Bộ nguồn (Power Supply)*: Sử dụng model RPS 305DU, biến đổi điện áp từ xoay chiều 220V-50Hz sang nguồn 1 chiều đối xứng.



Hình 3.26: Bộ nguồn RPS 305DU

- Bộ con lắc ngược quay:



Hình 3.27: Bộ nguồn RPS 305DU

Cảm biến góc quay: là 1 biến trở có giá trị $5k\Omega(\pm 20\%)$, được cấp nguồn +5V(DC), tín hiệu đầu ra có biên từ 0 đến +5V(DC) ứng với góc quay từ 0 đến 360° , dải mã hóa được 10 bit ADC. Trên cánh tay của hệ thống con lắc còn có cơ cấu giúp truyền tín hiệu từ cảm biến góc quay về mạch vi xử lý thông qua 3 vòng dây được quấn quanh một trống quay. Cơ cấu này giúp cho quá trình đưa tín hiệu vị trí con lắc về vi xử lý một cách dễ dàng, linh hoạt.

Encoder: Sử dụng loại E6B2-CWS6B – Rotary Encoder; điện áp +5 đến 24VDC; xung/vòng quay: 1000; đầu ra: A,B,Z; tốc độ quay max: 6000 vòng/phút.



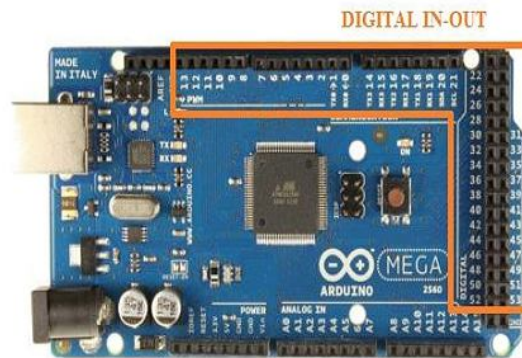
Hình 3.28: Vị trí Encoder

Hệ thống thí nghiệm sử dụng bo mạch điện tử Arduino Mega 2560 để kết nối với máy tính. Arduino Mega 2560 là một vi mạch điện tử điều khiển dựa trên nền tảng của chip ATmega 2560, nó tích hợp tất cả các tính năng cần thiết của một vi điều khiển. Ta có thể kết nối Arduino Mega 2560 với máy tính thông qua cổng USB, cài đặt phần mềm của Arduino và sau đó thực hiện cài đặt một cách dễ dàng các tập lệnh của bài toán điều khiển trên giao diện máy tính.

Nguồn cho vi mạch Arduino Mega 2560 được cấp qua cáp kết nối USB. Arduino Mega 2560 có 256 KB bộ nhớ flash để lưu trữ mã lệnh điều khiển, 8KB SRAM và 4 KB EEPROM để nâng cao khả năng đọc và ghi các dữ liệu điều khiển.

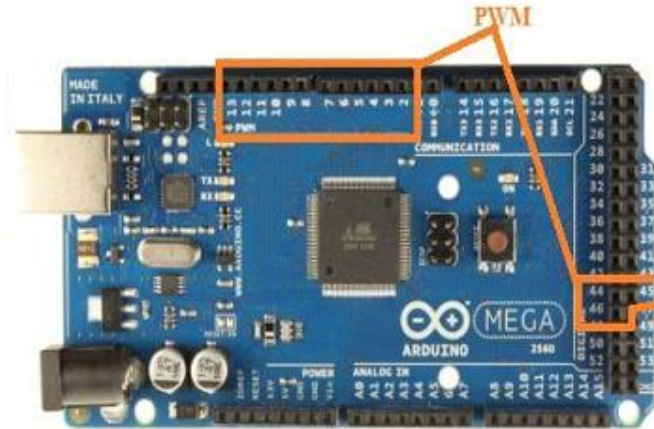
Arduino Mega 2560 có các cổng và các đầu vào, ra :

- Đầu vào, ra số (Digital I/O): sử dụng 54 chân sử dụng làm đầu vào, ra số.



Hình 3.29: Đầu vào, ra số trên Arduino Mega 2560

- Đầu ra PWM (Pulse Width Modulation): gồm các chân số 2 đến 13, chân số 44 đến 46, các chân này xuất ra tín hiệu PWM được mã hóa 8 bit.



Hình 3.30: Đầu ra PWM trên Arduino Mega 2560

- Cổng Communication: nhận và truyền dữ liệu vào – ra theo kiểu nối tiếp (COM):



Hình 3.31: Cổng Communication trên Arduino Mega 2560

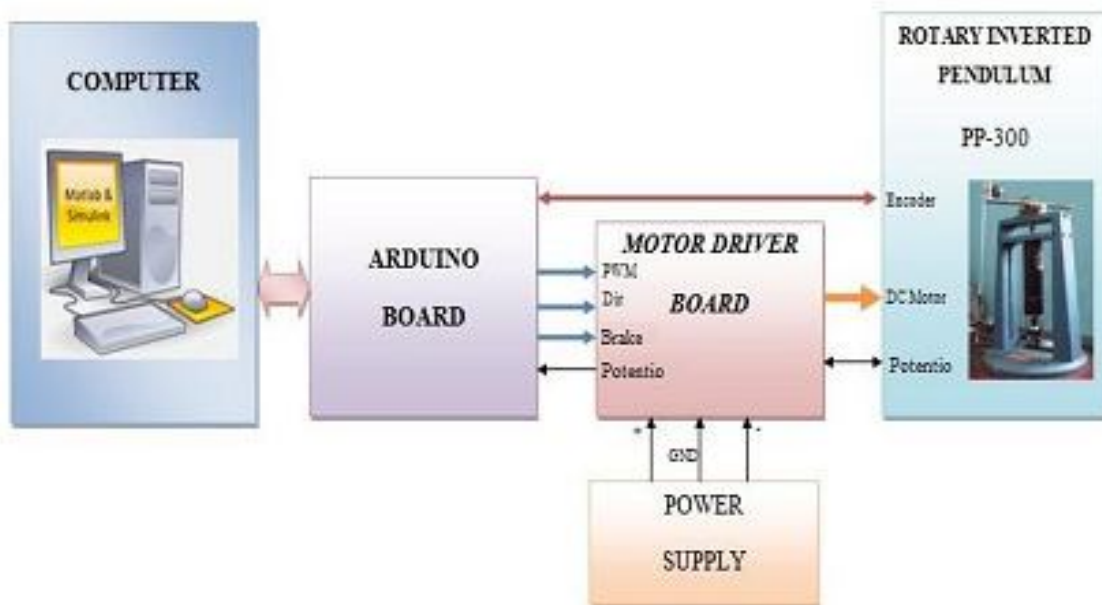
- Đầu vào tương tự (Analog in): tích hợp 16 đầu vào tương tự.



Hình 3.32: Đầu vào tương tự trên Arduino Mega 2560

3.2.3.2 Kết quả thí nghiệm

Sơ đồ kết nối thiết bị thí nghiệm được biểu diễn ở Hình 3.33. Hình 3.34. là hình ảnh thực tế bàn thí nghiệm con lắc ngược quay tại Trường ĐH Kỹ thuật Công nghiệp - ĐHTN.



Hình 3.33: Sơ đồ kết nối thiết bị thí nghiệm

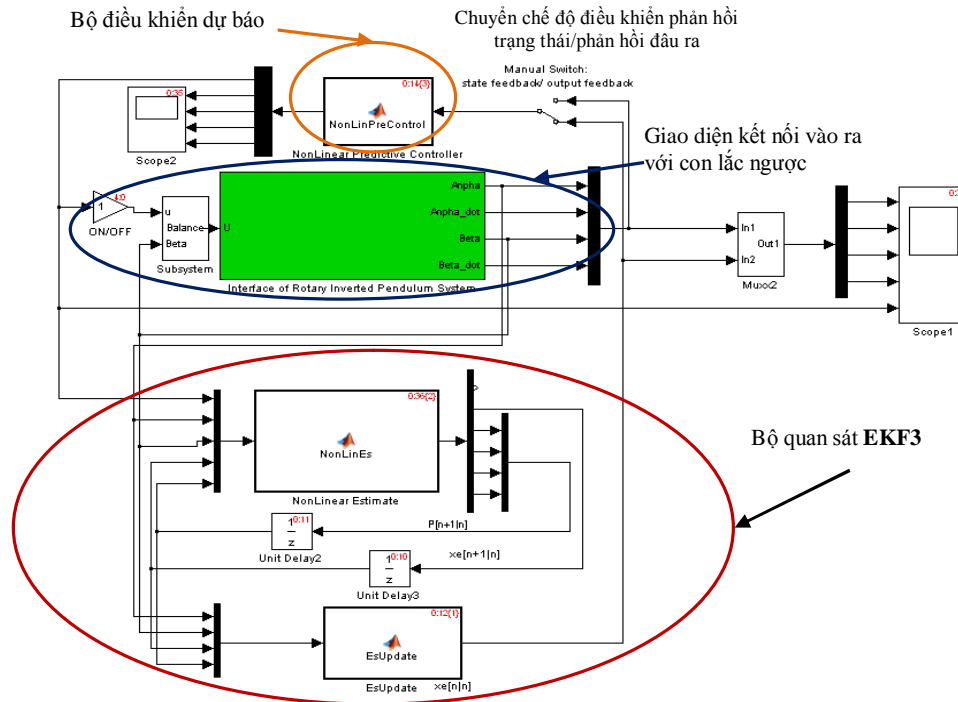
Chuyển đổi chương trình phần lập trình bằng M-file đã thực hiện trong mục 3.2.2 sang Simulink của Matlab, thực hiện kết nối giữa hệ thống thực với môi trường Matlab thông qua vi mạch Arduino và kết nối với máy tính.

Để có thể cài đặt cho Arduino Mega 2560, trước tiên ta kết nối Arduino Mega 2560 với máy tính, tải phần mềm Arduino Software và cài đặt trên máy tính, cài đặt thư viện Arduino trong Matlab/Simulink (thư viện giao tiếp giữa máy tính và Arduino trong toolbox của Matlab, thư viện kết nối vào – ra cho Arduino Mega 2560).

Giao diện thí nghiệm cho hệ con lắc ngược quay được thể hiện trên Hình 3.35. Khối NonlinPreControl chứa các cài đặt của bộ điều khiển dự báo theo Thuật toán 2.4 (mã nguồn chương trình được trình bày trong phụ lục **P5**). Khối Rotary Inverted Pendulum là thiết kế cài đặt giao diện vào – ra với con lắc ngược, với đầu vào là tín hiệu điều khiển đưa tới từ bộ điều khiển dự báo, các đầu ra được lấy lần lượt là các giá trị góc và vận tốc góc của con lắc và của cánh tay. Bộ quan sát trạng thái Kalman mở rộng để quan sát trạng thái cho đối tượng được thiết kế trong các khối Nonlinear Estimate (để tính toán các trạng thái ước đoán) và EsUpdate (để cập nhật các trạng thái theo từng vòng lặp). (Mã nguồn chương trình của hai khối này được trình bày trong phụ lục **P6**)



Hình 3.34: Hình ảnh bàn thí nghiệm con lắc ngược quay tại trường ĐH KTCN – Đại học Thái Nguyên

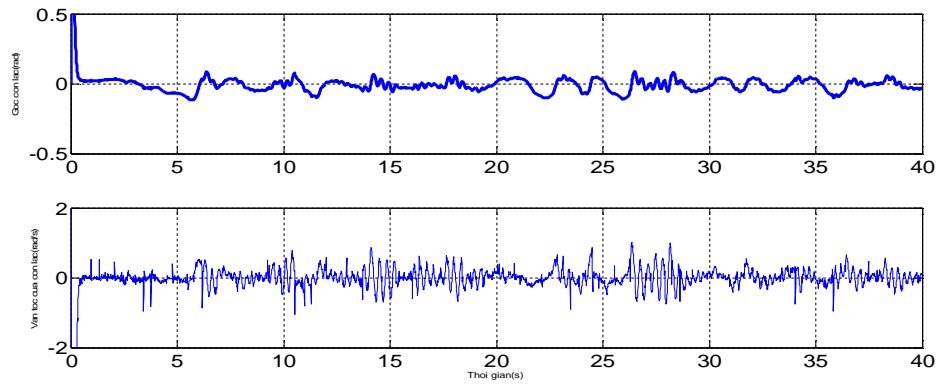


Hình 3.35: Giao diện thí nghiệm cho hệ con lắc ngược quay, thực hiện trên Matlab/Simulink Real Time

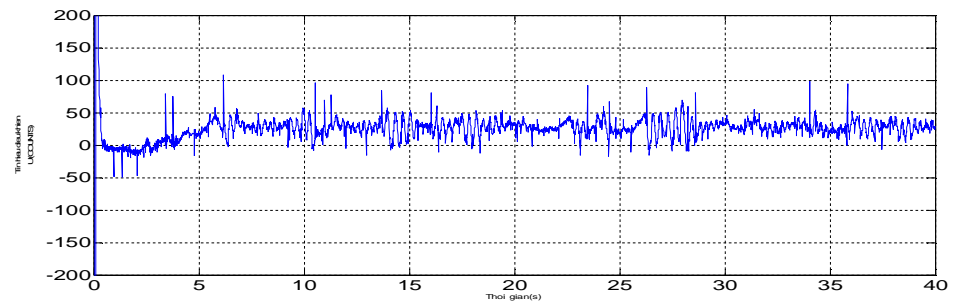
Kết quả khi sử dụng bộ điều khiển phản hồi trạng thái

Để tiến hành thí nghiệm với bộ điều khiển dự báo phản hồi trạng thái, trong giao diện thí nghiệm (Hình 3.35), khóa chuyển đổi đặt ở vị trí phía trên, các tín hiệu trạng thái đưa tới bộ điều khiển phản hồi trạng thái được lấy từ giao diện kết nối vào – ra với con lắc ngược. Góc và vận tốc góc của con lắc được đưa tới khối Scope 1. Tín hiệu điều khiển đầu ra của bộ điều khiển phản hồi trạng thái được đưa tới khối Scope 2.

