

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
KHOA CÔNG NGHỆ TỰ ĐỘNG HOÁ



ThS. Lê Hồng Thu
ThS. Lê Thị Thu Phương
ThS. Vũ Thị Oanh

BÀI GIẢNG
MATLAB VÀ ỨNG DỤNG TRONG KỸ THUẬT

Tài liệu lưu hành nội bộ

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
KHOA CÔNG NGHỆ TỰ ĐỘNG HOÁ

BÀI GIẢNG
MATLAB VÀ ỨNG DỤNG TRONG KỸ THUẬT

Thái Nguyên, tháng 6 năm 2022

Mục lục

Mục lục	4
Mở đầu	9
Chương I: KIẾN THỨC TỔNG QUÁT MATLAB	10
<i>Nội dung chính của chương</i>	<i>10</i>
<i>Mục tiêu cần đạt được của chương</i>	<i>10</i>
Bài 1: Kiến thức tổng quát MATLAB (Số tiết: 3 tiết)	10
1.1 Hoạt động của MATLAB	10
1.1.1 <i>Khái niệm về Matlab</i>	<i>10</i>
1.1.2. <i>Màn hình matlab</i>	<i>10</i>
1.1.3 <i>Tiện ích trợ giúp</i>	<i>12</i>
1.2 Phạm vi ứng dụng của MATLAB	12
1.3 Biến và các thao tác của biến	12
1.3.1 <i>Quy tắc đặt tên biến:</i>	<i>12</i>
1.3.2 <i>Xem nội dung của Workspace</i>	<i>13</i>
1.3.3 <i>Lưu nội dung của Workspace thành tập tin dữ liệu</i>	<i>13</i>
1.3.4. <i>Thao tác trên các biến có trong Workspace</i>	<i>14</i>
1.4 Ma trận và ma trận đặc biệt	14
1.4.1. <i>Khái niệm về ma trận</i>	<i>14</i>
1.4.2. <i>Tạo các ma trận đặc biệt</i>	<i>15</i>
1.5 Các phép toán trên mảng	16
1.6 Các phép toán ma trận	17
1.7 Giải hệ phương trình độc lập tuyến tính	18
Bài tập cuối chương	18
Chương II: CÁC HÀM TOÁN HỌC VÀ LẬP TRÌNH TRONG MATLAB	21
<i>Nội dung chính của chương</i>	<i>21</i>
<i>Mục tiêu cần đạt được của chương</i>	<i>21</i>
Bài 2: Các hàm toán học và lập trình trong Matlab (Số tiết: 3 tiết)	21
2.1 Đa thức và các hàm xử lý đa thức	21
2.1.1 <i>Nghiệm của đa thức</i>	<i>21</i>
2.1.2 <i>Cộng đa thức</i>	<i>23</i>

2.1.3 Nhân đa thức	23
2.1.4 Chia đa thức	23
2.1.5 Đạo hàm của đa thức	24
2.1.6 Nguyên hàm của đa thức	24
2.1.7 Phân thức hữu tỉ.....	24
2.2 Các phần tử cơ bản và các hàm toán học.....	25
2.2.1 Các phần tử cơ bản	25
2.2.2 Hàm toán học	26
2.3 Biểu thức quan hệ và hàm logic	27
2.3.1 Các phép toán quan hệ.....	27
2.3.2 Các phép toán logic.....	28
2.3.3 Các hàm quan hệ và logic	29
2.3.4. Symbolic Math Toolbox.....	30
Bài tập cuối chương.....	31
Chương III: ĐỒ HOẠ TRONG MATLAB.....	33
Nội dung chính của chương	33
Mục tiêu cần đạt được của chương.....	33
Bài 3: Đồ hoạ trong Matlab (Số tiết: 3 tiết).....	33
3.1 Đồ hoạ 2D.....	33
3.1.1 Sử dụng lệnh Plot	33
3.1.2 Kiểu đường, dấu và màu	34
3.1.3 Kiểu đồ thị.....	35
3.1.4 Đồ thị lưới, hộp chứa trục, nhãn và lời chú giải.....	35
3.1.5. Tập lệnh liên quan vẽ đồ thị	37
3.2. Đồ hoạ 3D.....	39
3.2.1. Đồ thị đường thẳng	39
3.2.2. Đồ thị bề mặt và lưới.....	40
3.2.3. Thao tác với đồ thị.....	43
Bài tập cuối chương.....	43
Chương IV : MÔ PHỎNG HỆ THỐNG VỚI MATLAB/SIMULIK	45
Nội dung chính của chương	45
Mục tiêu cần đạt được của chương.....	45
Bài 4: Công cụ simulink và simpower system (số tiết: 3 tiết)	45
4.1. Công cụ simulink và simpower system.....	45

4.1.1. Giới thiệu chung về Simulink	45
4.1.2 Cách tạo mô hình trong Simulink.....	46
4.1.3. Thiết lập các tham số mô phỏng.....	47
4.1.4. Thiết lập các tham số thay đổi theo vùng làm việc	49
4.1.5. Thực hiện mô phỏng.....	50
4.1.6. Giới thiệu về simpower system blocksets	50
Bài tập cuối bài :	64
Bài 5: Mô phỏng hệ thống với matlab/simulink (tiếp) (số tiết: 3 tiết).....	66
4.2. Mô phỏng hệ thống bằng matlab/simulink	66
4.2.1. Thư viện SIMULINK	66
Bài tập cuối bài.....	91
Bài tập cuối chương.....	94
Chương V: KIẾN THỨC CHUNG VỀ MÔ HÌNH HOÁ VÀ MÔ PHỎNG HỆ THỐNG.....	97
Nội dung chính của chương	97
Mục tiêu cần đạt được của chương.....	97
Bài 6: Kiến thức chung về mô hình hoá và mô phỏng hệ thống (số tiết: 3 tiết)	97
5.1. Mô hình hóa hệ thống	97
5.1.1. Khái niệm của mô hình hóa.....	97
5.1.2. Vai trò của phương pháp mô hình hóa hệ thống.....	99
5.2. Đối tượng, hệ thống, mô hình toán học	101
5.2.1. Đối tượng của hệ thống.....	101
5.2.2. Hệ thống.....	101
5.2.3. Mô hình, mô hình toán học.....	103
5.2.4. Cấu trúc hệ thống và mối liên hệ ngược	108
5.3. Nhận dạng hệ thống	110
5.4. Mô hình máy tính và phương pháp mô phỏng	112
5.4.1. Mô hình hoá hệ thống, mô hình máy tính và phương pháp mô phỏng.....	112
5.4.2. Bản chất của phương pháp mô phỏng.....	113
5.4.3. Các bước mô phỏng hệ thống.....	116
5.4.4 Một số môi trường mô phỏng thường gặp.....	118
Bài tập cuối chương.....	118
Chương VI: MÔ TẢ TOÁN HỌC CHO CÁC ĐỐI TƯỢNG.....	120
VÀ HỆ THỐNG	120
Nội dung chính của chương	120

<i>Mục tiêu cần đạt được của chương</i>	120
Bài 7: Mô tả toán học cho các đối tượng và hệ thống (số tiết: 3 tiết)	120
6.1. Hệ thống và các đặc tính của hệ thống	120
6.1.1. Hệ thống và hệ thống theo thời gian	120
6.1.2. Vấn đề tĩnh và động của hệ thống.....	123
6.1.3. Vấn đề trễ.....	124
6.1.4. Vấn đề tuyến tính hóa.....	124
6.2 Các định luật vật lý áp dụng trong hệ thống	125
6.2.1. Nhắc lại các phần tử cơ bản.....	125
6.2.2. Lý tưởng hóa các phần tử vật lý.....	127
6.2.3. Các phương trình cân bằng.....	128
6.2.4. Phân tích vật lý hệ thống điện.....	129
6.3 Mô hình cơ bản của các hệ động lực học	130
6.3.1. Mô hình hàm truyền của hệ thống liên tục.....	130
Bài tập cuối chương	133
Chương VII: ỨNG DỤNG CỦA MATLAB TRONG KỸ THUẬT	137
<i>Nội dung chính của chương</i>	137
<i>Mục tiêu cần đạt được của chương</i>	137
Bài 8: Ứng dụng Matlab trong điều khiển (số tiết: 3 tiết)	137
7.1 Ứng dụng Matlab trong điều khiển	137
7.1.1 Các lệnh kết nối hệ thống.....	137
7.1.2 Chuyển hệ thống từ hàm truyền thành dạng không gian trạng thái.....	140
7.1.3 Chuyển từ mô hình không gian trạng thái sang mô hình hàm truyền đạt.....	141
7.1.4 Đánh giá khả năng điều khiển được, quan sát được của hệ thống	141
7.1.5. Khảo sát hệ thống theo tiêu chuẩn tần số	142
Bài tập cuối bài:	143
Bài 9: Ứng dụng matlab trong mô phỏng các mạch điện – điện tử (số tiết: 3 tiết)	146
7.2 Ứng dụng Matlab trong mô phỏng các mạch điện – điện tử	146
7.2.1 Mạch tích phân RC.....	146
7.2.2. Mạch đóng mở sử dụng MOSFET.....	150
Bài 10: Ứng dụng Matlab trong mô phỏng các hệ điện cơ (số tiết: 3 tiết)	157
7.3 Ứng dụng Matlab trong mô phỏng các hệ điện cơ	157
Bài tập cuối chương	165
Các câu hỏi thường gặp	171

Bài thực hành số 1 (Số tiết: 3 tiết) Khai báo biến và các thao tác với biến, ma trận và phép toán với biến và ma trận.....	176
Bài thực hành số 2 (Số tiết: 3 tiết) Thao tác với các hàm toán học.....	184
Bài thực hành số 3 (Số tiết: 3 tiết) Đồ họa trong Matlab.....	190
Bài thực hành số 4 (Số tiết: 3 tiết) Mô phỏng với Matlab/Simulink.....	196
Bài thực hành số 5 (Số tiết: 3 tiết) Mô phỏng với Matlab/Simulink (Tiếp).....	199
Bài thực hành số 6 (Số tiết: 3 tiết) Ứng dụng Matlab trong điều khiển.....	204
Bài thực hành số 7 (Số tiết: 3 tiết) Ứng dụng Matlab trong điều khiển (Tiếp).....	211
Bài thực hành số 8 (Số tiết: 3 tiết) Mô phỏng các mạch điện, điện tử.....	218
Bài thực hành số 9 (Số tiết: 3 tiết) Mô phỏng hệ điện cơ.....	222
Bài thực hành số 10 (Số tiết: 3 tiết) Mô phỏng hệ điện cơ (Tiếp).....	226

Mở đầu

Bài giảng MATLAB và ứng dụng trong kỹ thuật được tập thể giảng viên thuộc bộ môn Kỹ thuật điện – điện tử biên soạn nhằm phục vụ cho việc giảng dạy của giảng viên và học tập của sinh viên Trường Đại học Công nghệ thông tin và Truyền thông - Đại học Thái Nguyên. Tập bài giảng này được biên soạn theo nội dung đề cương chi tiết học phần MATLAB và ứng dụng trong kỹ thuật ở trình độ đại học.

Nội dung tài liệu giúp sinh viên có khả năng sử dụng thành thạo phần mềm Matlab để áp dụng cho các bài toán kỹ thuật, áp dụng phương pháp mô phỏng để mô phỏng các đối tượng và hệ thống. Sinh viên có kiến thức phân tích và mô hình hóa các hệ động lực học trong hệ thống điều khiển. Sử dụng các ngôn ngữ mô hình hóa hệ thống, thiết kế hệ thống và thử nghiệm hệ thống. Có khả năng thực hành mô hình hóa và mô phỏng chế độ làm việc của các đối tượng và hệ thống trên máy tính nhằm kiểm chứng lại lý thuyết và có cơ sở để học tập, nghiên cứu các môn tiếp theo trong chương trình đào tạo. Nội dung tài liệu gồm 7 chương:

Chương 1. Kiến thức tổng quan Matlab

Chương 2. Các hàm toán học trong Matlab

Chương 3. Đồ họa trong Matlab

Chương 4. Mô phỏng hệ thống với Matlab/Simulink

Chương 5. Kiến thức chung về mô hình hóa và mô phỏng hệ thống

Chương 6. Mô tả toán học cho các đối tượng và hệ thống

Chương 7. Ứng dụng của Matlab trong kỹ thuật

Mặc dù tập thể tác giả đã dành nhiều thời gian và công sức để biên soạn, song khó tránh khỏi thiếu sót. Vậy, chúng tôi kính mong quý thầy cô và các bạn sinh viên đóng góp ý kiến để cuốn bài giảng được hoàn thiện hơn. Xin trân trọng cảm ơn.

Chương I: KIẾN THỨC TỔNG QUÁT MATLAB

Nội dung chính của chương

Chương I đưa ra các khái niệm về MATLAB, các cửa sổ ứng dụng trong Matlab và ý nghĩa của nó. Các biến và các thao tác trên biến như xem lại giá trị biến, xóa nội dung... Khái niệm cơ bản về ma trận, các phép toán cơ bản trong ma trận và các loại ma trận đặc biệt. Phương pháp giải các phương trình độc lập tuyến tính.

Mục tiêu cần đạt được của chương

Sinh viên cần nắm được các cửa sổ sử dụng trong MATLAB. và thành thạo việc tạo biến cũng như các thao tác khác trên biến. Áp dụng Matlab để tính toán và nhập ma trận cũng như các loại ma trận đặc biệt khác. Áp dụng bài toán giải phương trình độc lập tuyến tính trong Matlab.

Bài 1: Kiến thức tổng quát MATLAB (Số tiết: 3 tiết)

1.1 Hoạt động của MATLAB

1.1.1 Khái niệm về Matlab

MATLAB là một chương trình phần mềm lớn của lĩnh vực toán số. Tên chương trình là chữ viết tắt của từ Matrix Laboratory, thể hiện định hướng chính của chương trình là các phép tính vector và ma trận. Phần cốt lõi của chương trình bao gồm một số hàm tính toán, các chức năng nhập/ xuất cũng như các khả năng điều khiển chu trình và nhờ đó ta có thể dựng nên các Scripts.

MATLAB làm việc chủ yếu với ma trận. Ma trận cỡ $m \times n$ là bảng chữ nhật gồm m hàng và n cột. MATLAB có thể làm việc với nhiều kiểu dữ liệu khác nhau. Với chuỗi kí tự MATLAB cũng xem là một dãy các kí tự hay là dãy mã số của các kí tự. MATLAB dùng để giải quyết các bài toán về giải tích số, xử lý tín hiệu số, xử lý đồ họa, ... mà không phải lập trình cổ điển. Hiện nay, MATLAB có đến hàng ngàn lệnh và hàm tiện ích. Ngoài các hàm cài sẵn trong chính ngôn ngữ, MATLAB còn có các lệnh và hàm ứng dụng chuyên biệt trong các Toolbox, để mở rộng môi trường MATLAB nhằm giải quyết các bài toán thuộc các phạm trù riêng. Các Toolbox khá quan trọng và tiện ích cho người dùng như toán sơ cấp, xử lý tín hiệu số, xử lý ảnh, xử lý âm thanh, ma trận thưa, logic mờ,...

1.1.2. Màn hình matlab

Khi khởi động MATLAB ta thu được màn hình MATLAB với các cửa sổ sau đây:

Cửa sổ lệnh Command Window: Đây là cửa sổ chính của MATLAB. Tại đây ta thực hiện toàn bộ việc nhập lệnh và nhận kết quả tính toán. Dấu \gg là dấu đợi lệnh. Sau khi nhập lệnh và kết thúc bằng động tác nhấn phím ENTER, MATLAB sẽ xử lý lệnh và xuất kết quả liền ngay dưới dòng lệnh.

Ví dụ:

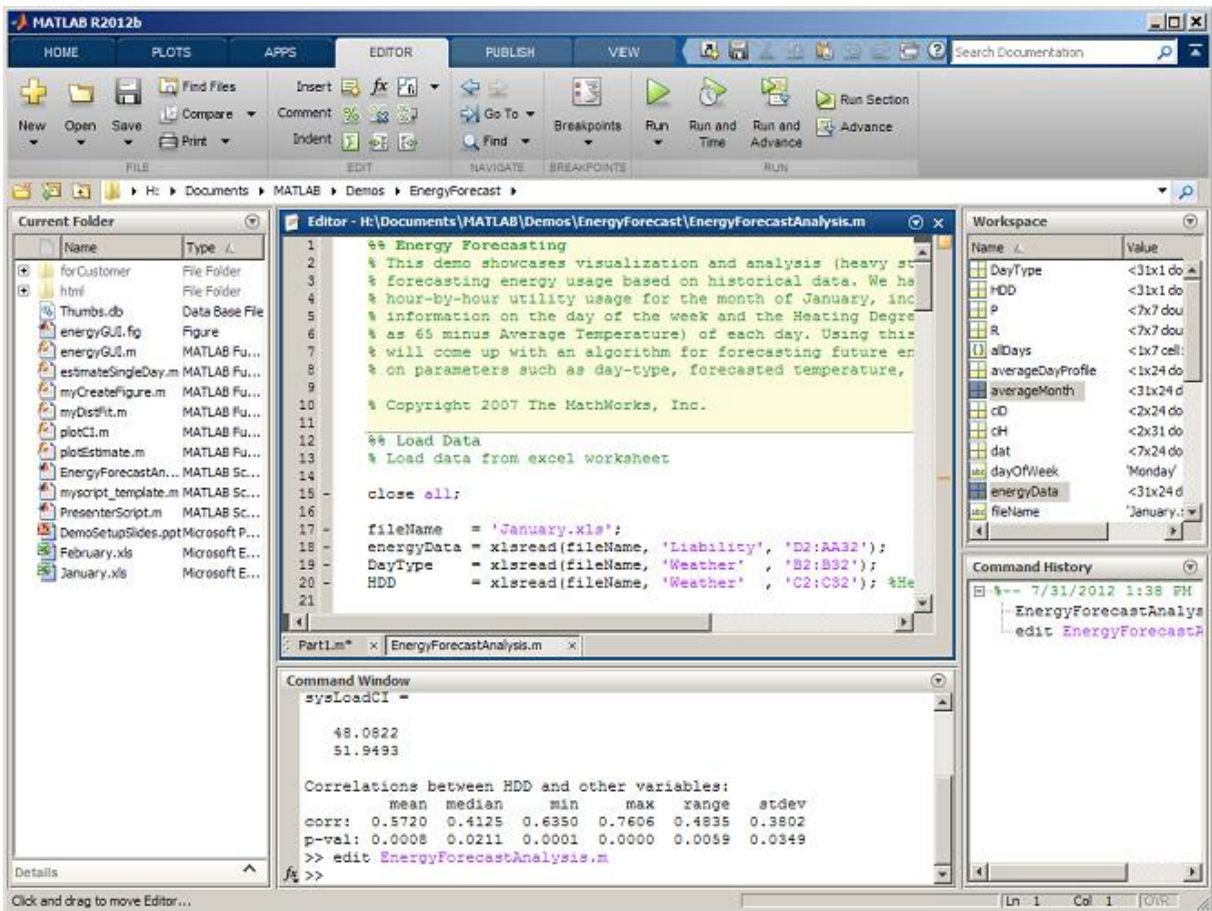
>> a=5*2+6 (ta nhập lệnh và ấn enter)

a=16 (kết quả thu được)

Cửa sổ **Command History**: Tất cả các lệnh đã sử dụng trong Command Window được lưu giữ và hiển thị tại đây. Có thể lặp lại lệnh cũ bằng cách nhấp đúp chuột vào lệnh đó. Cũng có thể cắt dán, sao chép, xóa cả nhóm lệnh hoặc từng lệnh riêng rẽ.

Cửa sổ **WorkspaceBrowser**: Khái niệm Workspace (không gian làm việc) là một vùng nhớ động trong bộ nhớ của chương trình, tự động hình thành khi MATLAB được khởi động và tự động xóa khi thoát MATLAB. Workspace lưu giữ các biến khi ta sử dụng MATLAB. Tất cả các biến tồn tại trong Workspace đều được hiển thị tại cửa sổ Workspace Browser với các thông tin về tên biến, giá trị, kích cỡ Byte và loại dữ liệu.

Cửa sổ thư mục hiện hành **Current Directory**: Được hiển thị khi nhấp chuột vào ô Current Directory. Nhờ cửa sổ này người sử dụng có thể nhanh chóng nhận biết các thư mục con và các tập tin (file) đang có trong thư mục hiện hành. Các thao tác mở file, lưu file, tìm M-file để thực thi...có mức ưu tiên cao nhất cho thư mục hiện hành. Mặc định khi khởi động MATLAB thì thư mục hiện hành là '..\Thư mục cài đặt MATLAB\work'. Tên thư mục hiện hành cũng được chỉ rõ trên thanh toolbar.



Hình 1.1: Màn hình Matlab 2012

1.1.3 Tiện ích trợ giúp

Tiện ích trợ giúp (Help) của MATLAB rất phong phú. Có thể gọi từ menu help trên thanh menu hoặc nhập lệnh tại Command window theo cú pháp:

help tênlệnh % xem trợ giúp tại command window.

doc tênlệnh % xem trợ giúp trong cửa sổ Help.

Ví dụ: Để tìm hiểu chức năng và cách dùng của lệnh input ta có thể nhập :

```
>> help input
```

Hoặc :

```
>> doc input
```

Ngoài ra, chúng ta cũng có thể xem các ví dụ minh họa có sẵn trong MATLAB bằng cách nhập lệnh demo.

1.2 Phạm vi ứng dụng của MATLAB

Các ứng dụng điển hình của Matlab là:

- Toán học và tính toán
- Phát triển thuật toán
- Tạo mô hình, mô phỏng và tạo giao thức
- Khảo sát, phân tích số liệu
- Đồ họa khoa học kỹ thuật
- Phát triển ứng dụng gồm cả xây dựng giao diện người dùng đồ họa GUI

MATLAB là một hệ thống tương tác mà phần tử dữ liệu cơ bản là một mảng không cần khai báo kích thước. Điều này cho phép bạn giải nhiều bài toán tính toán kỹ thuật đặc biệt là các bài toán liên quan đến ma trận và vector.

1.3 Biến và các thao tác của biến

Giống các ngôn ngữ lập trình khác, Matlab có những quy định riêng về tên biến.

1.3.1 Quy tắc đặt tên biến:

- + Tên biến phải bắt đầu bằng kí tự chữ. Kế tiếp có thể là chữ, số và dấu _
- + Không được dùng khoảng trống và các dấu (), ' , * , - , & , @ , ...
- + Có sự phân biệt chữ hoa và chữ thường.

Ví dụ:

Tên biến hợp lệ : a; b; A; A1; A2; chieu_cao; TT; TT_1; TT_2

Tên biến không hợp lệ : 1B; 2B; G(s); G'; G*, chieu cao; chieu-cao

Các tên biến sau đây là khác nhau: S; s ; the_tich; The_tich; THE_TICH

Lưu ý:

Không nên đặt tên biến trùng với các biến đặc biệt của MATLAB như: pi (số 3,14159...), i hay j (số ảo đơn vị), inf (số ∞), NaN hay nan (số bất định 0/0).

ans – là biến mặc định của MATLAB dùng để chứa dữ liệu hay kết quả tính toán nếu người dùng không đặt tên.

Sử dụng dấu = ta có thể khai báo một biến, đồng thời gán giá trị cho biến đó. Các biến được phân biệt với nhau bởi tên biến.

Một số lưu ý khi nhập lệnh:

- Bình thường MATLAB luôn hiển thị kết quả của câu lệnh trên màn hình. Nếu muốn MATLAB không hiển thị kết quả thì cuối câu lệnh ta đặt thêm dấu chấm phẩy (;).

- Nhiều câu lệnh có thể đặt chung trên một dòng nhưng bắt buộc phải phân cách nhau bởi dấu phẩy (,) hoặc chấm phẩy (;). Không cho phép phân cách các lệnh bằng khoảng trống. Nếu cuối lệnh nào có dấu phẩy thì MATLAB hiển thị kết quả, còn dấu chấm phẩy thì không hiển thị kết quả.

Nếu nhập lệnh: `>> S=a*b; V=S*h` thì MATLAB chỉ hiển thị giá trị của V, không hiển thị giá trị của S.

- Các phím mũi tên $\uparrow\downarrow\leftarrow\rightarrow$ trên bàn phím rất hữu ích khi nhập lệnh. Để gọi lại lệnh vừa gõ, bạn có thể nhấn phím mũi tên \uparrow , tiếp tục nhấn phím này, nó sẽ gọi tiếp lệnh trước đó. Phím mũi tên \downarrow có tác dụng ngược với \uparrow . Các phím mũi tên \leftarrow và \rightarrow có thể dùng để thay đổi vị trí con trỏ trong dòng lệnh tại dấu nhắc của MATLAB, giúp bạn dễ dàng chỉnh sửa nội dung dòng lệnh.

1.3.2 Xem nội dung của Workspace

Cách 1: Vào cửa sổ Workspace Browser xem danh sách liệt kê.

Cách 2: Dùng lệnh who hoặc whos

`>> who` % liệt kê tên các biến đang có trong Workspace ra màn hình Command

Your variables are: S V a ans b h

`>> whos` % liệt kê cả tên biến và các thông tin liên quan

Name	Size	Bytes	Class
S	1x1	8	double array
V	1x1	8	double array
a	1x1	8	double array
ans	1x1	8	double array
b	1x1	8	double array
h	1x1	8	double array

Grand total is 6 elements using 48 bytes

1.3.3 Lưu nội dung của Workspace thành tập tin dữ liệu

- Cách 1: Vào menu File > Save Workspace As > chọn thư mục khác (nếu cần)
> nhập tên tập tin > ấn nút Save. Tập tin dữ liệu có tên tổng quát là *.mat

- Cách 2: Nhập lệnh `>>save 'đường dẫn\tênfile.mat'`

Ví dụ:

`>>save 'C:\MATLAB 7\Work\mydata1.mat'`

Nếu bạn không nhập đường dẫn thì mặc định là lưu vào thư mục hiện hành.

Tải nội dung của một tập tin dữ liệu vào lại Workspace:

- Cách 1: Vào menu File > Import Data > MATLAB Data File (*.mat) > chọn tên tập tin > ấn nút Open.

- Cách 2: Nhập lệnh `>>load 'đường dẫn\tênfile.mat'`

- Cách 3: Vào cửa sổ Current Directory, nhấp đúp chuột vào tên tập tin cần mở.

1.3.4. Thao tác trên các biến có trong Workspace

- Xem lại giá trị của biến: Gõ tên biến tại dấu nhắc lệnh.

`>> tênbiến`

`>> tênbiến_1, tênbiến_2, ..., tênbiến_n` % giữa các tên biến có dấu phẩy

- Chỉnh sửa giá trị đã có của biến: Gõ lệnh gán mới.

Ví dụ: Thay đổi giá trị chiều cao h (đang là 4) thành 6 và tính lại thể tích :

`>> h=6`

`h= 6`

`>>V=S*h`

`V=60`

- Xoá sạch nội dung đang có trên màn hình Command window (nhưng không xoá biến) và đưa con trỏ về đầu màn hình

`>> clc`

- Xoá một số biến

`>> clear tênbiến_1 tênbiến_2 ... tênbiến_n` % chú ý là trường hợp này, giữa các tên biến có khoảng trống.

Ví dụ: Để xoá hai biến S và V ta gõ lệnh:

`>> clear S V`

- Xoá hết mọi biến trong Workspace

`>> clear`

Các thao tác xem nội dung, xoá, lưu, đổi tên, chỉnh sửa giá trị (edit value) của biến cũng có thể thực hiện tại cửa sổ Workspace Browser.

1.4 Ma trận và ma trận đặc biệt

1.4.1. Khái niệm về ma trận

Ma trận là mảng hai chiều có cấu trúc như sau:

$$\begin{matrix} & \begin{matrix} 1 & 2 & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ \vdots \\ m \end{matrix} & \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{matrix}$$

Với m là số hàng của ma trận và n là số cột của ma trận. Ta nói ma trận có kích thước $m \times n$.

Tạo một ma trận trong Matlab.

Để tạo một ma trận trong Matlab ta có thể sử dụng các cách sau:

- Tạo ma trận hàng

```
>>R=[a11 a12 a13]
```

Kết quả hiển thị là $a_{11} \ a_{12} \ a_{13}$

Ví dụ:

```
>> R=[18 23 5 ]
```

18 23 5

- Tạo ma trận cột: các phần tử của cột được cách nhau bởi dấu ;

```
>>C=[18; 23; 5]
```

18

23

5

- Tạo một ma trận bất kỳ: các phần tử cùng một hàng được phân cách nhau bằng dấu phẩy hoặc dấu cách. Các cột được phân biệt bởi dấu chấm phẩy

Ví dụ:

```
>> A=[2 3;4 5]
```

2 3

4 5

1.4.2. Tạo các ma trận đặc biệt

Matlab cho phép tạo nhanh các ma trận đặc biệt như sau:

Bảng 1.1: Các ma trận đặc biệt

STT	Dòng lệnh	Ý nghĩa	Ví dụ	Kết quả
1	zeros(a,b)	Tạo ma trận $a \times b$ các phần tử đều =0	Zeros(2,3)	0 0 0 0 0 0
2	ones(a,b)	Tạo ma trận $a \times b$ các phần tử đều =1	Ones(2,3)	1 1 1 1 1 1
3	eye(a,b)	Tạo ma trận $a \times b$ các phần tử đường chéo =1	Eye(3,3)	1 0 0 0 1 0 0 0 1
4	repmat(a,b)	Tạo ma trận $b \times b$ các phần tử có giá trị =a	Repmat(2,3)	2 2 2 2 2 2 2 2 2
5	rand(a,b)	Tạo ma trận $a \times b$, các phần tử ngẫu nhiên	Rand(2,3)	0.92 0.17 0.93 0.73 0.40 0.91

6	randn(a,b)	Tạo ma trận axb các phần tử ngẫu nhiên phân bố đều	Randn(2,3)	-0.43 0.12 -1.14 -1.66 0.28 1.19
7	linspace(a,b,n)	Tạo ma trận n phần tử phân bố đều từ a đến b	Linspace(1,1.1,4)	1 2

a. Phép toán giữa các ma trận

Ma trận cộng hoặc trừ phải có cùng kích thước. Kết quả nhận được là ma trận có cùng kích thước với các phần tử trong ma trận là tổng hoặc hiệu của 2 phần tử tương ứng. Ma trận A/B nếu A là ma trận vuông còn B là ma trận cột có cùng kích thước với A.

b. Kích thước của ma trận

- Kích thước ma trận: Size(z)
- Chiều dài ma trận: Length(z)
- Số phần tử ma trận: Numel(Z)
- Vector hàng chứa các phần tử lớn nhất theo cột: max(z)
- Vector hàng chứa phần tử nhỏ nhất theo từng cột: min(z)
- Vector hàng chứa tổng các phần tử theo từng cột: sum(z)
- Sắp xếp theo thứ tự tăng dần trong từng cột: sort(z)

1.5 Các phép toán trên mảng

Các phép toán về ma trận có thể thực hiện ngay dòng lệnh hoặc thực hiện trong M-file.

Phép toán trích xuất ma trận

Ví dụ: Dưới đây tạo ma trận 4 hàng 5 cột:

```
>> a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8]
```

MATLAB sẽ thực thi lệnh trên và trả về kết quả dưới đây:

a=

```
1  2  3  4  5
2  3  4  5  6
3  4  5  6  7
4  5  6  7  8
```

- Để trích xuất phần tử trong hàng thứ m và cột thứ n của ma trận a: a(m, n);

Ví dụ: Để trích xuất các phần tử của hàng 2 và cột 5 của ma trận a, ta nhập:

```
>> a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];
```

```
>> a(2,5)
```

MATLAB sẽ thực thi lệnh trên và trả về kết quả dưới đây:

ans = 6

- Để tham chiếu tất cả các phần tử trong cột thứ m, ta nhập: a(:,m)

Ví dụ: Để tạo cột vector v, từ phần tử của hàng thứ 4 của ma trận:

```
>> a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];
```

```
>> v = a(:,4)
```

- Chọn các phần tử trong cột thứ m thông qua cột thứ n: $a(:,m:n)$

Ví dụ: Để tạo ma trận nhỏ hơn lấy các phần tử từ cột thứ 2 và thứ 3:

```
>> a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];
```

```
>> a(:, 2:3)
```

- Xóa Cột Hoặc Hàng Trong Ma Trận Trong MATLAB: Có thể xóa toàn bộ cột của ma trận trong MATLAB bằng cách gán tập hợp dấu ngoặc vuông [] vào cột hoặc hàng đó. Về cơ bản [] biểu thị một mảng trống.

Ví dụ, để xóa hàng thứ tư của ma trận:

```
>> a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8];
```

```
>> a(4, :) = []
```

1.6 Các phép toán ma trận

- Tính định thức ma trận: $\det(A)$
- Chuyển vị ma trận: A' (là chuyển các hàng thành cột và cột thành hàng của ma trận ban đầu)

Ví dụ: Cho ma trận

```
>> A = [11 2 6; 4 17 26; 17 8 49]
```

```
>> det(A)
```

```
5825 //đây là kết quả của định thức ma trận A
```

```
>> A'
```

```
11    4    17
```

```
2     17    8
```

```
6     26    49
```

- Tạo ma trận tam giác dưới: $\text{tril}(A)$ tức là từ đường chéo trở lên các phần tử đều bằng 0

- Tạo ma trận tam giác trên: $\text{triu}(A)$ tức là từ đường chéo trở xuống các phần tử trong ma trận đều bằng 0

- Ma trận nghịch đảo của ma trận A là ma trận B sao cho $A*B=B*A=I$

Với I là ma trận đơn vị

```
>> inv(A)
```

```
0.1073    -0.0086    -0.0086
```

```
0.0422     0.0750    -0.0450
```

```
-0.0441    -0.0093     0.0307
```

- Phân tích ma trận thành thừa số LU: Ma trận A sẽ được phân tích thành 2 ma trận: L.U với L là ma trận tam giác dưới, U là ma trận tam giác trên..

>> [L,U]=lu(A)

1.7 Giải hệ phương trình độc lập tuyến tính

Ta có hệ phương trình:

$$\begin{cases} a_1x + b_1y + c_1z = d_1 \\ a_2x + b_2y + c_2z = d_2 \\ a_3x + b_3y + c_3z = d_3 \end{cases}$$

Trong Matlab hệ này được biểu diễn dưới dạng ma trận $Ax=b$ với

$$A = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \quad \text{và } b = [d_1 \ d_2 \ d_3]$$

Ví dụ: Giải hệ phương trình
$$\begin{cases} x + 2y - 3z = 5 \\ -3x - y + z = -8 \\ x - y + z = 0 \end{cases}$$

>> A=[1 2 -3;-3 -1 1;1 -1 1];

>> b=[5;-8;0];

>> X=A\b;

- X là một vector 3x1 chứa giá trị của x, y và z thỏa mãn phương trình. Phép chia sẽ làm việc với hệ phương trình có A là ma trận vuông hoặc ma trận chữ nhật. Phép chia đưa ra một nghiệm khi hệ có vô số nghiệm. Nếu hệ vô nghiệm, khi thực hiện MATLAB sẽ đưa ra một cảnh báo và vẫn trả về một vector 3x1.