Hình 3.36 thể hiện giá trị góc và vận tốc góc của con lắc khi điều khiển bằng bộ điều khiển dự báo phản hồi trạng thái theo Thuật toán 2.4. Góc con lắc được điều khiển bám theo giá trị ổn định 0 (vị trí con lắc thẳng đứng phía trên), với sai lệch $< 0,1(rad)$. Tín hiệu điều khiển đưa ra từ bộ điều khiển dự báo phản hồi trạng thái được thể hiện trên Hình 3.37.



Hình 3.36: Góc và vận tốc góc của con lắc khi sử dụng bộ điều khiển DBPHTT theo Thuật toán 2.4

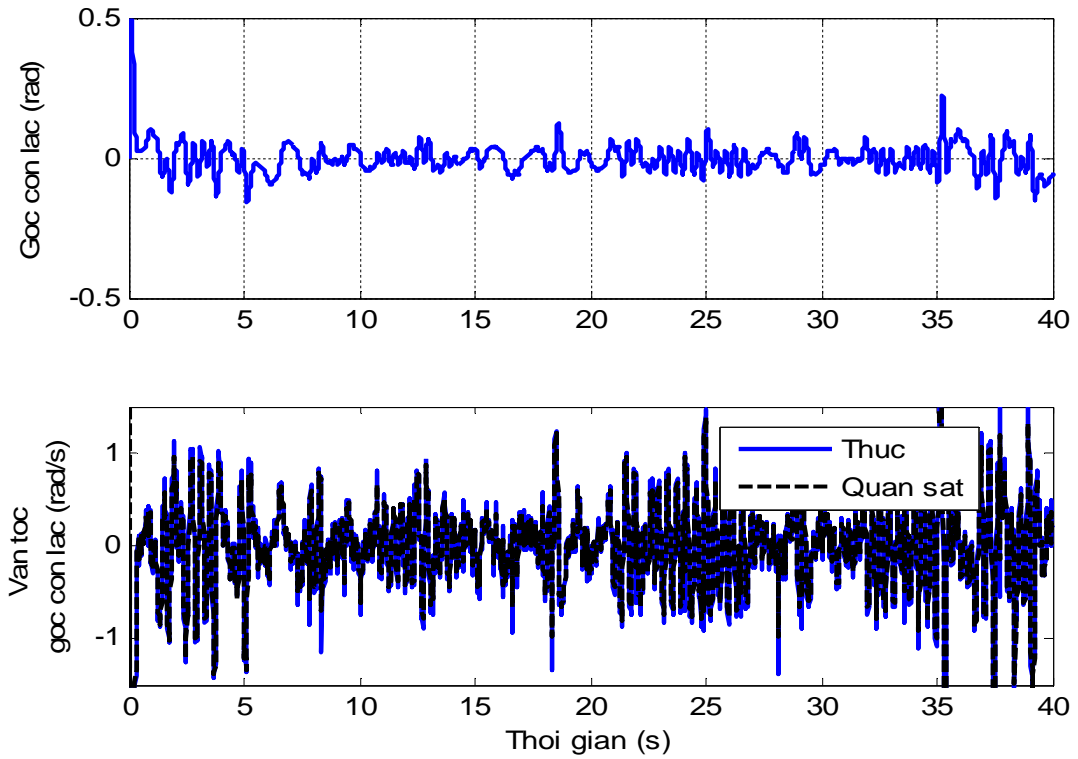


Hình 3.37: Tín hiệu điều khiển từ đầu ra của bộ điều khiển dự báo theo Thuật toán 2.4 (đưa tới đầu vào của bộ khuếch đại xung PWM)

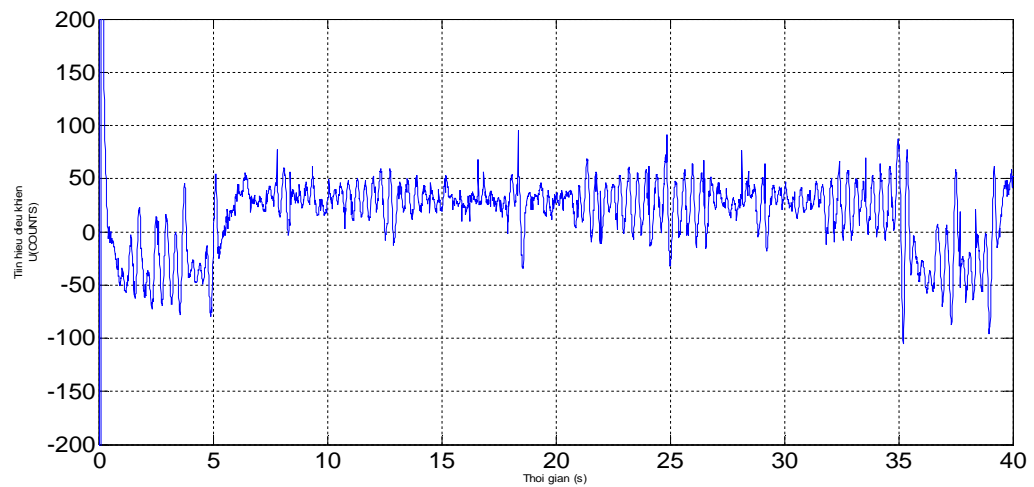
Kết quả khi sử dụng bộ điều khiển phản hồi đầu ra

Để tiến hành thí nghiệm với bộ điều khiển dự báo phản hồi đầu ra, lúc này khóa chuyển đổi được chuyển xuống vị trí phía dưới, các tín hiệu trạng thái đưa tới bộ điều khiển phản hồi trạng thái lúc này được lấy từ đầu ra của bộ quan sát EKF 3. Góc và vận tốc góc của con lắc được đưa tới khối Scope 1. Tín hiệu điều khiển đầu ra của bộ điều khiển phản hồi đầu ra được đưa tới khối Scope 2.

Các kết quả thí nghiệm khi sử dụng bộ điều khiển phản hồi đầu ra theo nguyên lý tách để ổn định hệ con lắc ngược quay ở vị trí phía trên được thể hiện trong các Hình 3.38 - Hình 3.40.



Hình 3.38: Góc và vận tốc góc của con lắc khi điều khiển bằng bộ điều khiển dự báo phản hồi đầu ra theo Thuật toán 2.7



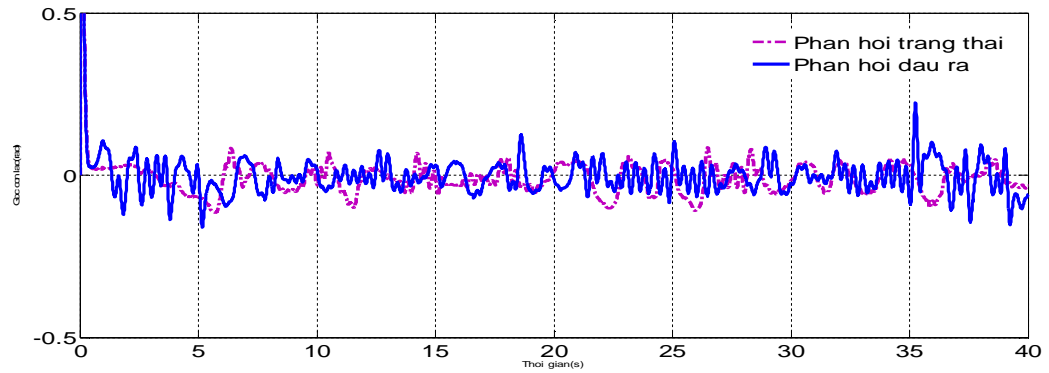
Hình 3.39: Tín hiệu điều khiển đưa ra bởi bộ điều khiển dự báo phản hồi đầu ra theo Thuật toán 2.7 (đưa tới đầu vào bộ khuếch đại xung PWM)



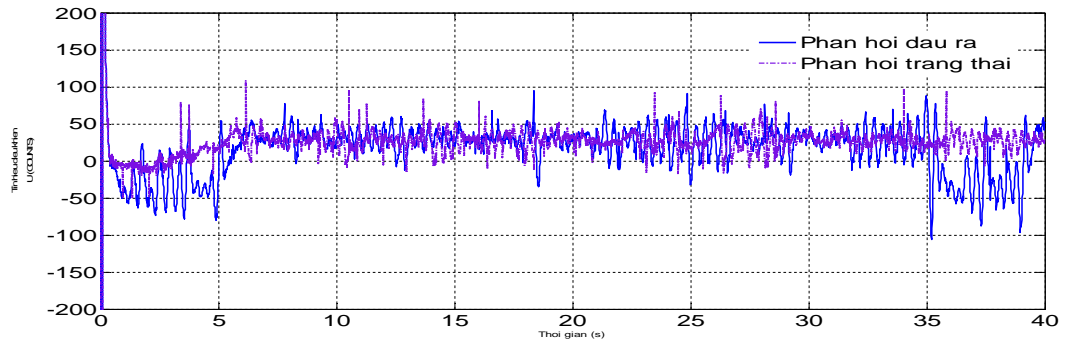
Hình 3.40: Hình ảnh tại bàn thí nghiệm khi điều khiển con lắc ở vị trí thẳng đứng hướng lên trên (góc con lắc bám theo giá trị 0) bằng bộ điều khiển dự báo phản hồi đầu ra theo Thuật toán 2.7

Hình 3.38 thể hiện giá trị góc và vận tốc con lắc khi sử dụng bộ điều khiển dự báo phản hồi đầu ra theo Thuật toán 2.7. Góc con lắc vẫn được điều khiển bám ổn định theo giá trị 0 hay con lắc ổn định ở vị trí thẳng đứng. Tín hiệu điều khiển từ bộ điều khiển phản hồi đầu ra được thể hiện trên Hình 3.39. Hình 3.40 là hình ảnh tại bàn thí nghiệm khi điều khiển con lắc ở vị trí thẳng đứng hướng lên trên bằng bộ điều khiển dự báo phản hồi đầu ra. Video của thí nghiệm được tải lên Youtube tại địa chỉ: <https://youtu.be/qxuwWDUs8UU>.

So sánh chất lượng của hai bộ điều khiển



Hình 3.41: So sánh góc con lắc được điều khiển trong hai trường hợp phản hồi đầu ra và phản hồi trạng thái



Hình 3.42: So sánh tín hiệu điều khiển trong hai trường hợp phản hồi đầu ra và phản hồi trạng thái

Hình 3.41 cho ta kết quả so sánh về chất lượng làm việc giữa hai bộ điều khiển phản hồi trạng thái theo Thuật toán 2.4 và bộ điều khiển dự báo phản hồi đầu ra theo Thuật toán 2.7. Trên Hình 3.41, góc con lắc được điều khiển bám ổn định giá trị 0 và có sai lệch nhỏ khi sử dụng 2 bộ điều khiển. Tín hiệu điều khiển được so sánh trên Hình 3.42. Có thể thấy, bộ điều khiển phản hồi đầu ra theo Thuật toán 2.7 đã làm việc tốt và điều khiển góc con lắc bám theo giá trị đặt bằng 0 giống như khi ta sử dụng bộ điều khiển phản hồi trạng thái. Điều này là rất quan trọng khi mà ở bộ điều khiển dự báo phản hồi đầu ra, ta đã sử dụng các biến trạng thái là các giá trị ước lượng được từ bộ quan sát trạng thái Kalman mở rộng có khả năng lọc tốt nhiễu Gauss còn ở bộ điều khiển dự báo phản hồi trạng thái, ta phải sử dụng cảm biến để đo các giá trị này.

3.3 Kết luận chương 3

Trong chương 3 luận án đã trình bày các nội dung sau:

- 1) Áp dụng phương pháp điều khiển dự báo phản hồi trạng thái hệ phi tuyến trên cơ sở sử dụng mô hình dự báo tuyến tính từng đoạn do luận án đề xuất (Thuật toán 2.3 - Thuật toán 2.6) vào điều khiển bám tín hiệu ra mẫu cho các đối tượng: con lắc ngược và con lắc ngược quay cho chất lượng điều khiển như mong muốn.
- 2) Bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách trên cơ sở ghép chung bộ quan sát trạng thái Kalman và bộ điều khiển dự báo phản hồi trạng thái do luận án đề xuất. Chi tiết các bước làm việc của bộ điều khiển này đã được luận án thể hiện ở Thuật toán 2.7. Chất lượng làm việc tốt, đạt yêu cầu mong muốn của bộ điều khiển phản hồi đầu ra này cũng đã được luận án khẳng định thông qua mô phỏng trên các đối tượng: con lắc ngược và con lắc ngược quay.
- 3) Tiến hành thí nghiệm kiểm chứng lý thuyết trên mô hình thực: Đối tượng con lắc ngược quay tại Phòng thí nghiệm Đo lường – Điều khiển của Trường Đại học Kỹ thuật Công nghiệp có mô hình cho ở công thức (3.41). Kết quả thí nghiệm đã kiểm chứng tính đúng đắn của các thuật toán đề xuất trong luận án. Các kết quả thực nghiệm thu được xác nhận tính khả dụng vào thực tế của phương pháp như mong muốn và hoàn toàn phù hợp với nhận định lý thuyết.