Bài tập cuối chương

Bài 1: Cho $x = [1 \ 4 \ 8]$, $y = [2 \ 1 \ 5]$, và $A = [2 \ 7 \ 9 \ 7 ; 3 \ 1 \ 5 \ 6 ; 8 \ 1 \ 2 \ 5]$. Xét xem dòng lệnh nào hợp lệ, dự đoán kết quả, giải thích và thử lại bằng Matlab

1. $[x; y \ 0]$
2. $[x; y]$
3. $A(:, [1 \ 4])$
4. $A([2 \ 3], [3 \ 1])$
5. $A(:)$
6. $[A; A(\text{end}, :)]$
7. $A(1:3,:)$
8. $[A; A(1 : 2, :)]$

Bài 2: Tính bằng tay các biểu thức sau, rồi thử lại bằng Matlab:

- a. $2 / 2 * 3$
- b. $6 - 2 / 5 + 7 ^ 2 - 1$
- c. $10 / 2 \setminus 5 - 3 + 2 * 4$
- d. $3 ^ 2 / 4$ e. $3 ^ 2 ^ 2$
- f. $2 + \text{round}(6 / 9 + 3 * 2) / 2 - 3$
- g. $2 + \text{floor}(6 / 9 + 3 * 2) / 2 - 3$

h. $2 + \text{ceil}(6 / 9 + 3 * 2) / 2 - 3$

k. $\text{fix}(4/9) + \text{fix}(3*(5/6))$

Bài 3: Dự đoán kết quả xuất ra màn hình:

a. $2 ; 4$

b. $2 / 4$

c. $2 \setminus 4$

d. $2 , 4$

e. $2 : 4$

Bài 4: Cho $x = 2, y = 3$. sử dụng Matlab xác định kết quả của các giá trị biết:

a. $z = x$

b. $y = y + z$

c. $x = y + x - z$

d. $x + y - z$

Bài 5: Cho $A = [2 \ 7 \ 9 \ 7 ; 3 \ 1 \ 5 \ 6 ; 8 \ 1 \ 2 \ 5]$, dự đoán kết quả, giải thích; rồi thử lại bằng Matlab:

a. A'

b. $A(:, [1 \ 4])$

c. $A([2 \ 3], [3 \ 1])$

Bài 6: Cho $A = [2 \ 7 \ 9 \ 7 ; 3 \ 1 \ 5 \ 6 ; 8 \ 1 \ 2 \ 5]$, dự đoán kết quả, giải thích; rồi thử lại bằng Matlab:

a. $\text{reshape}(A, 2, 6)$

b. $A(:)$

c. $\text{flipud}(A)$

d. $\text{fliplr}(A)$

e. $[A; A(\text{end}, :)]$

Bài 7: Cho $A = [2 \ 7 \ 9 \ 7 ; 3 \ 1 \ 5 \ 6 ; 8 \ 1 \ 2 \ 5]$, dự đoán kết quả, giải thích; rồi thử lại bằng Matlab:

a. $A(1:3, :)$

b. $[A; A(1:2, :)]$

c. $\text{sum}(A)$

d. $\text{sum}(A')$

e. $\text{sum}(A, 2)$

f. $[[A ; \text{sum}(A)] [\text{sum}(A, 2) ; \text{sum}(A(:))]]$

Bài 8: Cho ma trận $A = [2 \ 4 \ 1 ; 6 \ 7 \ 2 ; 3 \ 5 \ 9]$, viết lệnh Matlab để

a. Gán cho vector x là dòng thứ nhất của A .

b. Gán cho ma trận y là hai dòng còn lại (cuối) của A .

Bài 9: Cho ma trận $A = [2 \ 4 \ 1 ; 6 \ 7 \ 2 ; 3 \ 5 \ 9]$, viết lệnh Matlab để

- a. Tính tổng theo dòng ma trận A.
- b. Tính tổng theo cột ma trận A.

Bài 10: Cho ma trận $A = [2 \ 4 \ 1 ; 6 \ 7 \ 2 ; 3 \ 5 \ 9]$, viết lệnh Matlab để

- a. Tìm giá trị lớn nhất và nhỏ nhất của ma trận.
- b. Tính tổng các phần tử của A

Câu hỏi trắc nghiệm

Câu 1: Để tạo ma trận với các phần tử trong ma trận đều là 0 sử dụng lệnh

- A. zeros(a,b)
- B. zero(a,b)
- C. Add(a,0)
- D. Adds(a,0)

Câu 2: Chọn các phần tử trong cột thứ m thông qua cột thứ n để tạo ra một ma trận con bằng cách

- A. a(m, n)
- B. a=(m:n)
- C. a(:, m:n)
- D. a(:,m)

Câu 3: Tính định thức trong ma trận z

- A. size(z)
- B. det(z)
- C. det('z')
- D. max(z)

Câu 4: xác định số phần tử của ma trận A

- A. length(A)
- B. Numel(A)
- C. Sum(A)
- D. Number(A)

Câu 5: Cách để tạo một ma trận m x n với các phần tử đường chéo =1

- A. Rand(m, n)
- B. repmat(m,n)
- C. Randn(m,n)
- D. eye(m,n)

Chương II: CÁC HÀM TOÁN HỌC VÀ LẬP TRÌNH TRONG MATLAB

Nội dung chính của chương

Chương 2 giúp sinh viên biết cách nhập một đa thức trong Matlab, cách xác định nghiệm của đa thức và phép tính trong đa thức. Cách xác định các hàm toán học, biểu thức quan hệ và các hàm quan hệ trong Matlab.

Mục tiêu cần đạt được của chương

Sinh viên cần sử dụng thành thạo Matlab trong việc nhập, tính toán đa thức, các hàm toán học và các hàm quan hệ.

Bài 2: Các hàm toán học và lập trình trong Matlab (Số tiết: 3 tiết)

2.1 Đa thức và các hàm xử lý đa thức

Ta có một đa thức: $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$

Trong đó: n là bậc của đa thức.

Trong Matlab, đa thức được biểu diễn bởi một vector hàng có các thành phần chính là các hệ số. Vector trên phải chứa tất cả các hệ số, kể cả hệ số bậc 0.

Ví dụ: $f(x)=8x+5$ $p=[8 \ 5]$

$f(x)=6x^2+150$ $h=[6 \ 0 \ 150]$

- Để tính giá trị của một đa thức tại một điểm x ta sử dụng hàm:

Polyval (p, x)

Trong đó p là vector biểu diễn đa thức và x là một số, biến, hoặc biểu thức.

Ví dụ:

```
>> p=[5 6 -7 3];
```

```
>> x=2;
```

```
>> y=polyval (p, x)
```

```
y = 53
```

2.1.1 Nghiệm của đa thức

Tìm nghiệm của đa thức là tìm giá trị để đa thức bằng không. Matlab có thể giải quyết những bài toán này và cung cấp những công cụ để tính toán đa thức. Trong Matlab một đa thức được biểu diễn bằng một vector hàng các hệ số với bậc giảm dần.

Matlab có thể tìm các nghiệm của một đa thức bằng lệnh:

```
>> r = roots(p)
```

Ví dụ:

```
>> p= [1 -2 -3]
```

```
>> r = roots(p)
```

```
r =    3.000
```

```
    -1.000
```

Trong Matlab cả đa thức và nghiệm của nó đều là vector nên Matlab ngầm quy ước rằng đa thức là vector hàng, còn các nghiệm là các vector cột. Nếu biết trước nghiệm của một đa thức thì sẽ dễ dàng biết được đa thức đó.

Trong Matlab nếu biết trước nghiệm của đa thức có thể tìm được các hệ số của đa thức đó bằng lệnh:

```
>> p=poly(r)
```

Trong đó: r là vector hàng hoặc cột chứa các nghiệm của đa thức
 P là vector hàng chứa các hệ số.

Ví dụ:

```
>> r =[-3; 2];
```

```
>> p=poly(r)
```

```
p= 1    1    -6
```

khi này đa thức đó là: $f(x)=x^2+x-6$

Bài tập: Cho đa thức:

$$x^4 - 12x^3 + 25x + 116$$

a. xác định nghiệm của đa thức

b. Từ nghiệm của đa thức tìm hệ số của đa thức.

Giải:

```
>> p=[1 -12 0 25 116]
```

```
P=
```

```
1    -12     0     25    116
```

```
>> r =roots(p)
```

```
r =
```

```
11.7374
```

```
2.7028
```

```
-1.2251 + 1.4672i
```

```
-1.2251 - 1.4672i
```

```
>> p = poly(r)
```

```
p =
```

```
1    -12    -1.77644e-14     25    116
```

```
>> p(abs(pp)<1e-12)=0 % Gán những phần tử quá nhỏ bằng 0
```

```
1    -12     0     25    116
```

Trong tính toán thường gặp những sai số nên đôi khi kết quả của lệnh poly cho ra các đa thức có các hệ số gần bằng không và các đa thức có phần ảo rất nhỏ như được chỉ ra ở trên, các giá trị bằng không có thể làm tròn bằng các công cụ về mảng. Tương tự như vậy ta có thể làm tròn một số phức để trở thành một số thực bằng hàm **real**.

2.1.2 Cộng đa thức

Matlab không cung cấp các hàm trực tiếp thực hiện phép cộng hai đa thức. Sử dụng phép cộng ma trận chỉ có tác dụng khi hai đa thức là hai vector có cùng kích thước.

Ví dụ: ta có hàm $f_1(x) = 3x^6 + 15x^5 - 10x^3 - 3x^2 + 15x - 40$

$f_2(x) = 3x^3 - 2x - 6$

```
>>p1=[3 15 0 -10 -3 15 -40]
```

```
>>p2=[0 0 0 3 0 -2 -6]
```

```
>>p=p1+p2
```

```
p =
```

```
3    15    0    -7    -3    13   -46
```

Khi hai đa thức có bậc khác nhau thì đa thức có bậc thấp hơn phải được thêm vào các hệ số 0 vào đầu vector để cho bậc của nó có cùng bậc với đa thức có bậc cao hơn.

Ví dụ:

```
>>c=[1 6 20 50 75 84 64]
```

```
>>d=[2 6 12 20]
```

```
>>e=c+[0 0 0 d]
```

```
e =
```

```
1    6    20    52    81    96    84
```

Kết quả là $e(x) = x^6 + 6x^5 + 20x^4 + 52x^3 + 81x^2 + 96x + 84$

2.1.3 Nhân đa thức

Để nhân hai đa thức ta thực hiện cú pháp:

```
>>c=conv(a, b)
```

Trong đó: a và b là các vector hệ số của đa thức
c là vector hệ số của tích.

Ví dụ:

```
>>a=[2 1 -3];
```

```
>>b=[1 1];
```

```
>>c=conv(a, b)
```

```
c =
```

```
2    3   -2   -3
```

2.1.4 Chia đa thức

Thực hiện chia đa thức theo cú pháp: $[q, r]=deconv(u,v)$

Trong đó: u là vector hệ số của các đa thức bị chia

v là vector hệ số của đa thức chia

q là vector hệ số của thương

r là vector hệ số của phần dư

Ví dụ:

```
>> u = [1 -9 -10];
>> v = [1 1];
>> [q, r]=deconv(u,v)
q = 1 -10
r = 0 0 0
```

Ta thấy rằng khi u chia cho v được đa thức là q và đa thức dư là r trong trường hợp này phần dư là đa thức 0 vì u chia hết cho q.

2.1.5 Đạo hàm của đa thức

Để đạo hàm của đa thức ta thực hiện lệnh: k=polyder(p)

Trong đó: p là vector hệ số của đa thức

K là vector hệ số của đạo hàm

Ví dụ:

```
>> p = [3 -2 4];
>> k =polyder(p)
k = 6 -2
```

2.1.6 Nguyên hàm của đa thức

Matlab tính nguyên hàm của đa thức bởi lệnh: g=polyint(h, k)

Trong đó: h là vector hệ số của đa thức

g là vector hệ số của nguyên hàm

k là hằng số tích phân, mặc định là 0

Ví dụ:

```
>> h =[6 0 0];
>> g=polyint(h)
g= 2 0 0 0
```

2.1.7 Phân thức hữu tỉ

Đôi khi ta gặp những bài toán liên quan đến tỉ số giữa 2 đa thức người ta gọi đó là phân thức hữu tỉ, ví dụ như các hàm truyền hay hàm xấp xỉ pade có dạng như sau:

$$\frac{n(x)}{d(x)} = \frac{N_1x^m + N_2x^{m-1} + \dots + N_{m-1}x^0}{D_1x^n + D_2x^{n-1} + \dots + D_{m-1}x^0}$$

Trong Matlab phân thức cũng được mô phỏng bằng hai đa thức riêng.

Ví dụ:

```
>> n =[1 -10 100]
n =
1 -10 100
>> d =[1 10 100 0]
d =
1 10 100 0
```



```
>> z=roots(n)
```

```
z =
```

```
5.000+8.6603i
```

```
5.000-8.6603i
```

```
>> p=roots(d)
```

```
p =
```

```
0
```

```
-5.000+8.6603i
```

```
-5.000-8.6603i
```

Đạo hàm của phân thức này theo biến x được tính dựa trên hàm **polyder**

```
>>[nd,dd]=polyder(n,d)
```

```
nd= -1 20 -100 -2000 -10000
```

```
dd=
```

```
Columns 1 through 6
```

```
1 20 300 2000 10000 0
```

```
Column 7
```

```
0
```

Ở đây nd và dd là tử và mẫu của đạo hàm. Một thao tác thông thường khác là tìm phần dư của phân thức.

2.2 Các phần tử cơ bản và các hàm toán học

2.2.1 Các phần tử cơ bản

a. Giới hạn của các giá trị tính toán trong Matlab

- Đối với phần lớn máy tính, khoảng giá trị cho phép từ 10⁻³²³ đến 10³⁰⁸.

- Nếu có giá trị tràn số mũ trên, nó được biểu diễn bởi inf (số vô hạn).

- Nếu tràn mũ dưới, nó được biểu diễn là 0

- Chia cho 0 là toán tử không hợp lệ, kết quả là inf. Matlab sẽ cảnh báo và sử dụng giá trị inf để tính tiếp.

b. Biến string

Chuỗi ký tự được đặt giữa 2 dấu nháy đơn

Chuỗi ký tự là một mảng nhiều ký tự. Ký tự được lưu dưới dạng mã ASCII. >>

```
name= 'Trường Đại học DL Công Nghệ Sài Gòn'
```

Có thể truy xuất đến từng phần tử chuỗi

```
>> fprintf('Trường tôi là %s\n', name(8:35));
```

Kết hợp các string tạo string mới

```
>> text1='Tôi học tại'; text=[text1 ' ' name];
```

Nhập string từ bàn phím:

```
>> str= input('Nhập vào một chuỗi','s');
```

Các lệnh với biến string:

Bảng 2.1: Các lệnh với biến string

Hàm	Ý nghĩa
Char	Tạo mảng ký tự
double	Đổi chuỗi sang mã ASCII
num2str	Đổi số sang chuỗi
str2num	Đổi chuỗi sang số
int2str	Đổi số nguyên sang chuỗi
str2mat	Đổi chuỗi sang ma trận
mat2str	Đổi ma trận sang chuỗi

2.2.2 Hàm toán học

a. Các hàm toán học cơ bản

Bảng 2.2: Bảng các hàm toán học cơ bản

Hàm	Ý nghĩa
Round	Làm tròn về số nguyên gần nhất
fix	Làm tròn về 0
Floor	Làm tròn nhỏ hơn
ceil	Làm tròn lớn hơn
log(x)	$\ln(x)$
log10(x)	Log thập phân
pow2(x)	Luỹ thừa cơ số 2
nextpow2(N)	Tìm p: $2^p=N$
Exp(x)	Hàm e^x
Sqrt(x)	Căn bậc hai của x
Abs(x)	Modun của số phức x
Imag(x)	Phần ảo của x
Real(x)	Phần thực của x
Sign(x)	Dấu của x

Ví dụ:

```
>> a=[-1.9 -0.2 3.4 5.6 7 2.4 +3.6i];
>> fix(a)
-1.0000 0 3.0000 5.0000 7.0000 2.0000 0+3.0000i
>> ceil(a)
-1.0000 0 4.0000 6.0000 7.0000 3.0000 0+4.0000i
>> floor(a)
-2.0000 -1.0000 3.0000 5.0000 7.0000 2.0000 0+3.0000i
>> round(a)
-2.0000 0 3.0000 6.0000 7.0000 2.0000 0+4.0000i
```

b. Hàm lượng giác cơ bản

Bảng 2.3: Các hàm lượng giác cơ bản

Hàm	Ý nghĩa
sin(x)	sin của x khi x có đơn vị radian
cos(x)	cos của x khi x có đơn vị radian
tan(x)	tan của x khi x có đơn vị radian
asin(x)	$\in [-\pi/2, \pi/2]$ khi $x \in [-1, 1]$
acos(x)	$\in [0, \pi]$ khi $x \in [-1, 1]$
atan(x)	khi $x \in [-\pi/2, \pi/2]$

Đổi radian sang độ và ngược lại:

$$angle_degrees = angle_radians * (180/\pi)$$

$$angle_radians = angle_degrees * (\pi/180)$$

Ví dụ 1: Tính sin theo rad ta có:

```
>> a = sin(pi/6)
```

```
a =
```

```
0.5000
```

Ví dụ 2: Tính sin theo độ:

```
>> sind(90)
```

```
ans =
```

```
1
```

2.3 Biểu thức quan hệ và hàm logic

2.3.1 Các phép toán quan hệ

Các toán tử quan hệ cũng có thể làm việc trên cả dữ liệu vô hướng và phi vô hướng. Các toán tử quan hệ cho mảng thực hiện so sánh từng phần tử một giữa 2 mảng và trả về mảng logic có cùng kích thước, với các phần tử được thiết lập là 1 (true), trong đó quan hệ là true và các phần tử được thiết lập là 0 (false). Dưới đây là bảng danh sách các toán tử quan hệ trong MATLAB:

Bảng 2.4: các phép toán quan hệ

Toán tử	Ý nghĩa
<	Nhỏ hơn
<=	Nhỏ hơn hoặc bằng
>	Lớn hơn
>=	Lớn hơn hoặc bằng
==	Bằng
~=	Không bằng

Phép so sánh 2 ma trận là so sánh từng phần tử. Kết quả sinh ra ma trận {0,1} cùng cỡ. Nếu phép so sánh đúng, các phần tử =1, ngược lại thì các phần tử bằng 0

Ví dụ:

```

>> a=[3 4 3; 4 5 6];
>> b=[1 2 3; 7 8 6];
>> a==b
ans =
     0 0 1
     0 0 1

>> a>b
ans =
     1 1 0
     0 0 0

>> a>=b
ans =
     1 1 1
     0 0 1

```

2.3.2 Các phép toán logic

MATLAB cung cấp 2 kiểu toán tử logic và các hàm:

- **Element-wise**: Các toán tử thực hiện trên các phần tử tương ứng của các mảng logic.

- **Short-circuit**: Các toán tử thực hiện trên các biểu thức tích vô hướng và biểu thức logic.

Các toán tử logic Element-wise thực hiện từng phần tử một trên các mảng logic. Các biểu tượng &, |, và ~ là các toán tử mảng logic AND, OR, và NOT.

Các toán tử (Operator) trong MATLAB logic Short-circuit cho phép bỏ qua các phép tính logic. Các ký tự && và || là các toán tử logic short-circuit AND và OR.

Thứ tự các toán tử trong biểu thức logic từ cao đến thấp là not, and, or. Tuy nhiên có thể dùng ngoặc đơn để thay đổi

Trong Matlab, tất cả các giá trị khác không đều coi như đúng (true), còn giá trị 0 được coi như sai (false)

Ví dụ:

```

>> b=[1 1 0; 1 0 1]
>> a=[0 1 0; 0 0 1]
>> a&b
ans = 0 1 0
     0 0 1

>> a|b
ans = 1 1 0
     1 0 1

>> ~a
ans = 1 0 1
     1 1 0

```

>> 1 | 0 & ~ 1

2.3.3 Các hàm quan hệ và logic

Bảng 2.5: bảng các hàm quan hệ và logic

Hàm	Ý nghĩa
any(x)	Trả về vector hàng có các phần tử =1 nếu tồn tại bất kỳ phần tử cột của x khác 0, ngược lại =0
all(x)	Trả về vector hàng có các phần tử =1 nếu tất cả phần tử cột của x khác 0, ngược lại =0
find(x)	Trả về vector chứa chỉ số các phần tử của x khác 0
exist('a')	= 1 nếu a là biến, = 2 nếu là file, = 0 nếu a không tồn tại...
isnan(x)	Trả về ma trận cùng cỡ có các phần tử = 1 nếu các phần tử tương ứng của x là nan, ngược lại = 0
finite(x)	Trả về ma trận cùng cỡ có các phần tử = 1 nếu các phần tử tương ứng của x hữu hạn, = 0 nếu vô hạn hoặc nan
isempty(x)	= 1 nếu x rỗng, ngược lại = 0
isstr(x)	= 1 nếu x là một chuỗi, ngược lại = 0
strcmp(y1,y2)	So sánh 2 chuỗi, =1 nếu 2 chuỗi giống hệt nhau, ngược lại =0. Phân biệt hoa-thường, dấu cách, đầu dòng

Ví dụ:

```
>> a=[0 1 2; 0 0 3];
>> any(a)
ans = 0     1     1
>> all(a)
ans = 0     0     1
>> find(a)
ans = 3     5     6
>> a=[nan 12 4 0; inf 3 8 nan]
a = NaN     12     4     0
      Inf     3     8     NaN
>> isnan(a)
ans = 1     0     0     0
      0     0     0     1
>> finite(a)
ans = 0     1     1     1
      0     1     1     0
>> isempty(a)
ans = 0
>> text1='Lop HCDH';
```

```

>> text2='Lop';
>> text3='HCDH';
>> isstr(text1)
ans = 1
>> strcmp(text1,text2)
ans = 0
>> strcmp(text1,[text2 ' ' text3])
ans = 1

```

2.3.4. Symbolic Math Toolbox

Bộ công cụ bổ sung khả năng giải toán với các ký hiệu toán học cho MATLAB. Lõi của bộ công cụ này được phát triển bởi Maple. Nó cho phép thực hiện các phép toán sau:

- *Calculus*: đạo hàm, tích phân, giới hạn, chuỗi.
- *Đại số tuyến tính*: nghịch đảo, định thức, giá trị eigen, Inverses, determinants, eigenvalues, singular value decomposition, and canonical forms of symbolic matrices.
- *Rút gọn*: dùng để rút gọn biểu thức.
- *Giải phương trình*: đại số và vi phân
- *Các hàm đặc biệt*: cung cấp các hàm đặc biệt như beta, bessel, gamma.
- *Transforms*: Fourier, Laplace, z-transform.

Symbolic Math Toolbox định nghĩa một kiểu dữ liệu mới của Matlab gọi là đối tượng Symbolic. Một đối tượng Symbolic là một cấu trúc dữ liệu lưu trữ một đại diện kiểu xâu ký tự của một đối tượng. Symbolic Math Toolbox sử dụng các đối tượng Symbolic để biểu diễn các biến, biểu thức và ma trận Symbolic.

Tạo đối tượng (biến, biểu thức) Symbolic

Lệnh *sym* cho phép xây dựng các biến và biểu thức Symbolic. Ví dụ

```

>>x = sym('x'); y = sym('y') ; lệnh này tạo ra x, y là các biến Symbolic
hoặc >> syms x y
>>f = syms('a*x^2 +b*x+c') ; gán biểu thức  $ax^2 + bx + c$  vào biến f
>> x = sym('x', 'real') ; y = sym('y', 'real') ; x,y là biến thực kiểu Symbolic
>> syms x y real % x,y là biến thực kiểu Symbolic
>> syms x real y % x là biến thực kiểu Symbolic,y là biến bất kỳ kiểu Symbolic
>> syms t
>> Q = sym('Q(t)'); % t là biến Symbolic , Q là hàm Symbolic

```

Đạo hàm

Để tính đạo hàm của một biểu thức Symbolic ta sử dụng hàm `diff()`

- `diff(S)`: Đạo hàm biểu thức symbolic S với biến tự do được xác định bởi hàm `findsym(S)`

- `diff(S,v)`: Đạo hàm biểu thức symbolic S với biến lấy đạo hàm là v

- `diff(S,v,n)`: Đạo hàm cấp n biểu thức symbolic S với biến lấy đạo hàm là v

Tích phân

Để tính tích phân của một biểu thức Symbolic ta sử dụng hàm `int()`

- `int(S,v)` : Tích phân không xác định của biểu thức Symbolic với biến tích phân v

- `int(S,v,a,b)` : Tích phân xác định của biểu thức Symbolic với biến tích phân v và cận lấy tích phân từ $[a, b]$

Biến đổi Laplace thuận

- `L = laplace(F)`: Biến đổi Laplace của hàm F với biến mặc định độc lập t , nó cho ta một hàm của s .

- `L = laplace(F,x)`: Biến đổi Laplace của hàm F với biến mặc định độc lập t , nó cho ta một hàm của x .

- `L = laplace(F,w,z)`: Biến đổi Laplace của hàm F với biến độc lập w , nó cho ta một hàm của z .

Biến đổi Laplace ngược

- `F = ilaplace(L)`: Biến đổi Laplace ngược của hàm L với biến mặc định độc lập s , nó cho ta một hàm của t

- `F = ilaplace(L,x)`: Biến đổi Laplace ngược của hàm L với biến mặc định độc lập s , nó cho ta một hàm của x

- `F = ilaplace(L,w,z)`: Biến đổi Laplace của hàm L với biến độc lập w , nó cho ta một hàm của z

Bài tập cuối chương

Bài 1: Thực hiện lệnh sau và nhận xét kết quả trả về

a. `a = 1; b = 2; c = 3; (b*b==4*a*c)` và `a~=0`

b. `a = 1:10; a>=20` và `a<=20`

Bài 2: Thực hiện lệnh sau đây

1. `r=1; r<=0.5`

2. `r=1:5; r<=3`

3. `a=1:5; b=[0 2 4 5 6]; a==b`

Bài 3: Nhân hai đa thức sau đó sử dụng Matlab để kiểm tra

$$(3x^2+4x+5).(2x^3-3x^2+2)$$

Bài 4: Chia 2 đa thức sau đây, sau đó sử dụng Matlab kiểm tra lại

$$(2x^2+3x+6)/(2x+3)$$

Bài 5: Tìm nghiệm của phương trình sau đây sử dụng Matlab: $X^2-1=0$

Bài 6: cho $x = [4 \ 1 \ 6]$ và $y = [6 \ 2 \ 7]$ tính các mảng, vector, ma trận sau

a. $a_{ij}=x_i y_j$

b. $b_{ij}=x_i / y_j$

Bài 7. Dùng phép toán trên ma trận tìm tất cả các phân tử chia hết cho 3. Biết vector đã cho có 100000 phân tử trong khoảng từ [0, 100]

Bài 8: Cho $x = [1 \ 3 \ 5 \ 8 \ 9 \ 0 \ 1]$ Viết dòng lệnh để

a. Tính tổng các phân tử

b. Viết một hàm tính tổng dãy con từ 1 đến j.

Bài 9: Tạo một ma trận $M \times N$ các số ngẫu nhiên. Chuyển các giá trị nhỏ hơn 0,2 thành 0 và các giá trị lớn hơn 0,2 thành 1.

Bài 10: từ một vector số ngẫu nhiên bất kỳ xác định có bao nhiêu phần tử đứng trước một phần tử có giá trị nằm từ 0,8 đến 0,85.

Câu hỏi trắc nghiệm

Câu 1: Trong MATLAB các để tìm nghiệm của đa thức p là

A. zeros(p)

B. p(m, n)

C. r = roots(p)

D. Adds(a,0)

Câu 2: Để nhân hai đa thức ta thực hiện cú pháp

A. c=con(a, b)

B. a.b

C. c=cov(a, b)

D. c=conv(a, b)

Câu 3: Thực hiện chia đa thức theo cú pháp nào dưới đây

A. [q, r]=deconv(u,v)

B. [q, r]=conv(u,v)

C. [q, r]=(u :v)

D. max(z)

Câu 4: Matlab tính nguyên hàm của đa thức bởi lệnh

A. length(h, k)

B. z=roots(n)

C. g=polyint(h, k)

D. z=roots(h, k)

Câu 5: Trả về vector hàng có các phần tử =1 nếu tồn tại bất cứ phần tử cột của x khác 0, ngược lại =0

A. all(x)

B. any(x)

C. isnan(x)

D.

find(x)

Chương III: ĐỒ HOẠ TRONG MATLAB

Nội dung chính của chương

Chương 3 giúp sinh viên vẽ được đồ thị của một hàm bất kỳ, tạo các dạng đồ thị khác nhau với đồ thị dạng 2D và đồ thị 3D. Thay đổi màu sắc, gộp đồ thị hay thay đổi dạng biểu diễn của đồ thị.

Mục tiêu cần đạt được của chương

Sinh viên cần sử dụng thành thạo Matlab trong việc vẽ các loại đồ thị 2D và 3D

Bài 3: Đồ họa trong Matlab (Số tiết: 3 tiết)

3.1 Đồ họa 2D

3.1.1 Sử dụng lệnh Plot

Để vẽ đồ thị trong mặt phẳng người ta sử dụng lệnh **plot**. Lệnh này sẽ vẽ đồ thị một mảng dữ liệu trong một hệ trục thích hợp và nối các điểm bằng đường thẳng.

Ví dụ:

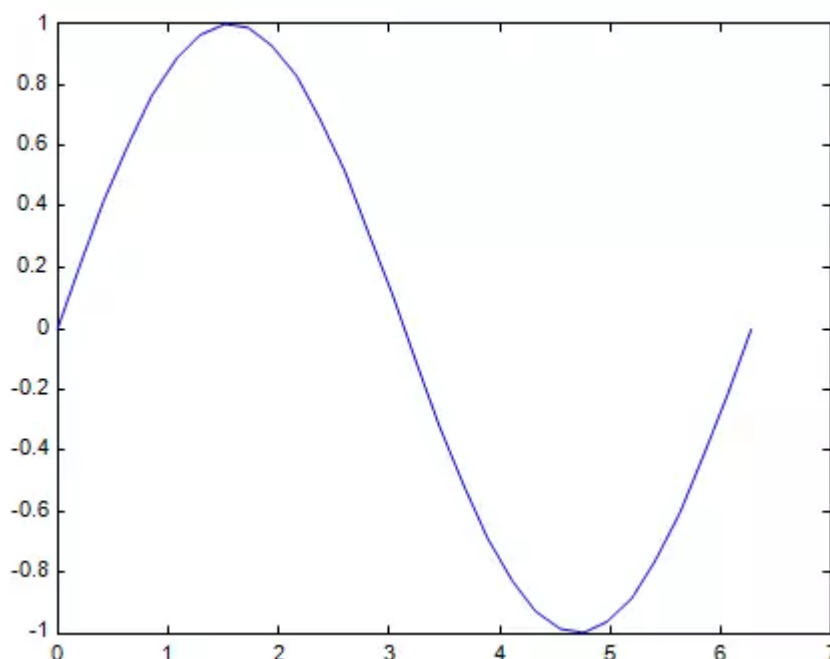
```
>> x=linspace(0,2*pi,30);
```

// Hàm linspace là lấy 30 điểm cách đều nhau trong khoảng từ 0 -> 2pi theo chiều ngang đồ thị.

```
>> y=sin(x); // tạo một vector y là hàm sin của x.
```

```
>> plot(x,y);
```

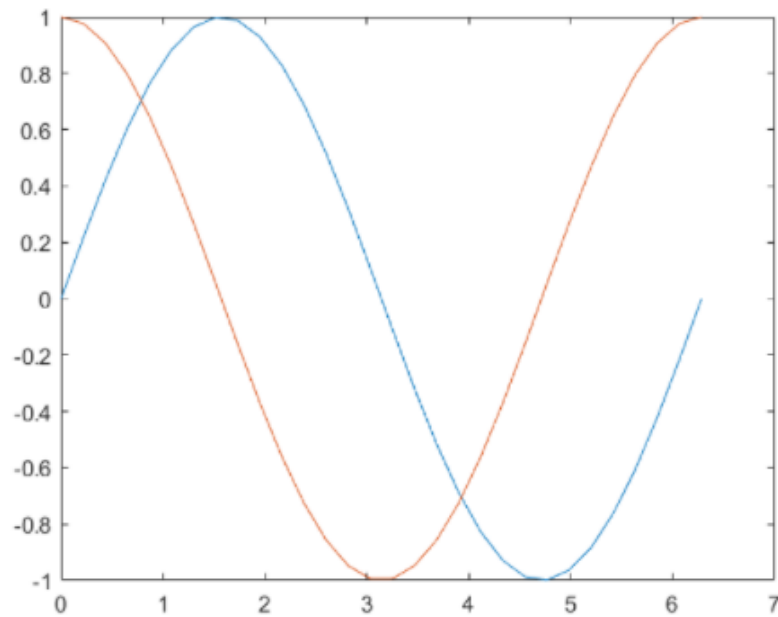
Lệnh plot mở ra một cửa sổ đồ họa gọi là cửa sổ **figure**, trong cửa sổ này sẽ tạo độ chia phù hợp với dữ liệu, vẽ đồ thị qua các điểm và đồ thị được tạo thành bởi việc nối các điểm này bằng đường nét liền. Các thang chia số và dấu được tự động cập nhật vào.



Hình 3.1: Đồ thị của hàm $y = \sin(x)$ được vẽ trên Matlab

Cùng vẽ hàm sin và cos trên cùng một đồ thị

```
>>z=cos(x);
>> plot(x,y,x,z);
```



Hình 3.2: đồ thị sin và cos trên cùng một hệ trục tọa độ.

Nếu muốn vẽ nhiều hơn một đồ thị trên cùng một hình vẽ ta chỉ việc đưa thêm vào **plot** một cặp đối số. Khi đó đồ thị thứ hai sẽ tự động được vẽ bằng màu khác trên màn hình.

Nếu như một trong các đối số là ma trận và đối số còn lại là vector thì lệnh **plot** sẽ được vẽ tương ứng mỗi cột của ma trận với vector đó.

```
>>W=[y;z];
>>plot(x,W)
```

3.1.2 Kiểu đường, dấu và màu

Trong Matlab ta có thể khai báo kiểu màu, nét vẽ của riêng bằng cách đưa vào plot một đối số thứ 3 sau mỗi cặp dữ liệu của mảng. các đối số tùy chọn này là một chuỗi ký tự, có thể chứa một hoặc nhiều người hơn theo bảng dưới đây.

Nếu không khai báo mẫu thì Matlab sẽ chọn màu mặc định là Blue. Kiểu đường mặc định là kiểu solid. Nếu một màu, dấu, kiểu đường tất cả điều chứa trong một chuỗi thì kiểu màu chung cho cả dấu và kiểu nét vẽ. Để khai báo màu khác cho dấu cần vẽ cùng một dữ liệu với các kiểu khai báo chuỗi khác nhau.