KẾT LUẬN VÀ KIẾN NGHỊ

Những vấn đề đã làm được

Đề tài luận án liên quan tới bài toán điều khiển phản hồi đầu ra cho các đối tượng công nghiệp, có mô hình phi tuyến không liên tục:

$$\begin{cases} \underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{u}_k) + \underline{\xi}_k \\ \underline{y}_k = \underline{g}(\underline{x}_k, \underline{u}_k) + \underline{\zeta}_k \end{cases} \quad (4.1)$$

thỏa mãn điều kiện ràng buộc $\underline{u}_k \in U$, sao cho đầu ra \underline{y}_k của hệ bám ổn định được giá trị mẫu đặt trước, trong đó hệ (4.1) còn bị nhiễu tác động cả ở bên trong hệ thống bởi $\underline{\xi}_k$ (nhiều quá trình) và $\underline{\zeta}_k$ ở tín hiệu ra (nhiều đo).

Đề giải quyết được bài toán trên, luận án đã đặt ra hướng đi là sử dụng lọc Kalman mở rộng (EKF, UKF) để lọc nhiễu, đồng thời quan sát trạng thái hệ thống để cung cấp giá trị trạng thái quan sát được cho bộ điều khiển dự báo phản hồi trạng thái, tạo ra bộ điều khiển phản hồi đầu ra theo nguyên lý tách.

Với hướng đi như trên, luận án đã đạt được những kết quả như sau:

1) Trình bày lại được các phương pháp lọc Kalman mở rộng (EKF) và lọc UKF cho hệ phi tuyến dưới dạng thuật toán chi tiết. Thêm nữa luận án cũng đã bổ sung phương pháp ứng dụng Kalman tuyến tính (KF) để quan sát từng đoạn hệ phi tuyến theo nguyên lý tối ưu. Phương pháp đề xuất thêm này đã được luận án xây dựng chi tiết thành:

- a) Thuật toán 2.1 để quan sát trạng thái hệ song tuyến.
- b) Thuật toán 2.2 để quan sát trạng thái hệ phi tuyến.

Khả năng áp dụng của hai thuật toán trên vào thực tế cũng đã được luận án mô phỏng trên:

- Hệ song tuyến theo tín hiệu vào (2.14), (2.15) ở ví dụ 2.1 và ví dụ 2.2 và kết quả mô phỏng thu được đã xác nhận chất lượng tốt của bộ quan sát này.

2) Xây dựng được phương pháp điều khiển dự báo phản hồi trạng thái hệ phi tuyến trên cơ sở sử dụng mô hình dự báo tuyến tính từng đoạn với cửa sổ dự báo hữu hạn, mà cụ thể là đã xây dựng được các thuật toán:

- a) Thuật toán 2.3 và Thuật toán 2.4 để điều khiển phản hồi trạng thái hệ song tuyến.
- b) Thuật toán 2.5 và Thuật toán 2.6 để điều khiển phản hồi trạng thái hệ phi tuyến.

Khả năng áp dụng của các thuật toán trên vào thực tế cũng đã được luận án mô phỏng với: Hệ con lắc ngược và con lắc ngược quay.

Kết quả mô phỏng thu được đã xác nhận chất lượng tốt của bộ điều khiển dự báo phi tuyến sử dụng mô hình dự báo tuyến tính từng đoạn này, đúng như nhận định từ lý thuyết.

3) Xây dựng được bộ điều khiển dự báo phản hồi đầu ra theo nguyên lý tách trên cơ sở ghép chung bộ quan sát trạng thái Kalman và bộ điều khiển dự báo phản hồi trạng thái do luận án đề xuất. Chi tiết các bước làm việc của bộ điều khiển này đã được luận án thể hiện ở Thuật toán 2.7 và phiên bản chỉnh sửa của nó dành riêng cho hệ song tuyến. Đưa ra và chứng minh 01 định lý ổn định, điều kiện để bộ điều khiển dự báo phản hồi đầu ra làm hệ ổn định ISS (ổn định thực tế).

Khả năng áp dụng của thuật toán trên vào thực tế cũng đã được luận án mô phỏng thành công trên: Hệ con lắc ngược và con lắc ngược quay.

Kết quả mô phỏng thu được cũng đã khẳng định tính khả dụng cao của phương pháp vào thực tế công nghiệp.

4) Tiến hành thí nghiệm kiểm chứng lý thuyết trên mô hình thực: đối tượng con lắc ngược quay tại Phòng thí nghiệm Đo lường – Điều khiển của Trường Đại học Kỹ thuật Công nghiệp. Kết quả thí nghiệm đã kiểm chứng tính đúng đắn của các thuật toán đề xuất trong luận án đồng thời khẳng định tính khả dụng vào thực tế của phương pháp đề xuất và hoàn toàn phù hợp với nhận định lý thuyết.

Các vấn đề còn tồn tại và hướng nghiên cứu tiếp theo

Có ba vấn đề còn tồn tại của luận án và cũng sẽ là hướng nghiên cứu tiếp theo của tác giả luận án trong tương lai. Đó là:

- 1) Mặc dù chất lượng bám ổn định tốt của những bộ điều khiển dự báo phản hồi đầu ra trên cơ sở sử dụng lọc Kalman mở rộng và bộ điều khiển dự báo phản hồi trạng thái nhờ tuyến tính hóa từng đoạn mô hình dự báo phi tuyến đã được khẳng định thông qua thực nghiệm mô phỏng với một số đối tượng công nghiệp, song vẫn còn thiếu phần chứng minh lý thuyết chặt chẽ cho nó. Bởi vậy vấn đề tồn tại này sẽ là một trong các hướng nghiên cứu tiếp theo của tác giả.
- 2) Từ kết quả của Thuật toán 2.5 và Thuật toán 2.6 cho lớp đối tượng phi tuyến (2.39), (2.49), tác giả nhận thấy hai thuật toán này hoàn toàn có thể mở rộng được cho cả lớp đối tượng phi tuyến trễ đầu vào $\underline{u}_{k-\tau}$. Đó cũng là hướng nghiên cứu tiếp theo nữa của tác giả luận án trong tương lai.
- 3) Lớp đối tượng phi tuyến mà luận án đề cập đều là không liên tục (4.1) có nhiều $\underline{\xi}_k, \underline{\zeta}_k$ cộng tính trong mô hình, trong khi đối tượng công nghiệp luôn tồn tại ở dạng liên tục theo thời gian. Việc lượng tử hóa mô hình liên tục theo thời gian để có mô hình không liên tục tương ứng phục vụ việc thiết kế bộ điều khiển không thể tránh khỏi sự ảnh hưởng của sai lệch mô hình đối với chất lượng điều khiển. Vì vậy trong tương lai, tác giả sẽ nghiên cứu phát triển tiếp các thuật toán điều khiển đã được luận án xây dựng để có thể áp dụng trực tiếp được cho hệ liên tục có nhiều $\underline{\xi}, \underline{\zeta}$ lan truyền phi tuyến trong mô hình dạng tổng quát:

$$\begin{cases} \frac{d\underline{x}}{dt} = \underline{f}(\underline{x}, \underline{u}, \underline{\xi}) \\ \underline{y} = \underline{g}(\underline{x}, \underline{u}, \underline{\zeta}) \end{cases} \quad (4.2)$$

nhằm nâng cao được hơn nữa chất lượng điều khiển trong thực tế công nghiệp.

Các công trình khoa học đã công bố

1. Hoàng Đức Quỳnh, Nguyễn Thị Việt Hương và Nguyễn Doãn Phước(2013), “Nhận dạng trạng thái hệ cầu treo 2 chiều bằng bộ quan sát Kalman rời rạc”, *Tạp chí Khoa học và Công nghệ, Đại học Thái Nguyên*, tập 106(số 06), tr.15-21.
2. Hoàng Đức Quỳnh và Nguyễn Doãn Phước(2013), “Xây dựng mô hình phi tuyến cho hệ điều khiển mức-nhiệt độ và phân tích hệ thống”, *Tạp chí Khoa học và Công nghệ, Đại học Thái Nguyên*, tập 110(số 10), tr. 15-26.
3. Hoàng Đức Quỳnh, Nguyễn Doãn Phước và Nguyễn Quang Hùng(2014), “Thiết kế bộ điều khiển dự báo cho đối tượng van công nghiệp”, *Tạp chí nghiên cứu khoa học và công nghệ quân sự, Viện Khoa học Công nghệ Quân sự*, đặc san tự động hóa, tr.12-18.
4. Hoàng Đức Quỳnh, Nguyễn Đình Hòa và Nguyễn Doãn Phước(2014), “Một số phương pháp thiết kế bộ điều khiển dự báo cho đối tượng van mở nhanh”, *Tạp chí Khoa học và Công nghệ, Đại học Thái Nguyên*, tập 122(số 08), tr.167-171.
5. Hoàng Đức Quỳnh, Nguyễn Văn Chí, Nguyễn Như Hiền và Nguyễn Doãn Phước(2016), “Ứng dụng lọc Kalman mở rộng trong thiết kế bộ điều khiển dự báo phản hồi đầu ra cho đối tượng con lắc ngược quay (Rotary Inverted Pendulum)”, *Tạp chí Khoa học và Công nghệ, Đại học Thái Nguyên*, tập 151(số 06), tr.185-191.
6. Nguyễn Doãn Phước, Nguyễn Đức Anh, Vũ Tiến Thành, Phạm Văn Hùng và Hoàng Đức Quỳnh(2016), “Robust output tracking control with constraints for nonlinear system base on piecewise linear quadratic optimization and its perspective for practical application”, *Workshop on Vietnamese – German Technology Cooperation and Cultural Exchange*, pp.57-67.
7. Hoàng Đức Quỳnh, Nguyễn Như Hiền và Nguyễn Doãn Phước(2016), “Thiết kế bộ điều khiển dự báo phản hồi đầu ra để điều khiển bền vững động cơ không đồng bộ ba pha”, *Tạp chí nghiên cứu khoa học và công nghệ quân sự, Viện Khoa học Công nghệ Quân sự*, số 44, tháng 8-2016, tr.13-22.

Tài liệu tham khảo

Tiếng Việt

1. Đỗ Thị Tú Anh(2014), *Điều khiển dự báo phản hồi đầu ra theo nguyên lý tách cho hệ phi tuyến*, Luận án Tiến sỹ Kỹ thuật, Trường Đại học Bách khoa, Hà Nội.
2. Đỗ Thị Tú Anh và Nguyễn Doãn Phước(2014), “Ổn định hóa hệ song tuyến liên tục với bộ điều khiển dự báo”, *Tạp chí Khoa học và Công nghệ, Đại học Thái Nguyên*, tập 20(số 06), tr.73-79.
3. Đặng Trung Dũng(2014), *Ứng dụng vi mạch điện tử Andruino Mega trong cài đặt thuật toán điều khiển ổn định vị trí con lắc ngược quay*, Luận văn Thạc sỹ Kỹ thuật, Trường Đại học Kỹ thuật Công nghiệp – ĐHTN, Thái Nguyên.
4. Nguyễn Thị Mai Hương(2016), *Nghiên cứu xây dựng thuật toán điều khiển theo mô hình cho đối tượng phi tuyến liên tục*, Luận án Tiến sỹ Kỹ thuật, Trường Đại học Kỹ thuật Công nghiệp – ĐHTN, Thái Nguyên.
5. Bùi Quốc Khánh và cs(2014), *Điều khiển quá trình*, NXB KH&KT, Hà Nội.
6. Nguyễn Thúc Loan, Nguyễn Thị Phương Hà, Huỳnh Thái Hoàng(2003): “Điều khiển dự báo hệ phi tuyến dựa vào mô hình mờ”, *Tạp chí Công nghệ thông tin & Truyền thông – Bộ Thông tin và Truyền thông*, chuyên san nghiên cứu khoa học(số 10), tr.67-72.
7. Lê Thị Huyền Linh(2015), *Nghiên cứu ứng dụng lý thuyết điều khiển hiện đại xây dựng mô hình trong điều khiển dự báo phi tuyến*, Luận án Tiến sỹ Kỹ thuật, Trường Đại học Kỹ thuật Công nghiệp – ĐHTN, Thái Nguyên.
8. Nguyễn Doãn Phước và Đỗ Thị Tú Anh(2013), *Giới thiệu về Điều khiển dự báo. Phần I: Hệ tuyến tính*, Hội nghị Khoa học Khoa Điện tử, Trường ĐH Kỹ thuật Công nghiệp – ĐHTN.
9. Nguyễn Doãn Phước(2009), *Lý thuyết điều khiển tuyến tính*, In lần thứ 4, NXB Khoa học và Kỹ thuật, Hà Nội.

10. Nguyễn Doãn Phước(2009), *Lý thuyết điều khiển nâng cao*, In lần thứ 4, NXB Khoa học và Kỹ thuật, Hà Nội.
11. Nguyễn Doãn Phước(2012), *Phân tích và điều khiển hệ phi tuyến*, NXB Bách Khoa, Hà Nội.
12. Nguyễn Doãn Phước(2015), *Tối ưu hóa trong điều khiển và điều khiển tối ưu*, NXB Bách Khoa, Hà Nội.
13. Hoàng Minh Sơn(2009), *Cơ sở hệ thống điều khiển quá trình*, NXB Bách Khoa, Hà Nội.
14. 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 sỹ kỹ Thuật, Trường Đại học Bách khoa Hà Nội.

Tiếng Anh

15. Abedi, M.; Yaghoobi,M.; Raesian,N.(2013), *Optimization of model predictive controller parameters based on Imperialist Competitive algorithm*, Electrical Engineering (ICEE), 2013 21st Iranian Conference.
16. Abdollahpouri, M. ; Khaki-Sedigh, A. ; Fatehi, A.(2012), *Lyapunov based multiple model predictive control: An LMI approach*, American Control Conference (ACC).
17. Adetola, V. and M. Guay(2003), *Nonlinear output feedback receding horizon control of sampled data systems*, In: Proc. of the American Control Conference, Denver, Colorado. pp. 4914–4919.
18. Ali,E. and Zafiriou,E.(1993), “Optimization based tuning of nonlinear model predictive control with states estimation”, *Journal of Process Control*, vol.3(2), pp.97-107.
19. Anders Robertsson(1999), *On ObserverBased Control of Nonlinear Systems*, Printed in Sweden by Wallin & Dahlholm Boktryckeri AB Lund 1999.

20. Anderson,R.T. and Li,P.Y.(2002), “Mathematical modelling of a two spool flow control servovalve using pressure control pilot”, *Journal of Dynamic Systems, Measurement and Control, Trans. on ASME, Vol.124*, pp.420-427.
21. Bela G. Liptak(Chief Editor 1995): *Process Measurement and Analysis - Handbook*. Pennsylvania Chilton Book Company, 3. Edition.
22. Boyd, S. and Vandenberghe, L.(2004), *Convex optimization*, Cambridge University Press.
23. Burl,B.J.(1999), *Linear Optimal control, H_2 and H_∞ Methods*, Addison Wesley.
24. Byron Bird; Warren Steward and Edwin Lichtfoot(1960), *Transport phenomena*. John Wiley&Sons.
25. Camacho, E. and Bordons, C.(1999), *Model predictive control*, Springer.
26. Chalupa,P.; Novak,J. and Bobal,V.(2011), “Mathematical model of control valve of three-tank system”, *Int. Journal of Mechanics*, Vol.5, Issue 4, pp.310-317.
27. Chapellat,H.;Dahleh,M. and Bhattacharyya,S.(1990), “Robust Stability under structured and under structured Perturbations”, *IEEE trans. on AC*, vol.35, pp 1100-1107
28. Chengli Su; Jinwei Ma(2012), “Nonlinear Predictive Control Using Fuzzy Hammerstein Model and Its Application to CSTR Process”, *AASRI Procedia* Volume 3, pp.8–13, Conference on Modelling, Identification and Control.
29. David A. Copp.(2016): *Nonlinear Output-Feedback Model Predictive Control with Moving Horizon Estimation*, Dissertation, University of California, Santa Barbara.
30. Engell,S.(1995), *Entwurf nichtlinearer Regelungen*, Oldenburg Verlag.
31. E. Kiriya and M. Buehler.(2002), *Three-state Extended Kalman Filter for Mobile Robot Localization*,
http://kom.aau.dk/group/05gr999/reference_material/filtering/eKf-3state.pdf.

Truy cập ngày 20/7/2015.

32. Findeisen,R.; Imsland,L.; Allgower,F. and Foss,B.A.(2003), “States and output feedback nonlinear model predictive controll: An Overview”, *European Journal of Control*, Vol.9, pp.179-195.
33. Foley,J; A. van Dam; Feiner,S.; Hughes, J.; Phillips, R.; Smith, R. and Cheeseman, P.(1987), “On the estimation and representation of spatial uncertainty”, *International Journal of Robotics Research* 5,4, pp.56-68
34. Kalman,R.(1960), “A new approach to linear filtering and prediction problems”, *Transactions ASME Journal of Basic Engineering* 82, pp.35-44.
35. K. Ogata.(1995), *Electromechanical System, Model Control Engineering*, Prentice Hall, pp.118-121.
36. Kumar,E.V. and Jerome,J.(2013), “Robust LQR Controller Design for Stabilizing and Trajectory tracking of Inverted Pendulum”, *Procedia Engineering*, Vol.64, pp.169-178
37. Gary Bishop and Greg Welch.(2001), *An Introduction to the Kalman Filter*, University of Norh Carolina SIGGRAPH 2001 course notes. ACM Inc., North Carolina.
38. Giancarlo Marafioti(2010), *Enhanced Model Predictive Control: Dual Control Approach and State Estimation Issues*. Doctoral thesis at Norwegian University of Science and Technology.
39. Grewal, M.S. and Andrews, A.P.(2001), *Kalman filtering: Theory and Practice using MatLab*, John Wiley & Sons.
40. Grüne,L. and Pannek,J.(2010), *Nonlinear model predictive control. Theory and Algorithms*. Springer.
41. H.L. Langhaar.(1962), *Energy Methods in Applied Mechanics*, John Wiley & Sons, pp.233-249.
42. Huiping Li.(2013), *Filtering and Model Predictive Control of Networked Nonlinear Systems*, Doctor of Philosophy in the Department of Mechanical Engineering, University of Victoria.
43. 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

- System”, *International Journal of Mechanical Engineerings and Robotics Reseach*, Vol.4, No.4, pp.209-203.
44. Julier,S.J. and Uhlmann,J.K.(1997), “A new extention of the Kalman filter to nonlinear systems”, *Int. Symp. Aerospace/Defense sensing, Signal and Control. Signal Procesing, Sensor Fussion and Target Recognition*.
 45. Kandepu,R.; Foss,B. and Imsland,L.(2008), “Applying the unscented Kalman filter for nonlinear state estimation”, *Journal of Process Control*, 2008, Nov.13.
 46. K.Futura, M.Yamakita.(1992), “Swing up control of inverted pedulum using pseudo-state feedback”, *Proc Instn Mech Engs*, Vol 206.
 47. Lewis,F.L.; Vrabi,D.L. and Syrmos,V.L.(2012), *The tracking problem and other LQR extention*, John Wiley and Sons, Inc.
 48. Maybeck,P. S.(1979), *Stochastic models, estimation and control*, Academic Press, Inc., New York, USA.
 49. Mehdi Maasoumy Haghghi(2013), *Controlling Energy-Efficient Buildings in the Context of Smart Grid: A Cyber Physical System Approach*, Doctor of Philosophy in Engineering - Mechanical Engineering.University of California, Berkeley.
 50. Meng Zhao, Bao-Cang Ding(2013), “A contractive sliding-mode MPC algorithm for nonlinear discrete-time systems”, *International Journal of Automation and Computing*, April 2013, Volume 10, Issue 2, pp 167-172
 51. MPC. From ControlsWiki, <https://controls.engin.umich.edu/wiki/index.php/MPC>, 20/7/2014.
 52. Muske,K.R. and Rawlings,J.B.(1993), “Model predictive control with linear models”, *AIChE J.*, 39(2):262-287, 1993.
 53. Nocedal,J. and Wright,S.J.(1996), *Numerical Optimization*, Springer-New York.
 54. Patrasciois,C; Panaitescu,C. and Paraschiv,N.(2010), “Control Valve-Modelling and Simulation”, *Proceedings of 5th WSEAS Int. Conference on Dynamical System and Contrrol*, pp.63-68.

55. Patwardhan, S.C.; Madhavan, K.P.(1995), “Nonlinear predictive control of an exothermic CSTR using recursive quadratic state space models”, *Control Applications, 1995., Proceedings of the 4th IEEE Conference on*, Date of Conference: 28-29 Sep 1995.
56. Payam,A.F.(2008), “An adaptive Input-Output feedback linearization controller for Doubly-Fed Induction Machine Drives”, *Serbian Journal of Electrical Engineering*, Vol.5, No.1, pp.139-154.
57. Phuoc, N.D. and Ha, L.T.T(2015), “Constrained Output Tracking Control for Time-Varying Bilinear Systems via RHC with Infinite Prediction Horizon”, *Journal of Computer Science and Cybernetics*, Vol.31, No.2, pp. 97-106
58. Qin,S.J. and Badgwell,T.A.: An overview of industrial model predictive control technology. In J.C. Kantor.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.52.8909&rep=rep1&type=pdf>. Truy cập ngày 20/7/2015.
59. Rafał Noga.(2015), *Non-linear Model Predictive Control for cooling strings of superconducting magnets using superfluid helium*. Dissertation, Doctor of Philosophy in Department of Systems Engineering and Automatic Control University of Valladolid.
60. Roset,B.J.P; Lazar,M; Heemels,W.P.M.H. and Nijmeijer,H.(2007), *Design of stabilizing output feedback nonlinear model predictive controllers with an application to DC-DC converters*, Proc. of European Control Conference, Greece, July 2-5, pp.2999-3004.
61. Rudolph van der Merwe, Alex T. Nelson, Eric Wan(2004), "An Introduction to Kalman Filtering", *OGI School of Science & Engineering lecture*. Oregon Health & Sciences University.
62. Rui Huang(2010), *Nonlinear Model Predictive Control and Dynamic Real Time Optimization for Large-scale Processes*, Doctor of Philosophy in Chemical Engineering, Carnegie Mellon University.
63. Seborg Dale E., Edgar Thomas F. and Mellichamp Ducan A.(2004), *Process Dynamic and Control*, John Wiley&Sons Inc.

64. Tan Kok Chye, Teo Chun Sang.(1998/99),“Rotary Inverted Pendulum”,
School of electrical and electronic engineering, Nanyang Technological
University.
65. Yan, J. and R. R. Bitmead(2002), “Model predictive control and state
estimation: A network example”, In: IFAC World Congress. Barcelona, Spain.

PHỤ LỤC

P1 Điều khiển dự báo phản hồi trạng thái hệ con lắc ngược

Chi tiết phần mã nguồn chương trình cài đặt trên MatLab để điều khiển dự báo phản hồi trạng thái hệ con lắc ngược (3.1) theo **Thuật toán 2.6** do luận án đề xuất có dạng như sau:

```
%MPC PHAN HOI TRANG THAI CON LAC NGUOC
clear all;clc;
T=0.001;           %thoi gian trich mau
N_sim=50;%thoi gian mo phong

q=1e-4;
r=1e-4;
%cac thong so cua doi tuong
t=[0:N_sim];      %ma tran thoi gian

%CAC THONG SO BAN DAU
%cac thong so cua doi tuong
L=0.27;
mh=0.1;
mc=1.2;
g=9.8;

N=10;
udk=1;
xtrue=[2;1;pi/6;1];

y=zeros(N_sim,1);
u=zeros(N_sim+1,1);
```

```

noisex=wgn(N_sim+1,4,q,'linear');%0.05*randn(N_sim+1,1);
noisey=wgn(N_sim+1,1,r,'linear');
delta=10*eye(N);
Lk=10*eye(N);
udk(1,1)=u(1,1);

```

```

T11=-
g*cos([0,0,1,0]*xtrue)+L*([0,0,0,1]*xtrue)*([0,0,0,1]*xtrue);
T12=mh*sin([0,0,0,1]*xtrue);
T1=T11*T12;
M1=mc+mh*sin([0,0,1,0]*xtrue)*sin([0,0,1,0]*xtrue);
T21=g-
L*([0,0,0,1]*xtrue)*([0,0,0,1]*xtrue)*cos([0,0,1,0]*xtrue);
T22=mh*sin([0,0,1,0]*xtrue);
T23=g*mc*sin([0,0,1,0]*xtrue);
T2=T21*T22+T23;
M2=L*mc+L*mh*sin([0,0,1,0]*xtrue)*sin([0,0,1,0]*xtrue);
dT11=g*sin([0,0,1,0]*xtrue)*[0,0,1,0]+2*L*([0,0,1,0]*xtrue)*[0,0,1,0];
dT12=mh*cos([0,0,0,1]*xtrue)*[0,0,0,1];
dT1=dT11*T12+T11*dT12;
dM1=2*mh*cos([0,0,1,0]*xtrue)*[0,0,1,0];
dT21=-
2*L*([0,0,0,1]*xtrue)*cos([0,0,1,0]*xtrue)*[0,0,0,1]-(g-
L*([0,0,0,1]*xtrue)*([0,0,0,1]*xtrue)*sin([0,0,1,0]*xtrue))*[0,0,1,0];
dT22=mh*cos([0,0,1,0]*xtrue)*[0,0,1,0];

```



```

dT23=g*mc*cos([0,0,1,0]*xtrue)*[0,0,1,0];
dT2=dT21*T22+T21*dT22+dT23;
dM2=2*L*mh*cos([0,0,1,0]*xtrue)*[0,0,1,0];
S1=[0,0,0,0];
S2=((M1*dT1-T1*dM1)/(M1*M1))-((dM1*udk)/(M1*M1));
S3=[0,0,0,0];
S4=((M2*dT2-T2*dM2)/(M2*M2))-
((M2*sin([0,0,1,0]*xtrue)*[0,0,1,0])-
(cos([0,0,1,0]*xtrue)*dM2))*udk/(M2*M2);
A_lt=[S1;S2;S3;S4];
B_lt=[0;1/M1;0;-cos([0,0,1,0]*xtrue)/M2];

%A_lt=[0,0,0,0;0,-((g*cos(xtrue(3))-
L*xtrue(4)*xtrue(4))*mh*sin(xtrue(4)))/((mc+mh*sin(xtrue
(3))*sin(xtrue(3)))*xtrue(2)),0,0;0,0,0,0;0,0,0,((g-
L*xtrue(4)*xtrue(4)*cos(xtrue(3)))*mh*sin(xtrue(3))+g*mc
*sin(xtrue(3)))/((L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))
*xtrue(4))];
%B_lt=[0;1/(mc+mh*sin(xtrue(3))*sin(xtrue(3)));0;-
cos(xtrue(3))/(L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))]];
C_lt=[0,0,1,0];
D_lt=0;
[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');

%%%%%%

wk=zeros(N_sim+N-1,1);
for i=1:N_sim+N-1
    wk(i)=pi/6;
end