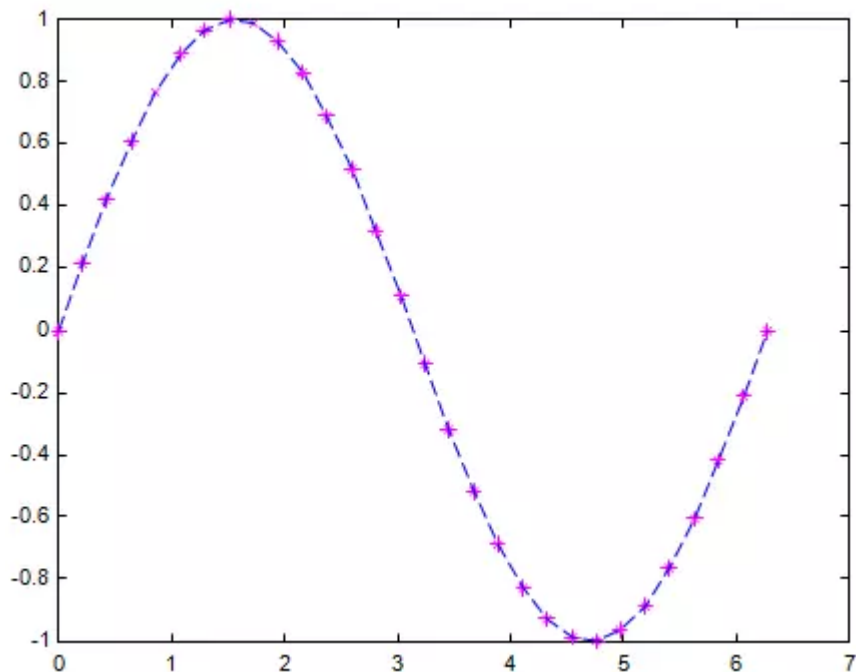
Bảng 3.1. Bảng màu sắc và ký tự trong vẽ đồ thị Matlab

Kí tự	Màu	Kí tự	Đánh dấu	Kí tự	Kiểu Nét Vẽ
b	xanh da trời	.	chấm	-	nét liền
g	xanh lá cây	o	vòng tròn	:	nét chấm
r	đỏ	x	dấu x	-.	nét gạch - chấm
c	xanh da trời nhạt	+	dấu +	--	nét đứt

m	đỏ tím	*	dấu hoa thị	-	nét liền
y	vàng	s	hình vuông		
k	đen	d	hình thoi		
w	trắng	^	tam giác hướng xuống		
		v	tam giác hướng lên		
		<	tam giác hướng phải		
		>	tam giác hướng trái		
		vp	sao năm cánh		
		h	sao sáu cánh		

Ví dụ:

```
>> plot(x,y,'m*',x,y,'b--')
```



Hình 3.3: Đồ thị biểu diễn với màu sắc và ký hiệu riêng.

3.1.3 Kiểu đồ thị

Lệnh **colordef** cho phép lựa chọn kiểu hiển thị. Giá trị mặc định của kiểu hiển thị là white. Kiểu này sử dụng trục tọa độ, màu nền nền hình vẽ là màu xám sáng và tên tiêu đề của trục là màu đen. Nếu muốn đổi màu nền thành màu đen ta có thể dùng lệnh `>> colordef black`

3.1.4 Đồ thị lưới, hộp chứa trục, nhãn và lời chú giải

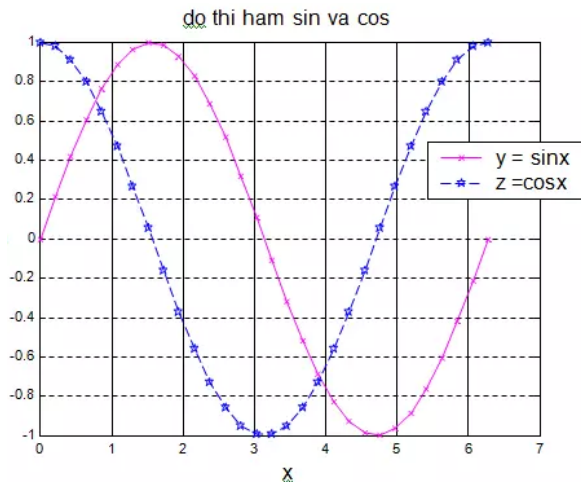
Lệnh **grid on** sẽ thêm đường lưới vào đồ thị hiện tại. Lệnh **grid off** xóa bỏ các nét này. Matlab khởi tạo với **griff off**. Thông thường trục tọa độ có nét kiểu **solid** nên gọi là hộp chứa trục. Hộp này có thể tắt đi nhờ lệnh **box off** và **box on** sẽ khôi phục lại. Ta có thể đưa tên trục x, y và tên của đồ thị vào hình vẽ nhờ các lệnh **xlabel** và **ylabel**.

Lệnh **title** sẽ thêm vào đồ thị tiêu đề ở đỉnh. Dòng ghi chú được đưa vào đồ thị nhờ hàm **legend**. Trong **legend** thì màu và kiểu của mỗi loại đường phù hợp với các đường đó trên đồ thị.

```

Ví dụ:
>> x=linspace(0,2*pi,30);
>> y=sin(x);
>> z=cos(x);
>> plot(x,y,'mx-',x,z,'bp--')
>> grid on
>> xlabel('x')
>> ylabel('y')
>> title('do thi ham sin va cos')
>> legend('y = sinx','z =cosx')

```



Hình 3.4: Đồ thị sin, cos có tên đồ thị, đường lưới, tên các trục và ghi chú

Ta có thể thêm nét vẽ vào đồ thị đã có sẵn bằng cách dùng lệnh **hold**. Khi dùng lệnh **hold on**, MATLAB không bỏ đi hệ trục đã tồn tại trong khi lệnh **plot** mới đang được thực hiện, thay vào đó, nó thêm đường cong mới vào hệ trục hiện tại. Tuy nhiên, nếu dữ liệu không phù hợp hệ trục tọa độ cũ, thì trục được chia lại. Dùng lệnh **hold off** sẽ bỏ đi cửa sổ figure hiện tại và thay vào bằng một đồ thị mới. Lệnh **hold** không có đối số sẽ bật tắt chức năng của chế độ thiết lập **hold** trước đó

```

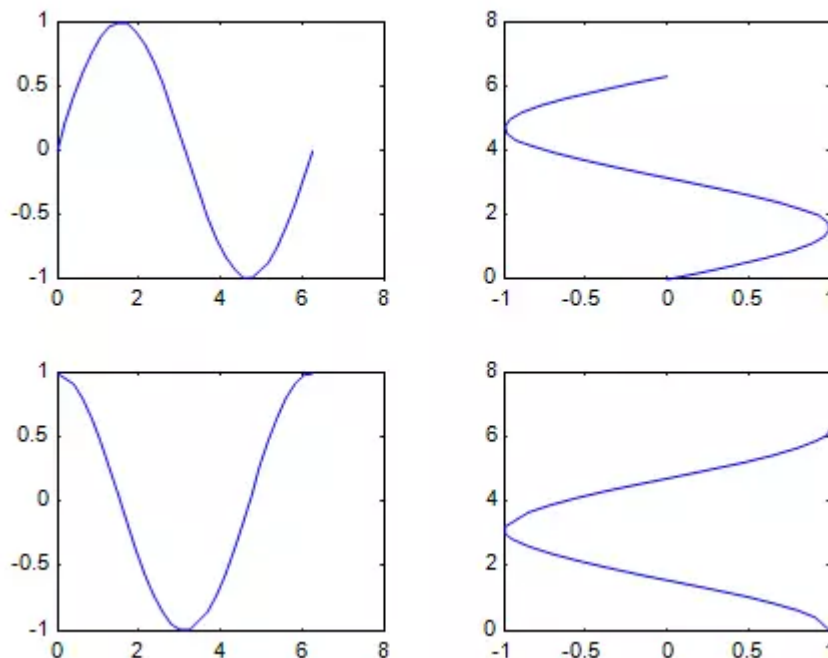
Ví dụ
>> x=linspace(0,2*pi,30);
>> y=sin(x);
>> z=cos(x);
>> plot(x,y);
    • Hình vẽ sẽ hiện ra trên cửa sổ figure
>> hold on plot(x,z,'m')
    • Sẽ vẽ thêm 1 đồ thị khác vào cùng cửa sổ figure
>> hold off

```

Lệnh **subplot(m,n,p)** chia cửa sổ hiện tại thành một ma trận $m \times n$ khoảng để vẽ đồ thị, và chọn p là cửa sổ hoạt động. Các đồ thị thành phần được đánh số từ trái qua phải, từ trên xuống dưới, sau đó đến hàng thứ hai

Ví dụ:

```
>> subplot(2,2,1)
>> plot(x,y)
>> subplot(2,2,2)
>> plot(y,x)
>> subplot(2,2,3)
>> plot(x,z)
>> subplot(2,2,4)
>> plot(z,x)
```



Hình 3.5: Vẽ các đồ thị xy , yx , xz , zx

3.1.5. Tập lệnh liên quan vẽ đồ thị

Lệnh AXES

- a) Chức năng: Đặt các trục tọa độ tại vị trí định trước
- b) Cú pháp

`axes('propertyname', propertyvalue ...)`

1. 'position', [left, bottom, width, height]: định vị trí và kích thước của trục

```
>> axes('position',[1, 1, 8, 6]    %căn 2 bên là 1 chiều rộng trục ngang là 8 trục đứng là 6
```

2. 'xlim', [min,max]: xác định giá trị nhỏ nhất và lớn nhất trên trục x

3. 'ylim', [min,max]: xác định giá trị nhỏ nhất và lớn nhất trên trục y

Ví dụ:

```
>> axes('xlim', [2 5])
```

```
>> axes('ylim', [2 5])
```

Lệnh AXIS

a. Chức năng: thực hiện chia lại trục tọa độ

b. Cú pháp

```
axis([xmin xmax ymin ymax])
```

```
axis([xmin xmax ymin ymax zmin zmax])
```

```
axis on
```

```
axis off
```

Với

xmin, ymin, zmin Là giá trị nhỏ nhất của các trục x, y, z.

xmax, ymax, zmax Là giá trị lớn nhất của các trục x, y, z.

on Cho hiển thị trục tọa độ.

off Không cho hiển thị trục tọa độ.

Lệnh Bar

a. Chức năng: Vẽ đồ thị dạng cột

b. Cú pháp: bar(x,y)

Ví dụ:

```
>> x = [-pi:0.2:pi];
```

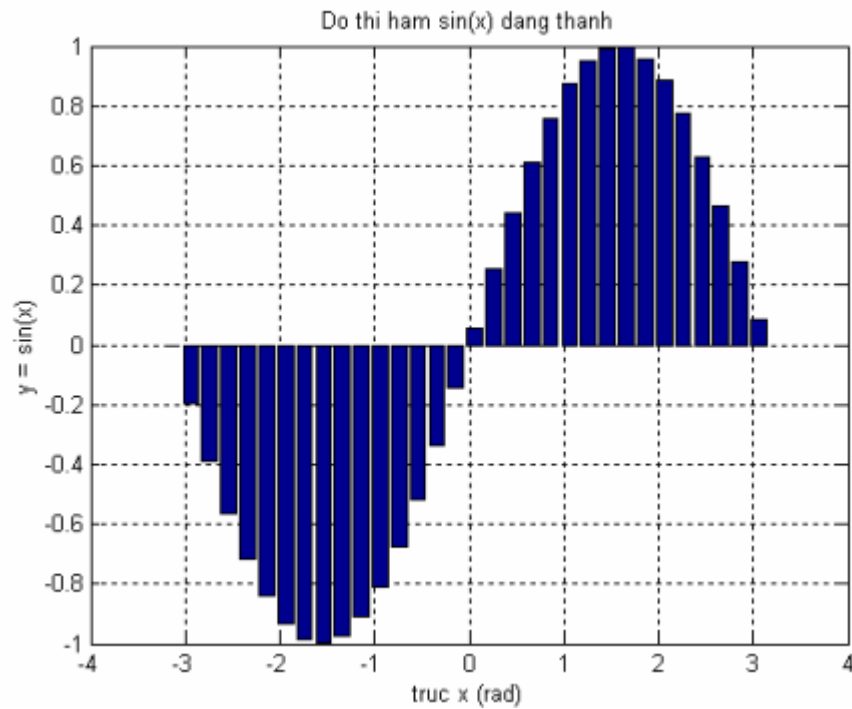
```
>> bar(x, sin(x));
```

```
>> grid on;
```

```
>> title('Đồ thị hàm sin(x) dạng thanh');
```

```
>> xlabel('trục x (rad)');
```

```
>> ylabel('y = sin(x)');
```



Hình 3.6: Đồ thị dạng cột

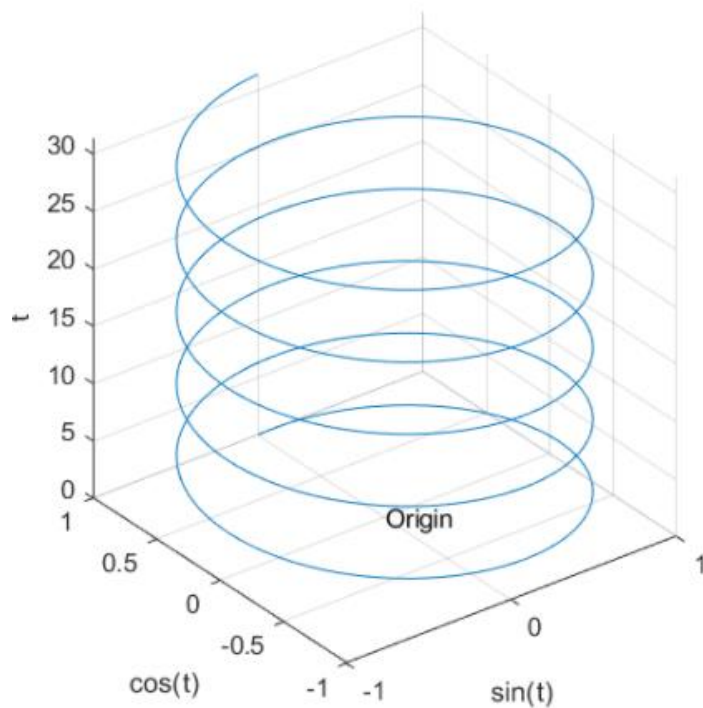
3.2. Đồ họa 3D

3.2.1. Đồ thị đường thẳng

Lệnh **Plot** trong không gian 2 chiều có thể mở rộng cho không gian 3 chiều bằng lệnh **plot3**. Lệnh **plot3** cho phép vẽ các điểm và đường trong không gian. Ngoài việc có thêm trục z, cách sử dụng hàm này giống như cách sử dụng hàm **plot**. **View([a,b])**: a là góc phương vị tính bằng độ ngược chiều kim đồng hồ từ phía âm của trục y. Giá trị mặc định của a là -37.5 độ. b là góc nhìn tính bằng độ xuống mặt phẳng x, y. Giá trị mặc định của b là 30 độ. Khi thay đổi các giá trị a và b sẽ nhìn được hình vẽ dưới các góc độ khác nhau.

Ví dụ:

```
>> t=linspace(0,10*pi);
>> plot3(sin(t),cos(t),t);
>> xlabel('sint');
>> ylabel('cost');
>> zlabel('t')
>> axis square;           % Thiết lập giới hạn của 3 trục theo hình vuông.
>> grid on;
>> text(0,0,0,'Origin')
```



Hình 3.7: đồ thị 3D hàm sin, cos

3.2.2. Đồ thị bề mặt và lưới

Matlab định nghĩa bề mặt lưới bằng các điểm theo hướng trục z ở trên đường kẻ ô hình vuông trên mặt phẳng xy. Nó tạo lên mẫu một đồ thị bằng cách ghép các điểm gần kề với các đường thẳng. Kết quả nó trông như mạng lưới với các mắt lưới là các điểm dữ liệu. Đồ thị lưới này được sử dụng để quan sát những ma trận lớn hoặc vẽ những hàm có hai biến.

Bước đầu tiên là đưa ra đồ thị lưới các hàm hai biến $z=f(x,y)$ tương ứng với ma trận X và Y chứa các hàng và các cột lặp đi lặp lại. Matlab cung cấp hàm **meshgrid** với mục đích này. $[X, Y]=\text{meshgrid}(x, y)$, Tạo ma trận X mà các hàng của nó là bản sao của vector x, và ma trận Y có cột của nó là bản sao của vector y. Cặp ma trận này sau đó được sử dụng để ước lượng hàm hai biến sử dụng đặc tính toán học về mảng của Matlab.

Để vẽ bề mặt ta sử dụng các hàm: **mesh (X,Y,Z)**: nối các điểm với nhau trong một lưới chữ nhật. **meshc (X,Y,Z)**: vẽ các đường contour bên dưới đồ thị. **meshz (X,Y,Z)**: vẽ các đường thẳng đứng viền quanh đồ thị. **waterfall (X,Y,Z)**: vẽ mặt với hiệu ứng như thác đổ.