```

```

w=zeros(N,1);

for k=1:N_sim

h(1,:)=C_d;
E(1,:)=C_d*A_d;%hang dau tien cua ma tran E

for i=2:N
h(i,:)=h(i-1,:)*A_d;
E(i,:)= E(i-1,:)*A_d;%Ma tran E
end
v(1:N,:)=[h*B_d];
F=zeros(N,N); %Khai bao kich thuoc cua F
%F1(1,1:5)=v; % hang thu nhat 5 cot dau tien cua F

for i=1:N
F1(i,1)=v(i,1);% cot dau tien cua ma tran F
delta(i,i)=N-i+1;
end

F(1:N,1)=F1;

for i=1:N-1
    F(1:N,i+1)=[zeros(i,1);F1(1:N-i,1)];
end
w(1:N,1)=wk(k:k+N-1);
z=w-E*xtrue;

p_sao=inv(F'*delta*F+Lk)*F'*delta*z;
udk=p_sao(1,1);

```

```

udknoise=udk+noisex(k+1);
%phan hoi trang thai
xtrue=A_d*xtrue+B_d*udk;
yknoise=C_lt*xtrue+noisy(k);

y(k+1)=yknoise;
u(i+1,1)=udknoise;

T11=-
g*cos([0,0,1,0]*xtrue)+L*([0,0,0,1]*xtrue)*([0,0,0,1]*xtrue);
T12=mh*sin([0,0,0,1]*xtrue);
T1=T11*T12;
M1=mc+mh*sin([0,0,1,0]*xtrue)*sin([0,0,1,0]*xtrue);
T21=g-
L*([0,0,0,1]*xtrue)*([0,0,0,1]*xtrue)*cos([0,0,1,0]*xtrue);
T22=mh*sin([0,0,1,0]*xtrue);
T23=g*mc*sin([0,0,1,0]*xtrue);
T2=T21*T22+T23;
M2=L*mc+L*mh*sin([0,0,1,0]*xtrue)*sin([0,0,1,0]*xtrue);
dT11=g*sin([0,0,1,0]*xtrue)*[0,0,1,0]+2*L*([0,0,1,0]*xtrue)*[0,0,1,0];
dT12=mh*cos([0,0,0,1]*xtrue)*[0,0,0,1];
dT1=dT11*T12+T11*dT12;
dM1=2*mh*cos([0,0,1,0]*xtrue)*[0,0,1,0];
dT21=-
2*L*([0,0,0,1]*xtrue)*cos([0,0,1,0]*xtrue)*[0,0,0,1]-(g-
L*([0,0,0,1]*xtrue)*([0,0,0,1]*xtrue)*sin([0,0,1,0]*xtrue))*[0,0,1,0];
dT22=mh*cos([0,0,1,0]*xtrue)*[0,0,1,0];

```

```

dT23=g*mc*cos([0,0,1,0]*xtrue)*[0,0,1,0];
dT2=dT21*T22+T21*dT22+dT23;
dM2=2*L*mh*cos([0,0,1,0]*xtrue)*[0,0,1,0];
S1=[0,0,0,0];
S2=((M1*dT1-T1*dM1)/(M1*M1))-((dM1*udknoise)/(M1*M1));
S3=[0,0,0,0];
S4=((M2*dT2-T2*dM2)/(M2*M2))-
((M2*sin([0,0,1,0]*xtrue)*[0,0,1,0])-
(cos([0,0,1,0]*xtrue)*dM2))*udknoise/(M2*M2);
A_lt=[S1;S2;S3;S4];
B_lt=[0;1/M1;0;-cos([0,0,1,0]*xtrue)/M2];
%A_lt=[0,0,0,0;0,-((g*cos(xtrue(3))-
L*xtrue(4)*xtrue(4))*mh*sin(xtrue(4)))/((mc+mh*sin(xtrue
(3))*sin(xtrue(3)))*xtrue(2)),0,0;0,0,0,0;0,0,0,((g-
L*xtrue(4)*xtrue(4)*cos(xtrue(3)))*mh*sin(xtrue(3))+g*mc
*sin(xtrue(3)))/((L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))
*xtrue(4))];
%B_lt=[0;1/(mc+mh*sin(xtrue(3))*sin(xtrue(3)));0;-
cos(xtrue(3))/(L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))]];
C_lt=[0,0,1,0];
D_lt=0;
[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');

end

t=0:N_sim;
t1=0:N_sim+N-2;
figure(1)
plot(t,y,'r-.o','LineWidth',3.5);grid on;
hold on
plot(t1,wk,'r','LineWidth',3.5);grid on;

```

```

legend('MPC phan hoi trang thai','gia tri vi tri goc
dat');
xlabel('k');
ylabel('vi tri goc (rad)');

figure(2)

plot (t,u,'r-o','LineWidth',1.5);grid on;
xlabel('k');
ylabel('control signal');

```

Chi tiết phần mã nguồn chương trình cài đặt trên MatLab để điều khiển dự báo phản hồi trạng thái hệ con lắc ngược (3.1) theo **Thuật toán 2.4** do luận án đề xuất có dạng như sau:

```

%MPC PHAN HOI TRANG THAI CON LAC NGUOC
clear all;clc;
T=0.001;          %thoi gian trich mau
N_sim=50;%thoi gian mo phong

q=1e-4;
r=1e-4;
%cac thong so cua doi tuong
t=[0:N_sim];     %ma tran thoi gian

%CAC THONG SO BAN DAU
%cac thong so cua doi tuong
L=0.27;
mh=0.1;
mc=1.2;

```

```

g=9.8;

N=10;

xtrue=[2;2;1;1];
A_lt=[xtrue(2)/xtrue(1),0,0,0;0,-((g*cos(xtrue(3)))-
L*xtrue(4)*xtrue(4))*mh*sin(xtrue(4)))/((mc+mh*sin(xtrue
(3))*sin(xtrue(3)))*xtrue(2)),0,0;0,0,xtrue(4)/xtrue(3),
0;0,0,0,((g-
L*xtrue(4)*xtrue(4)*cos(xtrue(3)))*mh*sin(xtrue(3))+g*mc
*sin(xtrue(3)))/((L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))
*xtrue(4))];
B_lt=[0;1/(mc+mh*sin(xtrue(3))*sin(xtrue(3)));0;-
cos(xtrue(3))/(L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))];
C_lt=[0,0,1,0];
D_lt=0;
[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');

%%%%%%%%

wk=zeros(N_sim+N-1,1);
for i=1:N_sim+N-1
    wk(i)=pi/6;
end
w=zeros(N,1);
y=zeros(N_sim,1);
u=zeros(N_sim+1,1);
noisex=wgn(N_sim+1,4,q,'linear');%0.05*randn(N_sim+1,1);
noisey=wgn(N_sim+1,1,r,'linear');
delta=10*eye(N);
Lk=10*eye(N);

```

```

udk(1,1)=u(1,1);
%usbeta(1,1)=u(2,1);

for k=1:N_sim

h(1,:)=C_d;
E(1,:)=C_d*A_d;%hang dau tien cua ma tran E

for i=2:N
h(i,:)=h(i-1,:)*A_d;
E(i,:)= E(i-1,:)*A_d;%Ma tran E
end
v(1:N,:)=[h*B_d];

F=zeros(N,N); %Khai bao kich thuoc cua F
%F1(1,1:5)=v; % hang thu nhat 5 cot dau tien cua F

for i=1:N
F1(i,1)=v(i,1);% cot dau tien cua ma tran F
delta(i,i)=N-i+1;
end

F(1:N,1)=F1;

for i=1:N-1
    F(1:N,i+1)=[zeros(i,1);F1(1:N-i,1)];
end
w(1:N,1)=wk(k:k+N-1);
z=w-E*xtrue;

p_sao=inv(F'*delta*F+Lk)*F'*delta*z;

```

```

udk=p_sao(1,1);
udknoise=udk+noisex(k+1);
%phan hoi trang thai
xtrue=A_d*xtrue+B_d*udk;
yknoise=C_lt*xtrue+noisy(k);

y(k+1)=yknoise;
u(i+1,1)=udknoise;

A_lt=[xtrue(2)/xtrue(1),0,0,0;0,-((g*cos(xtrue(3)))-
L*xtrue(4)*xtrue(4))*mh*sin(xtrue(4)))/((mc+mh*sin(xtrue
(3))*sin(xtrue(3)))*xtrue(2)),0,0;0,0,xtrue(4)/xtrue(3),
0;0,0,0,((g-
L*xtrue(4)*xtrue(4)*cos(xtrue(3)))*mh*sin(xtrue(3))+g*mc
*sin(xtrue(3)))/((L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))
*xtrue(4))];
B_lt=[0;1/(mc+mh*sin(xtrue(3))*sin(xtrue(3)));1e+4;-
cos(xtrue(3))/(L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))];
C_lt=[0,0,1,0];
D_lt=0;
[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');

end

t=0:N_sim;
t1=0:N_sim+N-2;
figure(1)
plot(t,y,'b-o','LineWidth',2.5);grid on;
hold on
plot(t1,wk,'b','LineWidth',2.5);grid on;

```



```

legend('MPC phan hoi trang thai','gia tri vi tri goc
dat');
xlabel('k');
ylabel('vi tri goc (rad)');

figure(2)

plot (t,u,'r-o','LineWidth',1.5);grid on;
xlabel('k');
ylabel('control signal');

```

Dữ liệu đầu vào khi hệ có một đầu ra là góc lắc và công thức lượng tử hóa (3.10)

```

A_lt=[0,1,0,0;0,0,0,-((g*cos(xtrue(3))-
L*xtrue(4)*xtrue(4))*mh*sin(xtrue(4)))/((mc+mh*sin(xtrue
(3))*sin(xtrue(3)))*xtrue(4));0,0,0,1;0,0,((g-
L*xtrue(4)*xtrue(4)*cos(xtrue(3)))*mh*sin(xtrue(3))+g*mc
*sin(xtrue(3)))/((L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))
*xtrue(3)),0];
B_lt=[0;1/(mc+mh*sin(xtrue(3))*sin(xtrue(3)));0;-
cos(xtrue(3))/(L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))];
C_lt=[0,0,1,0];
D_lt=0;

```

Dữ liệu đầu vào khi hệ có một đầu ra là góc lắc và công thức lượng tử hóa (3.16)

```

A_lt=[0,0,0,0;0,0,0,-((g*cos(xtrue(3))-
L*xtrue(4)*xtrue(4))*mh*sin(xtrue(4)))/((mc+mh*sin(xtrue
(3))*sin(xtrue(3)))*xtrue(4));0,0,0,0;0,0,((g-

```

```

L*xtrue(4)*xtrue(4)*cos(xtrue(3))*mh*sin(xtrue(3))+g*mc
*sin(xtrue(3)))/((L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))
*xtrue(3)),0];
B_lt=[0;1/(mc+mh*sin(xtrue(3))*sin(xtrue(3)));0;-
cos(xtrue(3))/(L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))];
C_lt=[0,0,1,0];
D_lt=0;

```

P2 Quan sát trạng thái hệ con lắc ngược

Toàn bộ phần mã nguồn chương trình cài đặt trên MatLab để quan sát trạng thái hệ con lắc ngược (3.1) theo **Thuật toán 1.9** có dạng như sau:

```

%Kalman mở rộng loại 3 (EKF3): CON LAC NGUOC 2 đầu ra
clear all;clc;
T=0.001;           %thời gian trích mẫu
N_sim=100;%thời gian mô phỏng
%q=0;
q=1e-6;
r=1e-5;
Q=[q 0 0 0;0 q 0 0;0 0 q 0;0 0 0 q];   %ma trận tương
quan của w
R=[r];           %ma trận tương quan của v

Pk=0.1;           %điều kiện đầu của sai lệch ước lượng
u=200*ones(1,N_sim+2); %biên đầu vào
t=[0:N_sim];     %ma trận thời gian

%CAC THONG SO BAN DAU
%cac thông số của đối tượng
L=0.27;
mh=0.1;

```

```

mc=1.2;
g=9.8;
%N=10;
x=[1.5;1;0.5;1];      %dieu kien dau cua trang thai
thuc
xes=[1.5;1;0.5;1];    %dieu kien dau cua trang thai
uoc luong
xtrue=[1.5;1;0.5;1];  %bien trang thai thuc
y_thuc=0;

for i=1:N_sim+1
A_lt=[0,1,0,0;-((g*cos(xtrue(3)))-
L*xtrue(4)^2)*mh*sin(xtrue(3)))/((mc+mh*sin(xtrue(3))^2)
*xtrue(1)),0,0,0;0,0,0,1;0,0,((g-
L*xtrue(4)^2*cos(xtrue(3)))*mh*sin(xtrue(3))+g*mc*sin(xt
rue(3)))/((L*mc+L*mh*sin(xtrue(3))^2)*xtrue(3)),0];
B_lt=[0;1/(mc+mh*sin(xtrue(3))*sin(xtrue(3)));0;-
cos(xtrue(3))/(L*mc+L*mh*sin(xtrue(3))*sin(xtrue(3)))]];
C_lt=[1,0,0,0;0,0,1,0];
D_lt=[0;0];
[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');
%A_d=1+T*A_lt; B_d=T*B_lt; C_d=C_lt; D_d=D_lt;

    x1(i)=x(1,1);    %them vao ma tran x1
    x2(i)=x(2,1);    %them vao ma tran x2
    x3(i)=x(3,1);
    x4(i)=x(4,1);

    %kich hoat he thong
    x=A_d*x+B_d*u(i)+wgn(4,1,q,'linear');
    y=C_d*x+wgn(1,1,r,'linear');