Ví dụ: Vẽ mặt xác định bởi phương trình: $z(x, y) = x \cdot e^{-(x^2 - y^2)}$

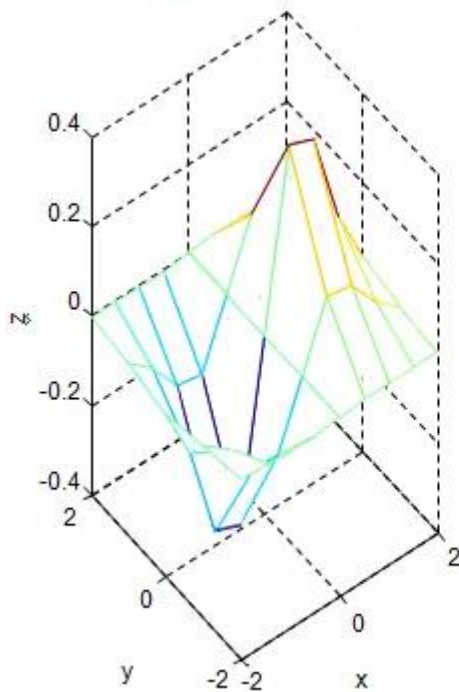
```
>> x=-2:0.5:2;
>> y=-2:1:2;
>> [X,Y]=meshgrid(x,y)
>> Z=X.*exp(-X.^2-Y.^2)
>> subplot(1,2,1)
```



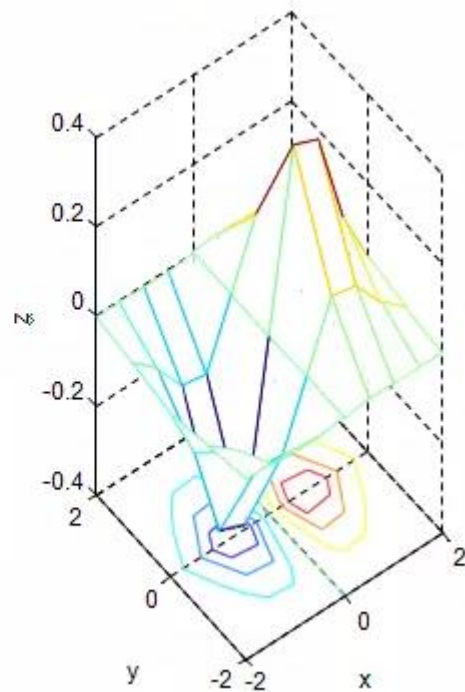
```

>> mesh(X,Y,Z)
>> xlabel('x')
>> ylabel('y')
>> zlabel('z')
>> title('ve mat voi lenh mesh')
>> subplot(1,2,2)
>> meshc(X,Y,Z)
>> xlabel('x')
>> ylabel('y')
>> zlabel('z')
>> title('ve mat voi lenh meshc')

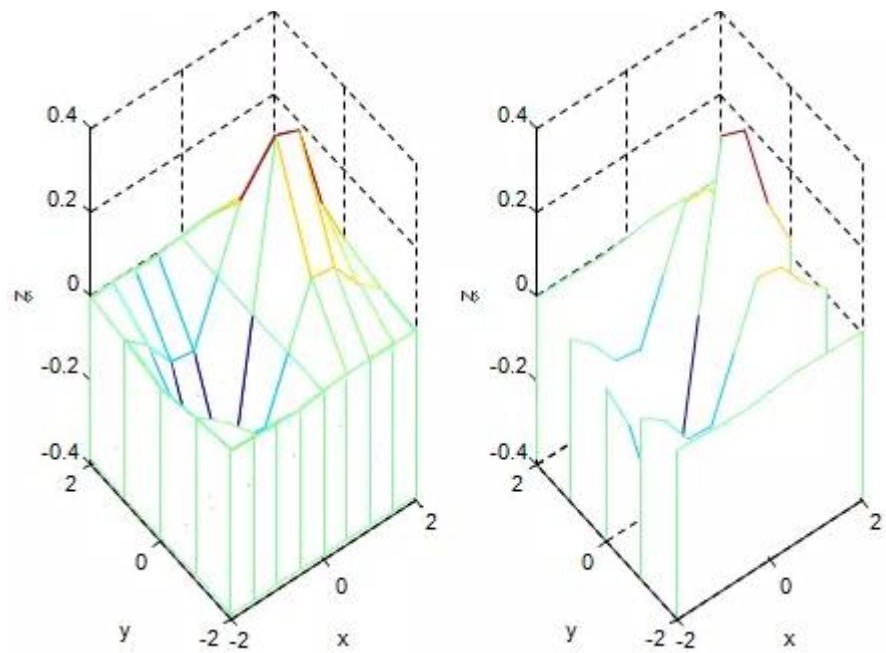
```



a) Vẽ với lệnh mesh



b) Vẽ mặt với lệnh meshc



c) Vẽ mặt với lệnh `meshz`

d) Vẽ mặt với lệnh `waterfall`

Hình 3.8: Đồ thị 3D vẽ mặt với các lệnh.

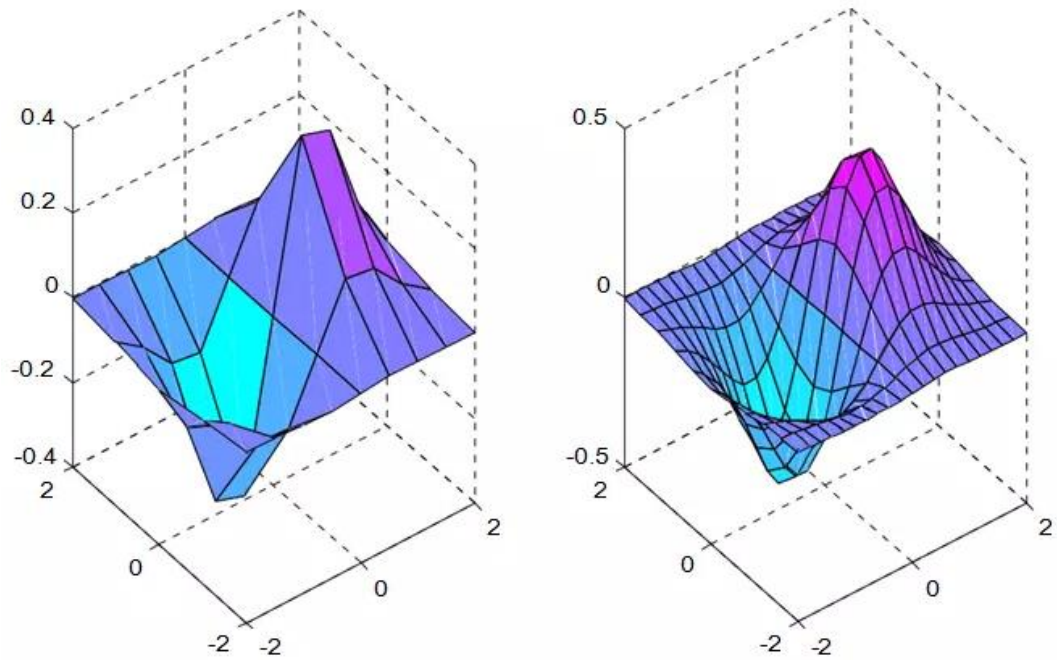
Vẽ các mặt được tô bóng từ một ma trận bằng các lệnh `surf`, `surfc` Lệnh **surf** (**X,Y,Z**): vẽ mặt có các đường contour phía dưới. Lệnh **surfl** (**X,Y,Z,s**): vẽ mặt có bóng sáng. Đối số `s` xác định hướng của nguồn sáng trên bề mặt vẽ. `s` là một vectơ tùy chọn trong hệ tọa độ decac hay trong tọa độ cầu. Nếu không khai báo giá trị mặc định của `s` là 45° theo chiều kim đồng hồ từ vị trí người quan sát. Khi vẽ đồ thị ta có thể thay đổi một số đặc điểm của đồ thị như tỉ lệ trên các trục, giá trị giới hạn của các trục, màu và kiểu đường cong đồ thị, hiển thị legend... ngay trên figure bằng cách vào menu tools rồi vào mục axes properties, line properties hay show legend...

```
>> x=-2:0.5:2;
>> y=-2:1:2;
>> [X,Y]=meshgrid(x,y);
>> Z=X.*exp(-X.^2-Y.^2);
>>surf(X,Y,Z)
>> colormap(hot)
```

Ta có thể tạo nhiều lưới hơn để có một mặt mịn hơn:

```
>> x=-2:0.2:2;
>> y=-2:0.4:2;
>> [X,Y]=meshgrid(x,y);
>> Z=X.*exp(-X.^2-Y.^2);
```

```
>> surf(X,Y,Z)
>> colormap(cool)
```



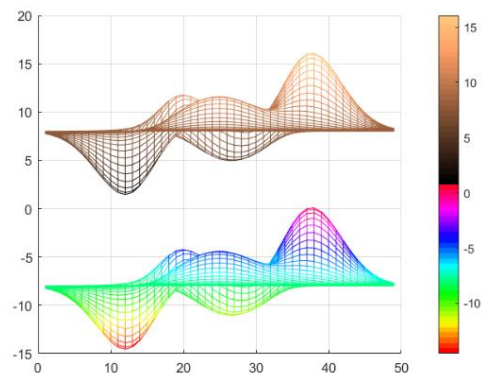
Hình 3.9: Vẽ mặt có bóng sáng

3.2.3. Thao tác với đồ thị

Matlab cho phép khai báo góc để từ đó quan sát được đồ thị trong không gian 3 chiều. Sử dụng hàm: **view(azimuth, elevation)** thiết lập góc xem với **azimuth** mô tả góc trong hệ trục nơi người quan sát đứng và **elevation** mô tả vị trí người quan sát.

Ví dụ:

```
>>view(90,0)
>>hold on
```



Hình 3.10: xem đồ thị với góc xem khác

Bài tập cuối chương

Bài 1: Vẽ đồ thị hàm số $y=2x$

Bài 2: Vẽ đồ thị hàm số $y=x^3$

Bài 3: Vẽ đồ thị hàm số $y=e^x$

Bài 4: Vẽ đồ thị hàm số $y = \sin 3x$ và $y = x^2$ trên cùng một đồ thị, ghi chú thích.

Bài 5: Vẽ hàm số $y = x^3 - 3x + 1$ sử dụng hàm plot và flot.

Bài 6: Vẽ đồ thị 3D của hàm $y = \cos x$

Bài 7: Vẽ đồ thị 3D dùng lệnh view để thay đổi góc quan sát hàm $y = x^2 + 3x + 1$

Bài 8: Vẽ đồ thị hàm số $y = \sin x$ và $y = 2x^2 - 1$ trên cùng một đồ thị, ghi chú thích.

Bài 9: Vẽ đồ thị 3D hàm số $y = x^3 - 3x + 1$ sử dụng hàm plot.

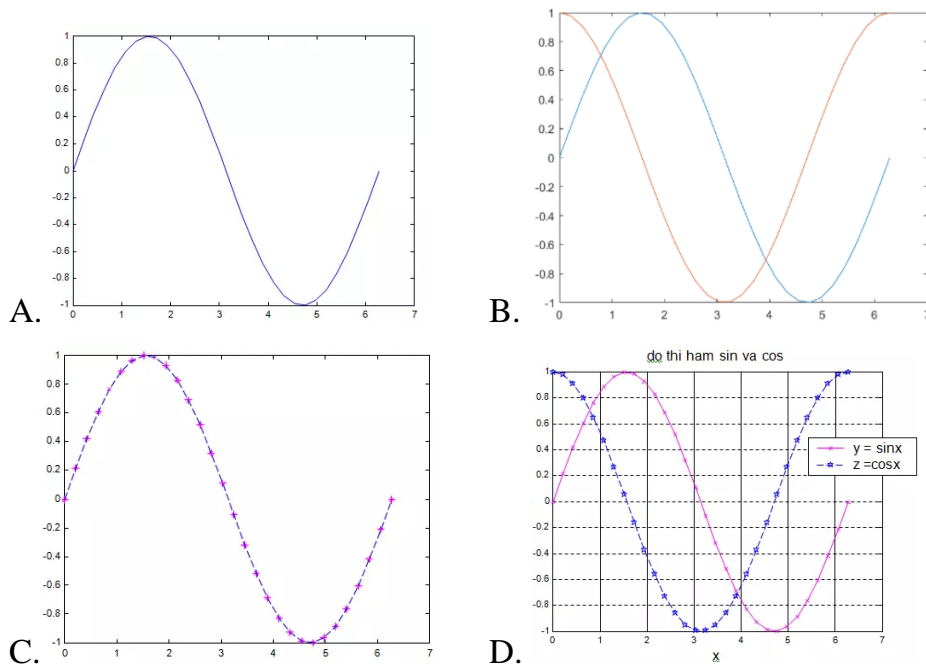
Bài 10: Vẽ đồ thị của hàm $y = \cos x$

Câu hỏi trắc nghiệm

Câu 1: Để vẽ đồ thị trong mặt phẳng người ta sử dụng lệnh nào sau đây?

- A. plot(x,y) B. drawn(x, y) C. r = roots(p) D. Adds(a,0)

Câu 2: Lệnh sau đây plot(x,y,'m*',x,y,'b--') đưa ra dạng đồ thị nào?



Câu 3: Để chia cửa sổ hiện tại thành một ma trận $m \times n$ khoảng để vẽ đồ thị sử dụng lệnh nào

- A. [q, r]=deconv(u,v) B. subplot(m,n,p) C. sub(m,n,p) D. plot(m,n,p)

Câu 4: Vẽ đồ thị trong không gian 3 chiều ta sử dụng lệnh nào sau đây?

- A. t=linspace(0,10*pi) B. grid on
C. g=polyint(h, k) D. plot3(sin(t),cos(t),t)

Câu 5: Hàm mesh (X,Y,Z) nhằm mục đích gì?

- A. vẽ bề mặt mà nối các điểm với nhau trong một lưới chữ nhật
B. Vẽ đồ thị 3 chiều
C. Nối các điểm x, y, z với nhau
D. Vẽ các đường thẳng đứng viền quanh đồ thị

Chương IV : MÔ PHỎNG HỆ THỐNG VỚI MATLAB/SIMULIK

Nội dung chính của chương

Chương 4 giúp sinh viên nắm được các thư viện cơ bản trong Simulink thực hiện xây dựng mô hình mô phỏng cho các đối tượng và hệ thống bằng Simulink

Mục tiêu cần đạt được của chương

Sinh viên cần sử dụng thành thạo các thư viện trong simulik để xây dựng mô hình mô phỏng cho các đối tượng và hệ thống

Bài 4: Công cụ simulink và simpower system (số tiết: 3 tiết)


4.1. Công cụ simulink và simpower system

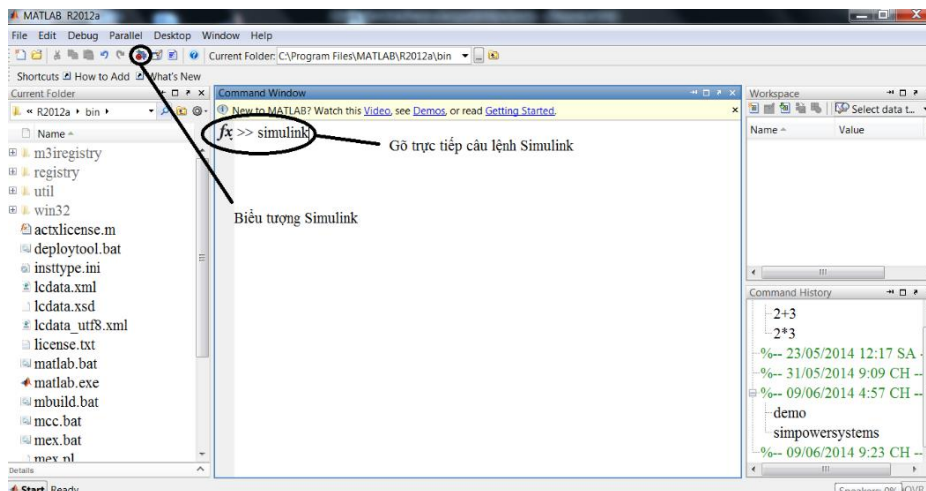
4.1.1. Giới thiệu chung về Simulink

Simulink là một phần mở rộng của Matlab dùng để mô hình hóa, mô phỏng và khảo sát các hệ thống động học; thiết kế các hệ thống điều khiển...

Simulink cho phép mô tả hệ thống tuyến tính, phi tuyến, các mô hình trong miền thời gian liên tục hay gián đoạn hoặc một hệ gồm cả liên tục và gián đoạn.

Simulink cung cấp cho người dùng giao diện dễ sử dụng bằng cách ghép cách ghép các khối chức năng quen thuộc. Chính vì vậy ta có thể xây dựng và khảo sát hệ thống một cách trực quan. Simulink cung cấp cho người sử dụng một thư viện phong phú với số lượng lớn các khối chức năng cho các hệ tuyến tính, phi tuyến hay gián đoạn. Simulink cũng cho phép người sử dụng tự tạo các khối chức năng của riêng mình.

*Cách vào môi trường làm việc Simulink:*Đầu tiên chúng ta phải chạy phần mềm Matlab sẽ xuất hiện cửa sổ chính của chương trình Matlab như hình 4.1. Sau đó tại dòng nhắc lệnh ta gõ trực tiếp câu lệnh Simulink hoặc kích vào biểu tượng Simulink . Cửa sổ của chương trình Simulink như hình 4.1

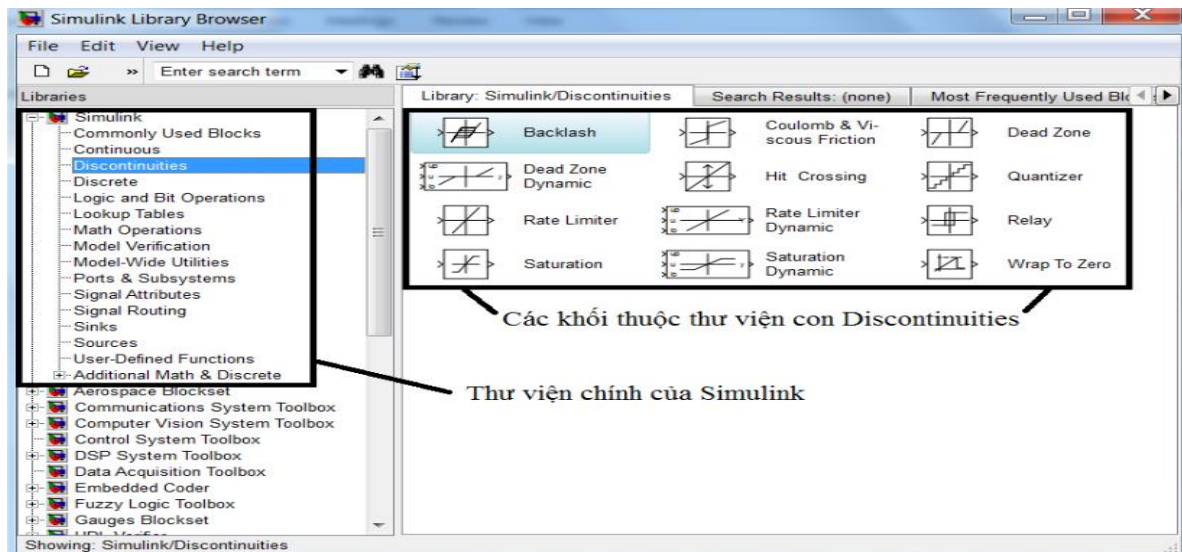


Hình 4.1: Cửa sổ chính của chương trình Matlab

Thư viện Simulink chứa các phần sau:

Commonly Used Blocks - các khối thường được sử dụng;

Continuous - các khối tuyến tính;



Hình 4.2: Cửa sổ của chương trình Simulink

Discontinuous - các khối phi tuyến;

Discrete - các khối rời rạc;

Logic and Bit Operation - các khối logic và phép toán Bit;

Lookup Tables - bảng tra cứu tương thích;

Math Operation - các hàm toán học;

Model Verification - các khối kiểm tra mô hình;

Model - Wide Utilities - các bảng;

Ports & Subsystems - các cổng và hệ thống con;

Signal Attributes - các tham số tín hiệu;

Signal Routing - định tuyến tính;


Sinks - các khối hiển thị;

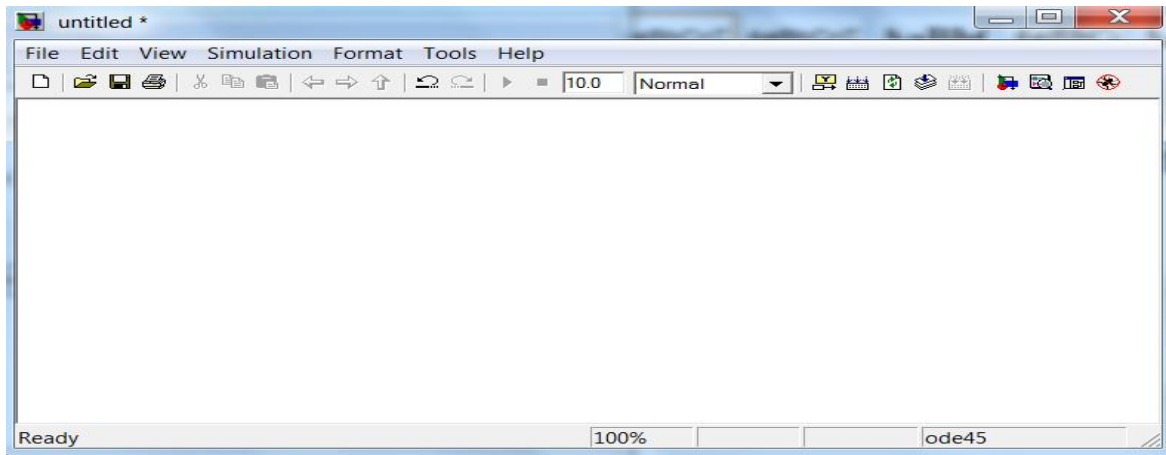
Sources - các nguồn tín hiệu;

User - Defined Functions - các hàm chức năng do người sử dụng định nghĩa;

Additional Math & Discrete - các khối rời rạc và toán học bổ sung

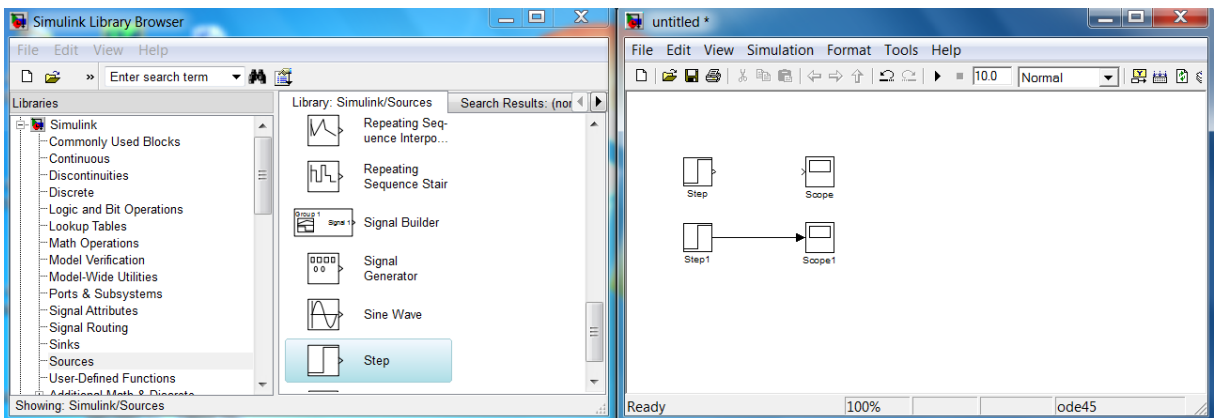
4.1.2 Cách tạo mô hình trong Simulink

Tạo file mới của mô hình bằng đường dẫn File/New/Model hoặc kích vào biểu tượng  trên thanh công cụ, cửa sổ mới được tạo ra như hình 4.3.



Hình 4.3: Cửa sổ trống tạo mô hình

Để tạo mô hình ta cần có các khối chức năng tương ứng với mô hình. Ta cần khối chức năng nào thì mở khối chức năng đó ra, kích và giữ chuột trái sau đó kéo sang cửa sổ trống tạo mô hình rồi thả chuột trái ra. Để thực hiện việc này ta phải cho hiện đồng thời cửa sổ trống tạo mô hình và cửa sổ thư viện Simulink như hình 4.4.



Hình 4.4: Cho hiện đồng thời 2 cửa sổ để thực hiện việc kéo khối từ cửa sổ thư viện Simulink sang cửa sổ tạo mô hình

Để nối các khối với nhau ta kích và giữ chuột trái đồng thời kéo từ đầu ra của khối này tới đầu vào của khối kia. Để xóa khối, ta kích vào khối cần xóa rồi ấn phím Delete.

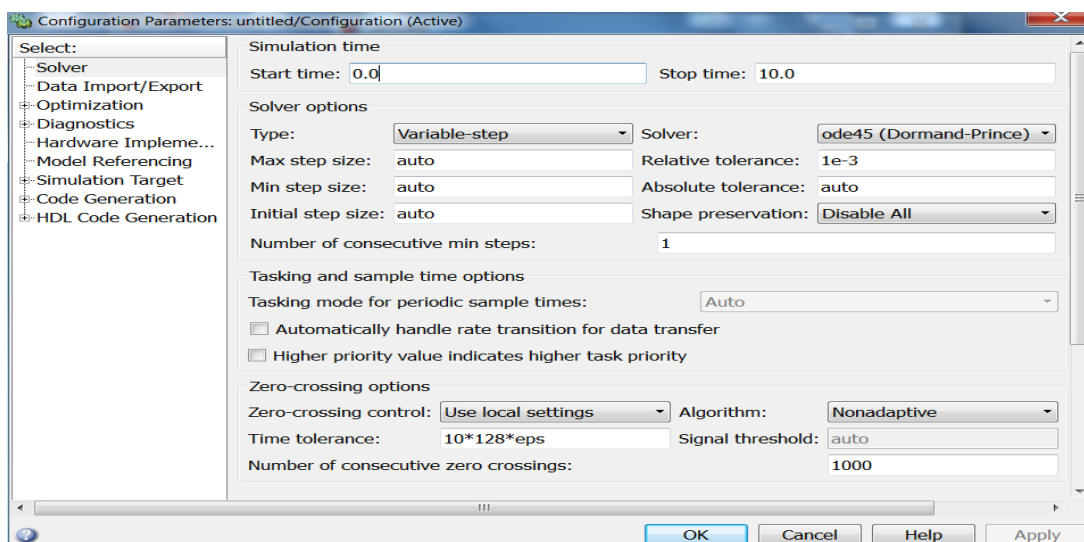
Để mở cửa sổ thiết lập tham số cho khối ta kích đúp chuột trái vào khối cần thiết lập tham số.

Cách mở file mô hình có sẵn hoặc lưu file ta làm tương tự như Microsoft Office Word.

4.1.3. Thiết lập các tham số mô phỏng

Trước khi thực hiện các tính toán cần phải thiết lập các tham số cấu hình mô phỏng.

Ta mở bảng thiết lập các tham số cấu hình mô phỏng như sau: Simulation/Configuration Parametes, bảng điều khiển được thực hiện như hình 4.5.



Hình 4.5: Thiết lập các tham số cấu hình mô phỏng

Bảng này chia làm 2 nhóm chính:

- * *Simulation time*: Khoảng thời gian mô phỏng

- * *Solver option*: Các tùy chọn bộ giải

Thời gian tính toán được đưa ra bằng cách xác định giá trị thời gian tính toán ban đầu (*Start time*) và thời gian kết thúc (*Stop time*). Thời gian bắt đầu thường được thiết lập bằng số không, giá trị của thời gian kết thúc do người sử dụng đưa ra dựa trên các điều kiện của bài toán cần giải quyết.

Khi lựa chọn các thông số cấu hình cần chỉ ra cách thức mô phỏng (*Type*) và phương pháp tính toán trạng thái mới của hệ thống (*Solver*). Đối với các tham số *Type* cho phép hai tùy chọn – với bước cố định (*Fixed-step*) hoặc bước thay đổi (*Variable-step*). Theo mặc định, *Variable-step* được sử dụng để mô phỏng các hệ thống liên tục, *Fixed-step* cho hệ thống rời rạc.

Danh sách các phương pháp phân tích của hệ thống (*Solver*) có một vài lựa chọn, những phương pháp này là khác nhau đối với bước thời gian thay đổi (*Variable*) và bước thời gian cố định (*Fixed*). Các bước tính mô phỏng hệ thống mặc định tự động, giá trị yêu cầu có thể được nhập thay vì giá trị tự động. Khi chọn *Variable-step* xuất hiện bộ ba thông số:

- *Max step size* – bước tính cực đại. Theo mặc định, nó được cài đặt tự động và giá trị của nó trong trường hợp này là $(\text{StopTime} - \text{StartTime})/50$. Thông thường giá trị này quá lớn, và đồ thị quan sát dạng đường gấp khúc (không mịn). Trong trường hợp này độ lớn các bước tối đa để tính toán phải được quy định rõ ràng.

- *Min step size* – bước tính cực tiểu.
- *Initial step size* – giá trị ban đầu của bước mô phỏng.

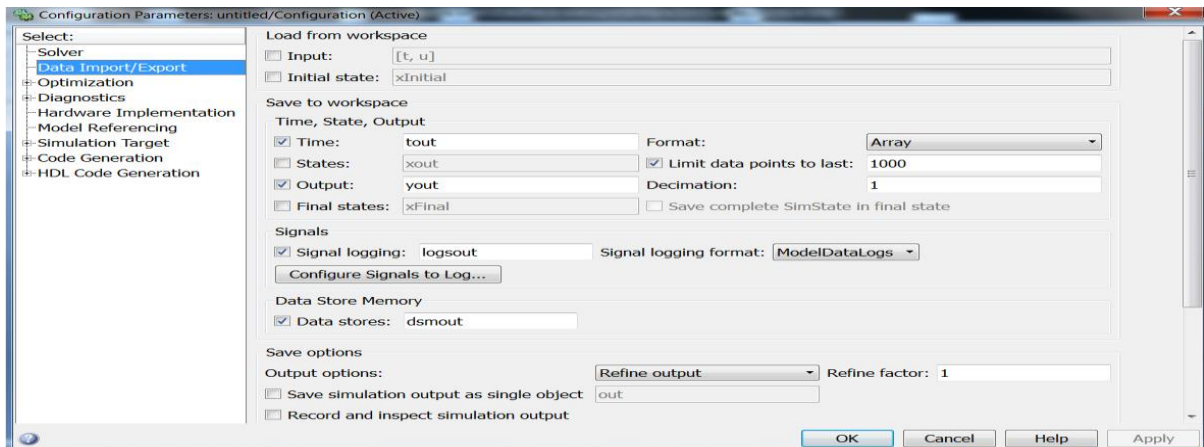
Trong mô hình hóa các hệ thống liên tục bằng cách sử dụng một biến số cần chỉ ra tính chính xác của phép tính: tương đối (Relative tolerance) và tuyệt đối (Absolute tolerance). Theo mặc định chúng tương ứng với 10^{-3} và tự động.

4.1.4. Thiết lập các tham số thay đổi theo vùng làm việc

Các phần tử cho phép điều khiển đầu vào và đầu ra dữ liệu trung gian và kết quả mô phỏng trong không gian làm việc *MATLAB* được đặt trên tab *Data Import/Export* như hình 4.6.

Load from workspace (Tải về từ vùng làm việc). Nếu các ô Input (các dữ liệu đầu vào) được lựa chọn, khi đó trong khối văn bản bên phải có thể định dạng dữ liệu được đọc từ không gian làm việc *MATLAB*.

Thiết lập *Initial State* (trạng thái ban đầu) cho phép đưa tên biến vào trường liên kết dạng văn bản với nó có chứa các thông số trạng thái ban đầu của mô hình. Dữ liệu đưa ra trong trường hợp *Input* và *Initial State* được chuyển vào một mô hình thực thi bằng cách sử dụng một hoặc nhiều khối *In* (từ nguồn thư viện *Sources*).



Hình 4.6: Tab *Workspace I/O* khối thoại để thiết lập các thông số mô hình

Save to workspace (Lưu vào không gian làm việc) – Cho phép thiết lập chế độ đầu ra của các giá trị tín hiệu trong không gian làm việc *MATLAB* và đặt tên cho chúng.

Save options (Các tùy chọn ghi lưu) – Chỉ định số dòng khu truyền biến vào vùng làm việc. Nếu các dòng *Limit rows to last* được thiết lập, sau đó đầu vào có thể chỉ định số lượng đường truyền (đếm số dòng được xác định sau khi hoàn thành tính toán). Nếu *Limit rows to last* không được thiết lập thì tất cả các dữ liệu sẽ được chuyển. Thông số *Decimation* tạo bước ghi biến vào vùng làm việc. Tham số *Format* (định dạng dữ liệu) quy định định dạng các biến trong không gian làm việc của dữ liệu. Định dạng *Array*



(Mảng), *Structure* (cấu trúc), *Structure With Time* (cấu trúc với một trường bổ sung – “thời gian”).

Output options (các tùy đầu ra) cài đặt tham số xuất tín hiệu ra của hệ thống mô phỏng. Có thể chọn một trong ba phương án:

Refine output (làm mịn đầu ra) – Lựa chọn này nhằm cung cấp thêm các điểm đầu ra phụ khi đầu ra mô phỏng quá thô. Tham số này xác định một số nguyên dương các điểm đầu ra giữa các bước thời gian mô phỏng. Ví dụ, nếu hệ số làm mịn bằng 2 sẽ xuất hiện tín hiệu ra tại các bước mô phỏng và cả ở giữa chúng. Hệ số làm mịn mặc định bằng 1.

Produce additional output (tạo đầu ra bổ sung) – Lựa chọn tạo đầu ra bổ sung cho phép ta xác định thêm những thời điểm mà tại đó bộ giải (solver) cần xuất tín hiệu ra. Khi chọn tùy chọn này một trường *Output times* sẽ xuất hiện trên *panel Data Import/Export*. Hãy nhập vào trường này một biểu thức *MATLAB* cho phép tính được thời điểm bổ sung hoặc một vectơ các thời điểm bổ sung. Tùy chọn này buộc bộ giải phải tạo ra các thời điểm phù hợp với các thời điểm bổ sung đã xác định. Ví dụ, nếu thời điểm bắt đầu mô phỏng được đặt bằng 0, thời điểm kết thúc đặt bằng 60, và nhập véc tơ [10:10:50] vào trường *Output Times*, thì kết quả là đầu ra mô phỏng sẽ có ở các thời điểm sau: 0, 10, 20, 30, 40, 50, 60.

4.1.5. Thực hiện mô phỏng

Bắt đầu tính toán được thực hiện bằng cách chọn mục menu *Simulation/Start*, hoặc công cụ  trên thanh công cụ. Quá trình tính toán có thể được hoàn thành trước thời hạn bằng cách chọn mục menu *Simulation/Stop* hoặc công cụ . Việc tính toán cũng có thể tạm dừng (*Simulation/Pause*) và sau đó tiếp tục (*Simulation/Continue*).

4.1.6. Giới thiệu về *simpower system blocksets*

Simpower system là một trong những thư viện đặc biệt nhằm mô phỏng thiết bị cụ thể. *Simpower system* bao gồm các khối dùng để mô phỏng thiết bị kỹ thuật điện như mô hình các nguồn điện, động cơ, máy biến áp, đường dây truyền tải,..... và có phần riêng để mô phỏng thiết bị điện tử công suất và hệ thống điều khiển chúng.