```

```

%trang thai thuc
xt1(i)=xtrue(1,1);    %them vao ma tran xt1
xt2(i)=xtrue(2,1);    %them vao ma tran xt2
xt3(i)=xtrue(3,1);    %them vao ma tran xt3
xt4(i)=xtrue(4,1);    %them vao ma tran xt4

xtrue=A_d*xtrue+B_d*u(i);

%cac buoc tinh toan
xes1(i)=xes(1,1);    %them vao ma tran xes1
xes2(i)=xes(2,1);    %them vao ma tran xes2
xes3(i)=xes(3,1);    %them vao ma tran xes3
xes4(i)=xes(4,1);    %them vao ma tran xes4

Fk=A_d;
xes=A_d*xes+B_d*u(i);
Gk=C_d;
Pk=Fk*Pk*Fk'+Q;
K_ekf=Pk*Gk'*inv(Gk*Pk*Gk'+R);
Pk=(eye(size(K_ekf*Gk))-K_ekf*Gk)*Pk;
xes=xes+K_ekf*(y-Gk*xes);
y_es=C_lt*xes+wgn(1,1,r,'linear');

end

%ve do thi

figure(1);
subplot(2,1,1)

plot(t,xt1,'b','LineWidth',3.5);grid on;

```

```

hold on
plot(t,xes1,'r--','LineWidth',3.5);grid on;
title('uoc luong quang duong xe di duoc')
legend('yc true','yc estimate','makersize',18);
xlabel('k');ylabel('yc[m]');

%figure(2);
subplot(2,1,2)

plot(t,xt2,'b','LineWidth',3.5);grid on;
hold on
plot(t,xes2,'r--','LineWidth',3.5);grid on;
title('uoc luong van toc')
legend('v true','v estimate');
xlabel('k');ylabel('v[m/s]');

figure(2);
subplot(2,1,1)

plot(t,xt3,'b','LineWidth',3.5);grid on;
hold on
plot(t,xes3,'r--','LineWidth',3.5);grid on;
title('uoc luong goc lech Phi')
legend('Phi true','Phi estimate','makersize',18);
xlabel('k');ylabel('Phi[rad]');

subplot(2,1,2)
%figure(3);
plot(t,xt4,'b','LineWidth',3.5);grid on;
hold on
plot(t,xes4,'r--','LineWidth',3.5);grid on;

```

```

title('uoc luong van toc goc')
legend('van toc goc true','van toc goc
estimate','makersize',18);
xlabel('k');ylabel('van toc goc[rad/s]');

```

P3 Điều khiển dự báo phản hồi đầu ra hệ con lắc ngược

Toàn bộ phần mã nguồn chương trình cài đặt trên MatLab để điều khiển phản hồi đầu ra hệ con lắc ngược (3.1) theo nguyên lý tách (hình 2.5), trên cơ sở ghép chung bộ điều khiển dự báo phản hồi trạng thái, xây dựng theo Thuật toán 2.4 do luận án đề xuất, và bộ quan sát trạng thái EKF loại 3 theo Thuật toán 1.9, có dạng như sau:

```

%MPC PHAN HOI DAU RA CON LAC NGUOC (2 DAU RA)
clear all;clc;
T=0.001;           %thoi gian trich mau
N_sim=50;%thoi gian mo phong

q=1e-5;
r=1e-4;
Q=[q 0 0 0;0 q 0 0;0 0 q 0;0 0 0 q];   %ma tran tuong
quan cua w
R=[r];           %ma tran tuong quan cua v

Pk=0.1;           %dieu kien dau cua sai lech uoc luong

%CAC THONG SO BAN DAU
%cac thong so cua doi tuong
L=0.27;
mh=0.1;

```

```

mc=1.2;
g=9.8;

N=10;
%udk=1;

xes=[1.5;1;0.5;1];
xtrue=[1.5;1;0.5;1];
A_lt=[0,0,0,0;-((g*cos(xes(3)))-
L*xes(4)^2)*mh*sin(xes(3)))/((mc+mh*sin(xes(3))^2)*xes(1
)),0,0,0;0,0,0,0;0,0,((g-
L*xes(4)^2*cos(xes(3)))*mh*sin(xes(3))+g*mc*sin(xes(3)))
/((L*mc+L*mh*sin(xes(3))^2)*xes(3)),0];
B_lt=[0;1/(mc+mh*sin(xes(3))*sin(xes(3)));0;-
cos(xes(3))/(L*mc+L*mh*sin(xes(3))*sin(xes(3)))];
C_lt=[1,0,0,0;0,0,1,0];
D_lt=[0;0];
[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');

%%%%%%%%

w=zeros(2*N,1);
w1=1.5*ones(N,1);
wk1=1.5*ones(N_sim+1,1);
w2=0.5*ones(N,1);
wk2=0.5*ones(N_sim+1,1);
w(1)=1.5;
w(2)=0.5;
for i=1:N-1
    w(2*i+1)=w1(i);
    w(2*i+2)=w2(i);

```

```

end

y1=zeros(N_sim,1);
y2=zeros(N_sim,1);
u=zeros(N_sim+1,1);
noisex=wgn(N_sim+1,4,q,'linear');%0.05*randn(N_sim+1,1);
noisey=wgn(N_sim+1,1,r,'linear');
delta=10*eye(2*N);
Lk=10*eye(2*N);
%udk(1,1)=u(1,1);
%usbeta(1,1)=u(2,1);

for k=1:N_sim

h(1:2,:)=C_d;
E(1:2,:)=C_d*A_d;%2 hang dau tien cua ma tran E

for i=1:N-1
h(2*i+1:2*i+2,:)=h(2*i-1:2*i,:)*A_d;
E(2*i+1:2*i+2,:)= E(2*i-1:2*i,:)*A_d;%Ma tran E
end
v(1:2*N,:)=h*B_d];

F=zeros(2*N,2*N); %Khai bao kich thuoc cua F
F(1:2,1)=v(1:2,1); % 2 hang dau tien cot dau tien cua
F

for i=1:N-1
F(2*i+1:2*i+2,1)=v(2*i+1:2*i+2,1);% cot dau tien cua
ma tran F
%delta(2*i+1,i)=2*N+2-i;

```



```

end
for i=1:2*N
    delta(i,i)=2*N-i+1;
end
F(1:2*N,1)=F1;

for i=1:N-1
    F(1:2*N,i+1)=[zeros(2*i,1);F1(1:2*N-2*i,1)];
end
%w(2*k+1,1)=w1(k,1);
%w(2*k+2,1)=w2(k,1);
z=w-E*xtrue;

p_sao=inv(F'*delta*F+Lk)*F'*delta*z;
udk=p_sao(1,1);
udknoise=udk+noisex(k+1);
%Kich hoat he thong
xtrue=A_d*xtrue+B_d*udk;
yknoise=C_lt*xtrue+noisey(k);

u(i+1,1)=udknoise;
%PHAN HOI DAU RA (QUAN SAT TRANG THAI BANG EKF LOAI 3)
    F_ekf=A_d;
    xes=A_d*xes+B_d*udknoise;
    G_ekf=C_d;
    Pk=F_ekf*Pk*F_ekf'+Q;
    K_ekf=Pk*G_ekf'*inv(G_ekf*Pk*G_ekf'+R);
    Pk=(eye(size(K_ekf*G_ekf))-K_ekf*G_ekf)*Pk;
    xes=xes+K_ekf*(yknoise-G_ekf*xes);

    ykthuc=C_lt*xes+noisey(k);

```

```

y1(k+1)=ykthuc(1,1);
y2(k+1)=ykthuc(2,1);

A_lt=[0,0,0,0;-((g*cos(xes(3))-
L*xes(4)^2)*mh*sin(xes(3)))/((mc+mh*sin(xes(3))^2)*xes(1
)),0,0,0;0,0,0,0;0,0,((g-
L*xes(4)^2*cos(xes(3)))*mh*sin(xes(3))+g*mc*sin(xes(3)))
/((L*mc+L*mh*sin(xes(3))^2)*xes(3)),0];
B_lt=[0;1/(mc+mh*sin(xes(3))*sin(xes(3)));0;-
cos(xes(3))/(L*mc+L*mh*sin(xes(3))*sin(xes(3)))];
C_lt=[1,0,0,0;0,0,1,0];
D_lt=[0;0];
[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');

end

t=0:N_sim;
t1=0:N_sim+N-2;

figure(1)
plot(t,y1,'r--','LineWidth',3.5);grid on;
hold on
plot(t,wk1,'b','LineWidth',3.5);grid on;
legend('MPC phan hoi trang thai','vi tri xe dat');
xlabel('k');
ylabel('vi tri xe(m)');

figure(2)
plot(t,y2,'r--','LineWidth',3.5);grid on;
hold on
plot(t,wk2,'b','LineWidth',3.5);grid on;

```

```

legend('MPC phan hoi trang thai','gia tri vi tri goc
dat');
xlabel('k');
ylabel('vi tri goc (rad)');

figure(3)

plot (t,u,'r-o','LineWidth',1.5);grid on;
xlabel('k');
ylabel('control signal');

```

P4 Điều khiển dự báo phản hồi đầu ra cho hệ con lắc ngược quay

P4a Quan sát trạng thái EKF loại 3 hệ con lắc ngược quay

```

%Kalman mo rong loai 3 (EKF3): CON LAC NGUOC QUAY
clear all;clc;
T=0.01;           %thoi gian trich mau
N_sim=100;%thoi gian mo phong
%q=0;
q=0;
r=1e-5;
Q=[q 0 0 0;0 q 0 0;0 0 q 0;0 0 0 q];   %ma tran tuong
quan cua w
R=[r];           %ma tran tuong quan cua v

Pk=0.1;         %dieu kien dau cua sai lech uoc luong
u=20*ones(1,N_sim+2); %bien dau vao

t=[0:N_sim];   %ma tran thoi gian

```

```

%CAC THONG SO BAN DAU
%cac thong so cua doi tuong
m1=0.0319; l1=0.1572; Ku=0.0636; Kt=0.0706; Kb=0.0707;
Ra=0.9; J0=0.008591; C0=0.006408; J1=0.000217;
C1=0.000158;g=9.8;L0=0.0312;

%N=10;
x=[pi/2;2;2*pi;2];           %dieu kien dau cua trang thai
thuc
xes=[pi/2;2;2*pi;2];         %dieu kien dau cua trang thai
uoc luong
xtrue=[pi/2;2;2*pi;2];       %bien trang thai thuc
y_thuc=0;

for i=1:N_sim+1
k1=J0+m1*L0*L0+m1*l1*l1*sin(xes(3))*sin(xes(3));k2=m1*L0
*l1*cos(xes(3));
k3=C0+(Kt*Kb/Ra)+0.5*(m1*l1*l1*xes(4)*sin(2*xes(3)));
k4=m1*L0*l1*xes(4)*sin(xes(3))+0.5*m1*l1*l1*xes(2)*sin(2
*xes(3)); k5=Kt*Ku/Ra; k6=J1+m1*l1*l1;
k7=0.5*m1*l1*l1*xes(2)*sin(2*xes(3));
k8=C1;k9=m1*g*l1*sin(xes(3));
K1=(k2*k7-k6*k3)/(k1*k6-k2*k2);K2=(k2*k8+k6*k4)/(k1*k6-
k2*k2); K3=(k2*k9)/(k1*k6-k2*k2); K4=(k6*k5)/(k1*k6-
k2*k2); K5=(k2*(k2*k7-k6*k3)+k7*(k1*k6-
k2*k2))/(k6*(k1*k6-k2*k2));

```

```

K6=(k2*(k2*k8+k6*k4)+k8*(k1*k6-k2*k2))/(k6*(k1*k6-
k2*k2)); K7=((k2*k2*k9)+k9*(k1*k6-k2*k2))/(k6*(k1*k6-
k2*k2)); K8=(k2*k6*k5)/(k6*(k1*k6-k2*k2));

```

```

A_lt=[0,0,0,0;0,K1+K3/xes(2),0,-
K2;0,0,0,0;0,K5+K7/xes(2),0,-K6];
B_lt=[0;K4;0;K8];
C_lt=[0,0,1,0];
D_lt=0;
[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');
%A_d=1+T*A_lt; B_d=T*B_lt; C_d=C_lt; D_d=D_lt;

```

```

x1(i)=x(1,1); %them vao ma tran x1
x2(i)=x(2,1); %them vao ma tran x2
x3(i)=x(3,1);
x4(i)=x(4,1);

```

```

%kich hoat he thong
x=A_d*x+B_d*u(i)+wgn(4,1,q,'linear');
y=C_d*x+wgn(1,1,r,'linear');

```

```

%trang thai thuc
xt1(i)=xtrue(1,1); %them vao ma tran xt1
xt2(i)=xtrue(2,1); %them vao ma tran xt2
xt3(i)=xtrue(3,1); %them vao ma tran xt3
xt4(i)=xtrue(4,1); %them vao ma tran xt4

```

```

xtrue=A_d*xtrue+B_d*u(i);

```

```

%cac buoc tinh toan

```

```

xes1(i)=xes(1,1);    %them vao ma tran xes1
xes2(i)=xes(2,1);    %them vao ma tran xes2
xes3(i)=xes(3,1);    %them vao ma tran xes3
xes4(i)=xes(4,1);    %them vao ma tran xes4

Fk=A_d;
xes=A_d*xes+B_d*u(i);
Gk=C_d;
Pk=Fk*Pk*Fk'+Q;
K_ekf=Pk*Gk'*inv(Gk*Pk*Gk'+R);
Pk=(eye(size(K_ekf*Gk))-K_ekf*Gk)*Pk;
xes=xes+K_ekf*(y-Gk*xes);
y_thuc=C_lt*xes+wgn(1,1,r,'linear');

end
%ve do thi

figure(1);
subplot(2,1,1)

plot(t,xt1,'b','LineWidth',3.5);grid on;
hold on
plot(t,xes1,'r--','LineWidth',3.5);grid on;
title('uoc luong goc giua canh tay voi truc x')
legend('alpha true','alpha estimate','makersize',18);
xlabel('k');ylabel('alpha[rad]');

%figure(2);
subplot(2,1,2)

```

```

plot(t,xt2,'b','LineWidth',3.5);grid on;
hold on
plot(t,xes2,'r--','LineWidth',3.5);grid on;
title('uoc luong van toc goc theo truc x')
legend('alpha-dot true','alpha-dot estimate');
xlabel('k');ylabel('alpha-dot[rad/s]');

figure(2);
subplot(2,1,1)

plot(t,xt3,'b','LineWidth',3.5);grid on;
hold on
plot(t,xes3,'r--','LineWidth',3.5);grid on;
title('uoc luong goc lech giua canh tay voi truc z')
legend('Beta true','Beta estimate','makersize',18);
xlabel('k');ylabel('Beta[rad]');

subplot(2,1,2)
%figure(3);
plot(t,xt4,'b','LineWidth',3.5);grid on;
hold on
plot(t,xes4,'r--','LineWidth',3.5);grid on;
title('uoc luong van toc goc theo truc z')
legend('Beta-dot true','Beta-dot
estimate','makersize',18);
xlabel('k');ylabel('beta-dot[rad/s]');

```

P4b Điều khiển dự báo phản hồi trạng thái hệ con lắc ngược quay

```

%MPC PHAN HOI TRANG THAI CON LAC NGUOC QUAY
clear all;clc;

```

```

T=0.001;           %thoi gian trich mau
N_sim=50;%thoi gian mo phong

q=1e-5;
r=1e-5;
%cac thong so cua doi tuong
t=[0:N_sim];      %ma tran thoi gian

%CAC THONG SO BAN DAU
%cac thong so cua doi tuong
m1=0.0319; l1=0.1572; Ku=0.0636; Kt=0.0706; Kb=0.0707;
Ra=0.9; J0=0.008591; C0=0.006408; J1=0.000217;
C1=0.000158;g=9.8;L0=0.0312;

N=10;
udk=1;
xtrue=[pi/2;2;pi/10;2];
k1=J0+m1*L0*L0+m1*l1*l1*sin(xtrue(3))*sin(xtrue(3));k2=m
1*L0*l1*cos(xtrue(3));
k3=C0+(Kt*Kb/Ra)+0.5*(m1*l1*l1*xtrue(4)*sin(2*xtrue(3)))
;
k4=m1*L0*l1*xtrue(4)*sin(xtrue(3))+0.5*m1*l1*l1*xtrue(2)
*sin(2*xtrue(3)); k5=Kt*Ku/Ra; k6=J1+m1*l1*l1;
k7=0.5*m1*l1*l1*xtrue(2)*sin(2*xtrue(3));
k8=C1;k9=m1*g*l1*sin(xtrue(3));
K1=(k2*k7-k6*k3)/(k1*k6-k2*k2);K2=(k2*k8+k6*k4)/(k1*k6-
k2*k2); K3=(k2*k9)/(k1*k6-k2*k2); K4=(k6*k5)/(k1*k6-
k2*k2); K5=(k2*(k2*k7-k6*k3)+k7*(k1*k6-
k2*k2))/(k6*(k1*k6-k2*k2));

```



```

K6=(k2*(k2*k8+k6*k4)+k8*(k1*k6-k2*k2))/(k6*(k1*k6-
k2*k2)); K7=((k2*k2*k9)+k9*(k1*k6-k2*k2))/(k6*(k1*k6-
k2*k2)); K8=(k2*k6*k5)/(k6*(k1*k6-k2*k2));

```

```

A_lt=[0,0,0,0;0,K1+K3/xtrue(2),0,-
K2;0,0,0,0;0,K5+K7/xtrue(2),0,-K6];
B_lt=[0;K4;0;K8];
C_lt=[0,0,1,0];
D_lt=0;
[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');

```

```

%%%%%%%%

```

```

wk=zeros(N_sim+N-1,1);
for i=1:N_sim+N-1
    wk(i)=pi/10;
end
w=zeros(N,1);
y=zeros(N_sim,1);
u=zeros(N_sim+1,1);
noisex=wgn(N_sim+1,4,q,'linear');%0.05*randn(N_sim+1,1);
noisey=wgn(N_sim+1,1,r,'linear');
delta=10*eye(N);
Lk=10*eye(N);
udk(1,1)=u(1,1);
%usbeta(1,1)=u(2,1);

for k=1:N_sim

h(1,:)=C_d;
E(1,:)=C_d*A_d;%hang dau tien cua ma tran E

```

```

for i=2:N
h(i,:)=h(i-1,:)*A_d;
E(i,:)= E(i-1,:)*A_d;%Ma tran E
end
v(1:N,:)=[h*B_d];

F=zeros(N,N); %Khai bao kich thuoc cua F
%F1(1,1:5)=v; % hang thu nhat 5 cot dau tien cua F

for i=1:N
F1(i,1)=v(i,1);% cot dau tien cua ma tran F
delta(i,i)=N-i+1;
end

F(1:N,1)=F1;

for i=1:N-1
    F(1:N,i+1)=[zeros(i,1);F1(1:N-i,1)];
end
w(1:N,1)=wk(k:k+N-1);
z=w-E*xtrue;

p_sao=inv(F'*delta*F+Lk)*F'*delta*z;
udk=p_sao(1,1);
udknoise=udk+noisex(k+1);
%phan hoi trang thai
xtrue=A_d*xtrue+B_d*udk;
yknoise=C_lt*xtrue+noisy(k);

```

```

y(k+1)=yknoise;
u(i+1,1)=udknoise;

k1=J0+m1*L0*L0+m1*l1*l1*sin(xtrue(3))*sin(xtrue(3));k2=m
1*L0*l1*cos(xtrue(3));
k3=C0+(Kt*Kb/Ra)+0.5*(m1*l1*l1*xtrue(4)*sin(2*xtrue(3)))
;
k4=m1*L0*l1*xtrue(4)*sin(xtrue(3))+0.5*m1*l1*l1*xtrue(2)
*sin(2*xtrue(3)); k5=Kt*Ku/Ra; k6=J1+m1*l1*l1;
k7=0.5*m1*l1*l1*xtrue(2)*sin(2*xtrue(3));
k8=C1;k9=m1*g*l1*sin(xtrue(3));
K1=(k2*k7-k6*k3)/(k1*k6-k2*k2);K2=(k2*k8+k6*k4)/(k1*k6-
k2*k2); K3=(k2*k9)/(k1*k6-k2*k2); K4=(k6*k5)/(k1*k6-
k2*k2); K5=(k2*(k2*k7-k6*k3)+k7*(k1*k6-
k2*k2))/(k6*(k1*k6-k2*k2));
K6=(k2*(k2*k8+k6*k4)+k8*(k1*k6-k2*k2))/(k6*(k1*k6-
k2*k2)); K7=((k2*k2*k9)+k9*(k1*k6-k2*k2))/(k6*(k1*k6-
k2*k2)); K8=(k2*k6*k5)/(k6*(k1*k6-k2*k2));

A_lt=[0,0,0,0;0,K1+K3/xtrue(2),0,-
K2;0,0,0,0;0,K5+K7/xtrue(2),0,-K6];
B_lt=[0;K4;0;K8];
C_lt=[0,0,1,0];
D_lt=0;
[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');
end

t=0:N_sim;
t1=0:N_sim+N-2;
figure(1)
plot(t,y,'b-.o','LineWidth',3.5);grid on;

```

```

hold on
plot(t1,wk,'b','LineWidth',3.5);grid on;
legend('MPC phan hoi trang thai','gia tri vi tri goc
Beta dat');
xlabel('k');
ylabel('vi tri goc Beta (rad)');

```

```
figure(2)
```

```

plot (t,u,'r-o','LineWidth',1.5);grid on;
xlabel('k');
ylabel('control signal');

```

P4c Điều khiển dự báo phản hồi đầu ra hệ con lắc ngược quay

```

%MPC PHAN HOI DAU RA CON LAC NGUOC QUAY (SU DUNG EKF
LOAI 3)
clear all;clc;
T=0.001;           %thoi gian trich mau
N_sim=50;%thoi gian mo phong

q=1e-5;
r=1e-5;
Q=[q 0 0 0;0 q 0 0;0 0 q 0;0 0 0 q];   %ma tran tuong
quan cua w
R=[r];           %ma tran tuong quan cua v

Pk=0.1;           %dieu kien dau cua sai lech uoc luong

%cac thong so cua doi tuong

```

```

m1=0.0319; l1=0.1572; Ku=0.0636; Kt=0.0706; Kb=0.0707;
Ra=0.9; J0=0.008591; C0=0.006408; J1=0.000217;
C1=0.000158;g=9.8;L0=0.0312;

```

```

N=10;

```

```

xes=[pi/2;2;pi/10;2];      %dieu kien dau cua trang thai
uoc luong

```

```

xtrue=[pi/2;2;pi/10;2];    %bien trang thai thuc

```

```

k1=J0+m1*L0*L0+m1*l1*l1*sin(xes(3))*sin(xes(3));k2=m1*L0
*l1*cos(xes(3));

```

```

k3=C0+(Kt*Kb/Ra)+0.5*(m1*l1*l1*xes(4)*sin(2*xes(3)));

```

```

k4=m1*L0*l1*xes(4)*sin(xes(3))+0.5*m1*l1*l1*xes(2)*sin(2
*xes(3)); k5=Kt*Ku/Ra; k6=J1+m1*l1*l1;

```

```

k7=0.5*m1*l1*l1*xes(2)*sin(2*xes(3));

```

```

k8=C1;k9=m1*g*l1*sin(xes(3));

```

```

K1=(k2*k7-k6*k3)/(k1*k6-k2*k2);K2=(k2*k8+k6*k4)/(k1*k6-
k2*k2); K3=(k2*k9)/(k1*k6-k2*k2); K4=(k6*k5)/(k1*k6-
k2*k2); K5=(k2*(k2*k7-k6*k3)+k7*(k1*k6-
k2*k2))/(k6*(k1*k6-k2*k2));

```

```

K6=(k2*(k2*k8+k6*k4)+k8*(k1*k6-k2*k2))/(k6*(k1*k6-
k2*k2)); K7=((k2*k2*k9)+k9*(k1*k6-k2*k2))/(k6*(k1*k6-
k2*k2)); K8=(k2*k6*k5)/(k6*(k1*k6-k2*k2));

```

```

A_lt=[0,0,0,0;0,K1+K3/xes(2),0,-
K2;0,0,0,0;0,K5+K7/xes(2),0,-K6];

```

```

B_lt=[0;K4;0;K8];

```

```

C_lt=[0,0,1,0];

```

```

D_lt=0;

```

```

[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');

%%%%%%%%

wk=zeros(N_sim+N-1,1);
for i=1:N_sim+N-1
    wk(i)=pi/10;
end
%for i=1:10
%    wk(i)=80;
%end
%for i=11:20
%    wk(i)=150;
%end
%for i=21:N_sim+N-1
%    wk(i)=100;
%end
w=zeros(N,1);
y=zeros(N_sim,1);
u=zeros(N_sim+1,1);
noisex=wgn(N_sim+1,4,q,'linear');%0.05*randn(N_sim+1,1);
noisey=wgn(N_sim+1,1,r,'linear');
delta=10*eye(N);
Lk=10*eye(N);
%udk(1,1)=u(1,1);

for k=1:N_sim

```

```

h(1,:)=C_d;
E(1,:)=C_d*A_d;%hang dau tien cua ma tran E

for i=2:N
h(i,:)=h(i-1,:)*A_d;
E(i,:)= E(i-1,:)*A_d;%Ma tran E
end
v(1:N,:)=[h*B_d];

F=zeros(N,N); %Khai bao kich thuoc cua F
%F1(1,1:5)=v; % hang thu nhat 5 cot dau tien cua F

for i=1:N
F1(i,1)=v(i,1);% cot dau tien cua ma tran F
delta(i,i)=N-i+1;
end

F(1:N,1)=F1;

for i=1:N-1
    F(1:N,i+1)=[zeros(i,1);F1(1:N-i,1)];
end
w(1:N,1)=wk(k:k+N-1);
z=w-E*xtrue;

p_sao=inv(F'*delta*F+Lk)*F'*delta*z;
udk=p_sao(1,1);
udknoise=udk+noisex(k+1);
%kich hoat he thong

```

```

xtrue=A_d*xtrue+B_d*udknoise;
yknoise=C_lt*xtrue+noisey(k);

%y(k+1)=yknoise;
u(i+1,1)=udknoise;
%PHAN HOI DAU RA (QUAN SAT TRANG THAI BANG EKF LOAI 3)
    F_ekf=A_d;
    xes=A_d*xes+B_d*udknoise;
    G_ekf=C_d;
    Pk=F_ekf*Pk*F_ekf'+Q;
    K_ekf=Pk*G_ekf'*inv(G_ekf*Pk*G_ekf'+R);
    Pk=(eye(size(K_ekf*G_ekf))-K_ekf*G_ekf)*Pk;
    xes=xes+K_ekf*(yknoise-G_ekf*xes);

    ykthuc=C_lt*xes+noisey(k);
    y(k+1)=ykthuc;

k1=J0+m1*L0*L0+m1*l1*l1*sin(xes(3))*sin(xes(3));k2=m1*L0
*l1*cos(xes(3));
k3=C0+(Kt*Kb/Ra)+0.5*(m1*l1*l1*xes(4)*sin(2*xes(3)));
k4=m1*L0*l1*xes(4)*sin(xes(3))+0.5*m1*l1*l1*xes(2)*sin(2
*xes(3)); k5=Kt*Ku/Ra; k6=J1+m1*l1*l1;
k7=0.5*m1*l1*l1*xes(2)*sin(2*xes(3));
k8=C1;k9=m1*g*l1*sin(xes(3));
K1=(k2*k7-k6*k3)/(k1*k6-k2*k2);K2=(k2*k8+k6*k4)/(k1*k6-
k2*k2); K3=(k2*k9)/(k1*k6-k2*k2); K4=(k6*k5)/(k1*k6-
k2*k2); K5=(k2*(k2*k7-k6*k3)+k7*(k1*k6-
k2*k2))/(k6*(k1*k6-k2*k2));
K6=(k2*(k2*k8+k6*k4)+k8*(k1*k6-k2*k2))/(k6*(k1*k6-
k2*k2)); K7=((k2*k2*k9)+k9*(k1*k6-k2*k2))/(k6*(k1*k6-
k2*k2)); K8=(k2*k6*k5)/(k6*(k1*k6-k2*k2));

```



```

A_lt=[0,0,0,0;0,K1+K3/xes(2),0,-
K2;0,0,0,0;0,K5+K7/xes(2),0,-K6];
B_lt=[0;K4;0;K8];
C_lt=[0,0,1,0];
D_lt=0;
[A_d,B_d,C_d,D_d]=c2dm(A_lt,B_lt,C_lt,D_lt,T,'zoh');

end

t=0:N_sim;
t1=0:N_sim+N-2;
figure(1)
plot(t,y,'r--','LineWidth',3.5);grid on;
hold on
plot(t1,wk,'b','LineWidth',2.5);grid on;
legend('MPC phan hoi dau ra','gia tri vi tri goc Beta
dat');
xlabel('k');
ylabel('vi tri goc Beta(rad)');

figure(2)

plot (t,u,'r-o','LineWidth',1.5);grid on;
xlabel('k');
ylabel('control signal');

```

P5 Mã nguồn chương trình của khối NonlinPreControl

```

function Out=NonLinPreControl(Input)
%#codegen

```

```

%Dau ra: udk (1)
%Dau vao: xe[n] (4)

xf=[Input(1);Input(2);Input(3);Input(4)];

%----- Cac ma tran hang A,B,C,D -----
A_d =[1.0000    0.0099    0.0000    0.0000
      0    0.9862    0.0088    0.0000
      0   -0.0000    1.0025    0.0100
      0   -0.0021    0.4902    1.0009];
B_d =[0.0000
      0.0058
      0.0000
      0.0009];
C_d =[0 0 1 0];
D_d=zeros(2,1);
%=====

%----- Cac ma tran A_d[x],B_d[x] -----
-----

%----- Cac tham so cua mo hinh -----
m1=0.0319; l1=0.1572; Ku=0.0636; Kt=0.0706; Kb=0.0707;
Ra=0.9;
J0=0.008591; C0=0.006408; J1=0.000217;
C1=0.000158;g=9.8;L0=0.0312;

```

```

% ===== OK1 =====
% Ku=19.6/253;
% Kt=0.0706*1.15;
% Kb=Kt;
% C0=0.006408*10;
% J0=0.008591*1.2;
% =====
% C1=0.000158*15;
% J1=0.000217*0.9;
  Ku=16.6/253;
  Kt=0.0706*0.8;
  Kb=Kt;
  C0=0.006408*20;
  J0=0.008591*1.2;
%=====
C1=0.000158*10;
J1=0.000217*0.9;

%----- Phan bo lai bien trang thai -----
x1=xf(1);x2=xf(2);x3=xf(3);x4=xf(4);q_dot=[x2;x4];

%----- Tinh cac ma tran Bq,Cq,Gq,T -----
%Mo hình toán: Bq*q_2dot + Cq*q_dot + Gq = T*u
b11=J0+m1*L0*L0+m1*l1*l1*sin(x3)*sin(x3);
b12=-m1*L0*l1*cos(x3);
b22=J1+m1*l1*l1;
Bq=[b11 b12;b12 b22];
%=====
c11=C0+Kt*Kb/Ra+0.5*m1*l1*l1*x4*sin(2*x3);

```

```

c12=m1*L0*l1*x4*sin(x3)+0.5*m1*l1*l1*x2*sin(2*x3);
c21=-0.5*m1*l1*l1*x2*sin(2*x3);
Cq=[c11 c12;c21 C1];
%=====
if abs(x3)>0.1
Gq=[0;-m1*g*l1*(sin(x3)/x3)];
else
Gq=[0;-m1*g*l1];
end
%=====
T=[Kt*Ku/Ra;0];

%-- Tinh toan mo hinh lien tuc ----
B1=inv(Bq);
a2=-B1*Gq;
a1=-B1*Cq;
b1=B1*T;
A=[0 1 0 0;0 a1(1,1) a2(1,1) a1(1,2);0 0 0 1;0 a1(2,1)
a2(2,1) a1(2,2)];
B=[0;b1(1,1);0;b1(2,1)];
%---- Chuyen sang mo hinh roi rac----
X=[A B;zeros(1,5)];
M=expm(X*0.01);
A_d=M(1:4,1:4);
B_d=M(1:4,5);

%----- Bo dieu khien du bao -----
%----- Cac khoi tao -----
----
N=10;
delta=1e7*eye(N);

```

```

delta=[1e0 0 0 0 0 0 0 0 0 0
       0 1e0 0 0 0 0 0 0 0 0
       0 0 1e0 0 0 0 0 0 0 0
       0 0 0 1e0 0 0 0 0 0 0
       0 0 0 0 1e0 0 0 0 0 0
       0 0 0 0 0 1e0 0 0 0 0
       0 0 0 0 0 0 1e0 0 0 0
       0 0 0 0 0 0 0 1e8 0 0
       0 0 0 0 0 0 0 0 1e0 0
       0 0 0 0 0 0 0 0 0 1e0];

Lk=0.1*eye(N);
Lk=[0.3 0 0 0 0 0 0 0 0 0
     0 0.1 0 0 0 0 0 0 0 0
     0 0 0.1 0 0 0 0 0 0 0
     0 0 0 0.1 0 0 0 0 0 0
     0 0 0 0 0.1 0 0 0 0 0
     0 0 0 0 0 0.1 0 0 0 0
     0 0 0 0 0 0 0.01 0 0 0
     0 0 0 0 0 0 0 1 0 0
     0 0 0 0 0 0 0 0 1 0
     0 0 0 0 0 0 0 0 0 1];

F=zeros(N,N);
h=zeros(N,4);
E=zeros(N,4);
%=====
%----- Tinh toan cac ma tran E, F -----
h(1,:)=C_d;
E(1,:)=C_d*A_d;      %hang dau tien cua ma tran E
for i=2:N
    h(i,:)=h(i-1,:)*A_d;
    E(i,:)= E(i-1,:)*A_d;    %Ma tran E

```

```

end
F(:,1)=h*B_d;      % Cot dau tien ma tran F
for i=1:N-1
    F(:,i+1)=[zeros(i,1);F(1:N-i,1)]; % Ma tran F
end
%=====

%----- Tinh toan tin hieu dk -----
Kf=-[1 zeros(1,9)]*inv(F'*delta*F+Lk)*F'*delta*E
Kf_hc=[Kf(1) Kf(2)*1.28 Kf(3)/10.47 Kf(4)/3.54];
udk=Kf_hc*xf;
%=====

Out=[udk;Kf_hc(1);Kf_hc(2);Kf_hc(3);Kf_hc(4)];

```

P6 Mã nguồn chương trình của khối Nonlinear Estimate và EsUpdate

```

function Out=NonLinEs(Input)
%#codegen

%Dau ra: xe[n|n] (4) + xe[n+1|n] (4) + P[n+1|n] (16)
=(24)
%Dau vao: u[n] (1) + y[n] (2) + xe[n|n-1] (4) + P[n|n-1]
(16) = (23)

u=Input(1);
y=[Input(2);Input(3)];
xe=Input(4:7);
P=[Input(8:11) Input(12:15) Input(16:19) Input(20:23)];
P=P';

```

```

%===== Bo Quan sat Kalman cho mo hinh con lac nguoc
quay =====
%  $x[n+1]=Ad[x].x[n]+Bd[x].u[n]+Gd.w[n]$ 
%  $y[n]=Cdx[n]+v[n]$ 
%  $w[n], v[n]$  la cac vecto nhieu he thong va nhieu do
luong
%=====
=====

%===== Cac ma tran mo hinh =====
Ad=zeros(4,4);
Bd=zeros(4,1);
Cd =[1 0 0 0;0 0 1 0];
Gd=eye(4,4);
% =====

%===== Cac Tham so he thong =====
%----- Cac tham so cua mo hinh -----
m1=0.0319; l1=0.1572; Ku=0.0636; Kt=0.0706; Kb=0.0707;
Ra=0.9;
J0=0.008591; C0=0.006408; J1=0.000217;
C1=0.000158;g=9.8;L0=0.0312;

%===== OK1 =====
Ku=19.6/253;
Kt=0.0706*1.15;
Kb=Kt;
C0=0.006408*4;
J0=0.008591*1.2;

```

```

%=====
C1=0.000158*5;
J1=0.000217*0.5;

%Tham so nhieu
Q=[1e-5 0 0 0;0 1 0 0;0 0 1e-5 0;0 0 0 0.5];
R=[0.00001 0;0 0.00001];
%=====

%===== Uoc luong bien trang thai =====
Mn = P*Cd'/(Cd*P*Cd'+R);
xe = xe + Mn*(y-Cd*xe);    % xe[n|n]
P = (eye(4)-Mn*Cd)*P;      % P[n|n]
%=====

%===== Du bao dau ra buoc tiep theo
=====
%----- Bien trang thai hien tai -----
x1=xe(1);
x2=xe(2);
x3=xe(3);
x4=xe(4);
q_dot=[x2;x4];
%-----

%----- Tinh cac ma tran Bq,Cq,Gq,T -----
%Mo hinh toan: Bq*q_2dot + Cq*q_dot + Gq = T*u
b11=J0+m1*L0*L0+m1*l1*l1*sin(x3)*sin(x3);
b12=-m1*L0*l1*cos(x3);

```



```

b22=J1+m1*l1*l1;
Bq=[b11 b12;b12 b22];
%=====
c11=C0+Kt*Kb/Ra+0.5*m1*l1*l1*x4*sin(2*x3);
c12=m1*L0*l1*x4*sin(x3)+0.5*m1*l1*l1*x2*sin(2*x3);
c21=-0.5*m1*l1*l1*x2*sin(2*x3);
Cq=[c11 c12;c21 C1];
%=====
if abs(x3)>0.1
Gq=[0;-m1*g*l1*(sin(x3)/x3)];
else
Gq=[0;-m1*g*l1];
end
%=====
T=[Kt*Ku/Ra;0];

%-- Tinh toan mo hinh lien tục ----
B1=inv(Bq);
a2=-B1*Gq;
a1=-B1*Cq;
b1=B1*T;
A=[0 1 0 0;0 a1(1,1) a2(1,1) a1(1,2);0 0 0 1;0 a1(2,1)
a2(2,1) a1(2,2)];
B=[0;b1(1,1);0;b1(2,1)];
%---- Chuyen sang mo hinh roi rac----
X=[A B;zeros(1,5)];
M=expm(X*0.01);
Ad=M(1:4,1:4);
Bd=M(1:4,5);
%=====
%----- Du bao dau ra buoc tiep theo -----

```

```

xe = Ad*xe + Bd*u;          % x[n+1|n]
P = Ad*P*Ad' + Gd*Q*Gd';    % P[n+1|n]

Out=[ [x1;x2;x3;x4];xe;(P(1,:))';(P(2,:))';(P(3,:))';(P(4
,:))'];

```

```

function Out=EsUpdate(Input)
%#codegen

%Dau ra: xe[n|n] (4)
%Dau vao: y[n] (2) + xe[n|n-1] (4) + P[n|n-1] (16) =
(22)
%Kalman mo rong loai 3 (EKF3): CON LAC NGUOC QUAY

y=[Input(1);Input(2)];
xe=Input(3:6);
P=[Input(7:10) Input(11:14) Input(15:18) Input(19:22)];
P=P';
Cd =[1 0 0 0;0 0 1 0];
R=[0.00001 0;0 0.00001];
Mn = P*Cd'/(Cd*P*Cd'+R);
xe = xe + Mn*(y-Cd*xe);    % xe[n|n]
Out=xe;

```