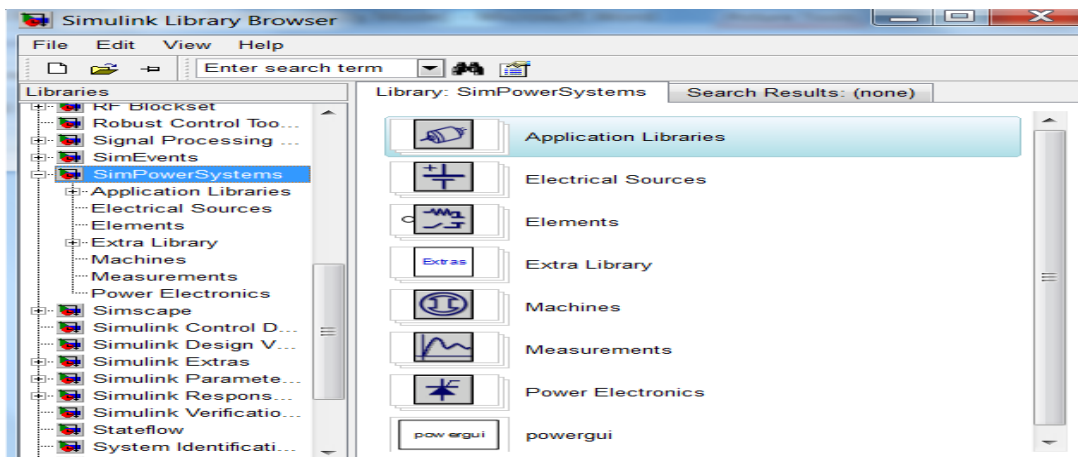
Sử dụng các khả năng ứng dụng của *Simulink* và *Simpower system* mô phỏng không chỉ quá trình làm việc của thiết bị mà còn phân tích các dạng khác nhau của thiết bị đó. Ngoài ra ta còn có thể tính toán chế độ ổn định của hệ thống xoay chiều, tính điện trở của từng đoạn mạch, quan sát dạng sóng, xác định đặc tính tần số, phân tích ổn định của dòng điện và điện áp.

Mạch động lực của bộ biến đổi công suất được mô phỏng bằng các khối trong *Simpower system* còn mạch điều khiển thì có thể sử dụng các khối của *Simulink*. Điều

đó giúp cho sơ đồ đơn giản đi rất nhiều. Hơn nữa, có thể kết hợp sử dụng các thư viện của MATLAB tạo điều kiện thuận lợi trong việc mô phỏng hệ thống điện. Hiện nay Simpower system được coi là một trong các thư viện tốt nhất để mô phỏng thiết bị và hệ thống điện.

Thư viện Simpowersystem bao gồm:

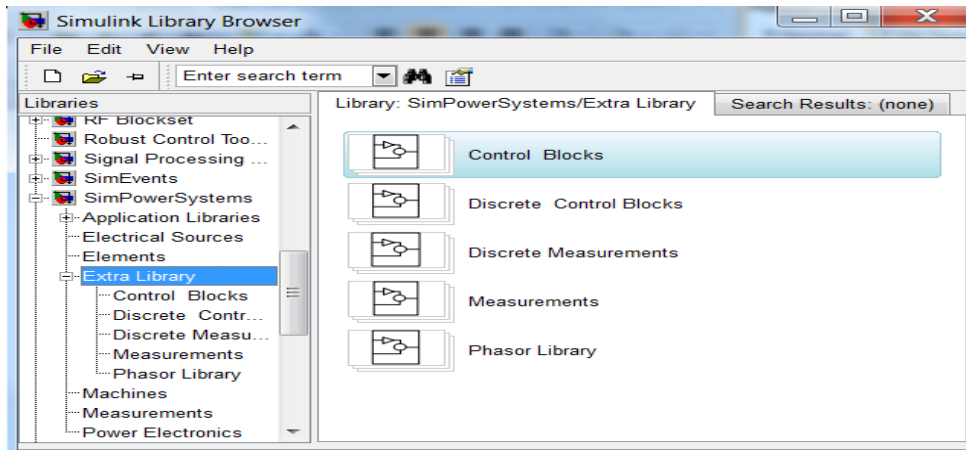
- Extras Library - Thư viện mở rộng
- Application Libraries – Thư viện ứng dụng
- Electrical Sources – Các nguồn điện
- Machines – Máy điện
- Elements – Các thiết bị điện
- Measurements – Thiết bị đo lường
- Power Electronics – Các linh kiện điện tử công suất
- Powergui – Chế độ phân tích



Hình 4.7. Thư viện Simpowersystem

4.1.6.1. Thư viện mở rộng -Extras Library

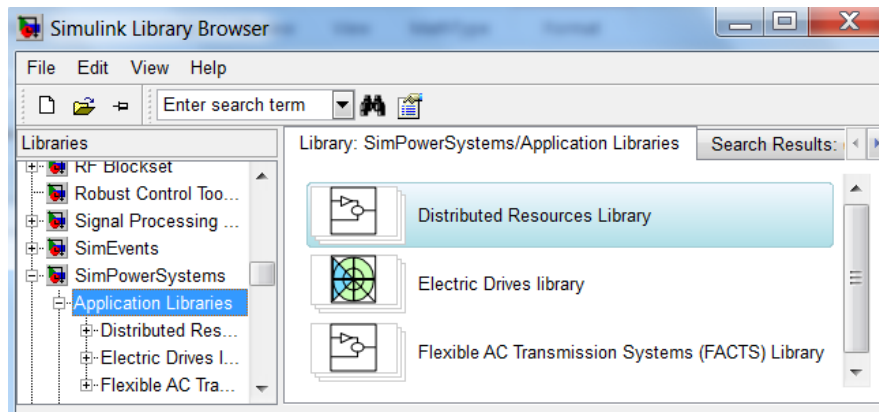
Thư viện mở rộng -Extras Library bao gồm các khối đo lường tương tự, đo lường rời rạc, khối điều khiển tương tự, khối điều khiển rời rạc, khối phân tích theo pha.



Hình 4.8. Thư viện mở rộng

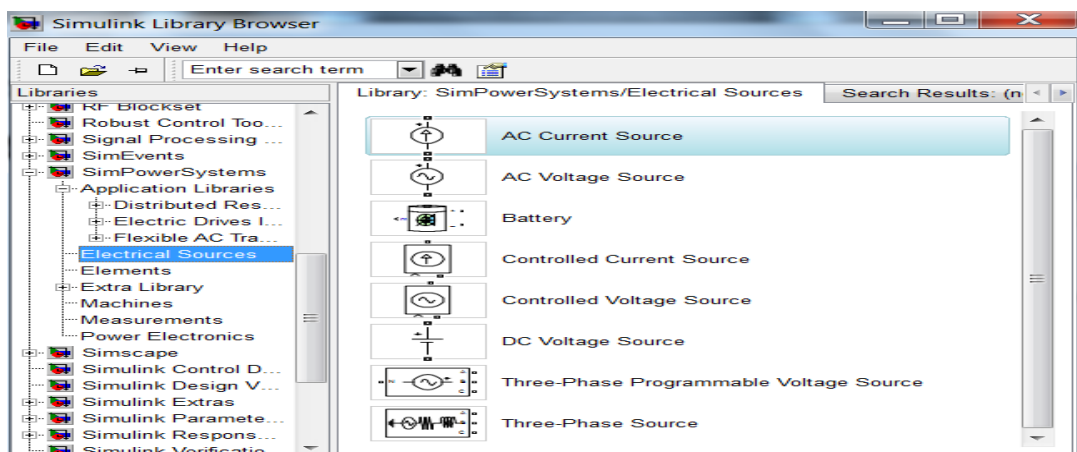
4.1.6.2. Thư viện các khối ứng dụng - Application Libraries

Thư viện các khối ứng dụng bao gồm các thư viện các hệ truyền động phức tạp, hệ thống truyền tải, hệ thống năng lượng gió.



Hình 4.9. Thư viện các khối ứng dụng

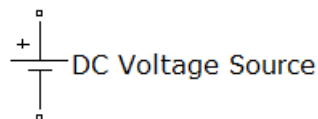
4.1.6.3. Khối nguồn điện -Electrical Source



Hình 4.10. Khối nguồn điện Electrical Source

a. DC Voltage Source - Nguồn điện áp một chiều lý tưởng

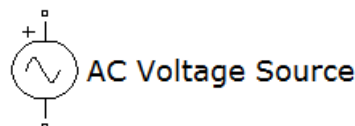
Mô tả



Nguồn điện áp một chiều lý tưởng tạo nguồn điện áp một chiều lý tưởng với điện áp không đổi với điện trở trong bằng 0.

b. AC Voltage Source - Nguồn điện áp xoay chiều lý tưởng

Mô tả



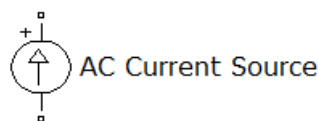
AC Voltage Source tạo nguồn điện áp xoay chiều lý tưởng theo phương trình sau:

$$U = A \cdot \sin(\omega t + \phi)$$

Cho phép nhập điện áp có giá trị âm. Trường hợp tần bằng 0 thì nguồn AC trở thành nguồn DC. Tần số không thể nhập giá trị âm, trong trường hợp này chương trình sẽ báo lỗi.

c. AC Current Source - Nguồn dòng xoay chiều lý tưởng

Mô tả



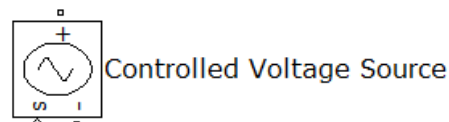
AC Current Source tạo nguồn dòng điện xoay chiều lý tưởng theo phương trình sau:

$$I = A \cdot \sin(\omega t + \phi) \quad (7.1)$$

Cho phép nhập dòng điện có giá trị âm. Trường hợp tần bằng 0 thì nguồn dòng AC trở thành nguồn DC. Tần số không thể nhập giá trị âm, trong trường hợp này chương trình sẽ báo lỗi.

d. Controlled Voltage Source - Nguồn áp điều khiển

Mô tả

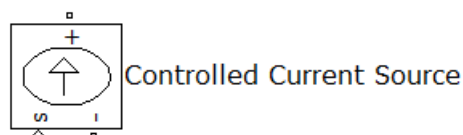


Controlled Voltage Source tạo nguồn áp điều khiển được nhờ tín hiệu Simulink.

Khởi Controlled Voltage Source có thể tạo điện áp là AC hoặc DC. Nếu muốn khởi động mô hình, đầu vào phải nối với tín hiệu khởi động Sin hay DC tương ứng với giá trị ban đầu.

e. Controlled Current Source - Nguồn dòng điều khiển

Mô tả

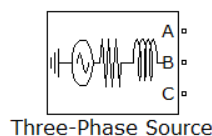


Controlled Current Source: tạo nguồn dòng điện điều khiển được nhờ tín hiệu Simulink.

Khối Controlled Current Source có thể tạo dòng điện điều khiển là AC hoặc DC. Nếu muốn khởi động mô hình, đầu vào phải nối với tín hiệu khởi động Sin hay DC tương ứng với giá trị ban đầu.

f. Three-Phase Source - Nguồn xoay chiều 3 pha

Mô tả

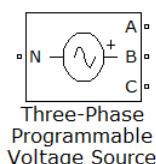


Khối Three-Phase Source tạo nguồn điện áp xoay chiều 3 pha có điện trở trong là R-L.

Khối Three Phase Source tạo nguồn điện áp xoay chiều ba pha cân bằng với điện trở trong R-L. Nguồn được nối hình sao Y với điểm trung tính, điểm này có thể nối đất hoặc không. Ta có thể cài đặt giá trị điện trở trong R-L của nguồn bằng cách nhập tỷ số điện kháng và điện trở ngắn mạch X/R.

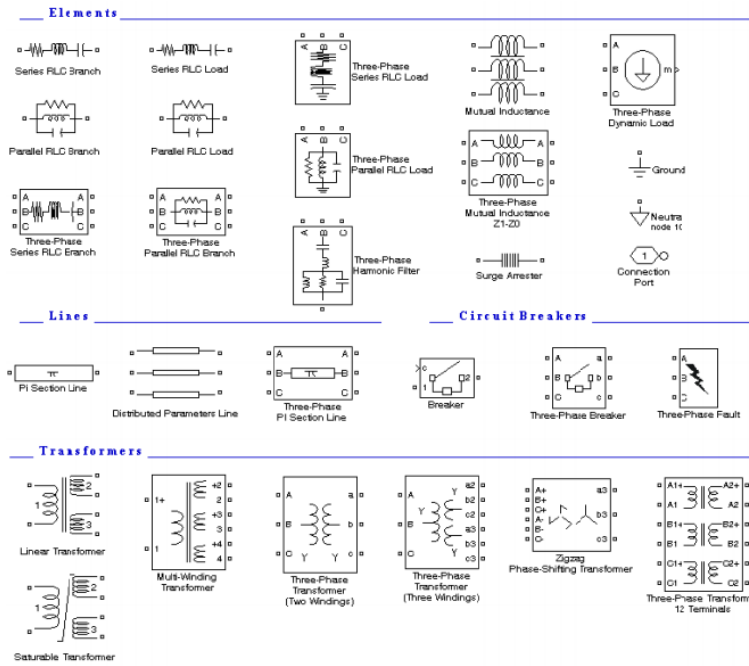
g. Three-Phase programmable Voltage Source - Nguồn áp xoay chiều 3 pha lập trình

Mô tả



Khối Three-Phase Programmable Voltage Source được sử dụng để tạo điện áp 3 pha với các thông số thay đổi theo thời gian như biên độ, trễ pha, tần số, và thành phần hài. Ngoài ra 2 thành phần hài có thể cộng vào thành phần hài cơ bản theo nguyên lý xếp chồng.

4.1.6.4. Khối thiết bị điện – Elements



Hình 4.11. Thư viện các thiết bị điện

a. Khối tiếp đất – Ground

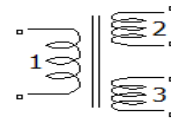
Mô tả



Khối Ground dùng để nối thiết bị hay hệ thống xuống đất.

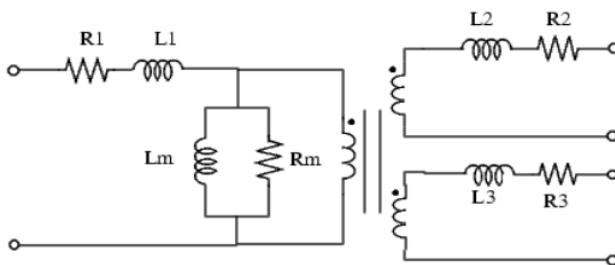
b. Máy biến áp tuyến tính - Linear Transformer

Mô tả



Linear Transformer

Sơ đồ thay thế của mô hình máy biến áp tuyến tính như hình 4.12



Hình 4.12. Sơ đồ thay thế của mô hình máy biến áp tuyến tính

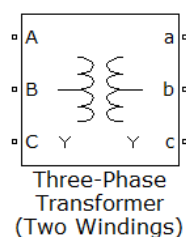
Trong mô hình này điện trở các cuộn dây là (R1, R2, R3) và cảm kháng của chúng là (L1, L2, L3). Từ tính của lõi thép được mô hình hóa bằng (Rm Lm) tuyến tính.

Mô hình hóa máy biến áp lý tưởng

Để tạo máy biến áp lý tưởng ta phải cho điện trở và điện kháng cuộn dây bằng 0 và điện trở, cảm kháng mạch từ hóa bằng vô cùng lớn.

c. Máy biến áp ba pha hai cuộn dây - Three-Phase Transformer (Two Windings)

Mô tả



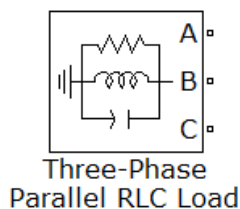
Khối máy biến áp ba pha hai cuộn dây sử dụng ba máy biến áp một pha.

Tổ đấu dây của các cuộn dây:

- Y - sao
- Y – sao nối trung tính.
- Grounded Y — sao nối đất
- Delta (D1), delta lagging Y by 30 degrees – tam giác chậm với sao 30 độ
- Delta (D11), delta leading Y by 30 degrees – tam giác nhanh với sao 30 độ

d. Tải công suất 3 pha RLC song song (Three-Phase Parallel RLC Load)

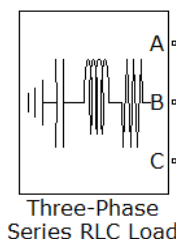
Mô tả



Tải công suất ba pha có điện trở không đổi.

e. Tải công suất 3 pha RLC nối tiếp Three-Phase Series RLC Load

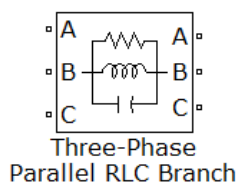
Mô tả



Tải công suất ba pha có điện trở không đổi.

f. Tải 3 pha RLC song song Three-Phase Parallel RLC Branch

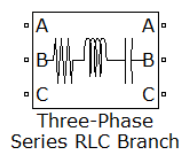
Mô tả



Là tải RLC ba pha cân bằng, mỗi pha có điện trở, cuộn kháng và tụ điện được kết nối song song. Để loại trừ các phần tử thì nhập các giá trị sau cho chúng, tương ứng R, L là (inf), còn C là (0). Có thể nhập giá trị âm cho điện trở, cảm kháng và điện dung.

g. Tải 3 pha RLC nối tiếp Three-Phase Series RLC Branch

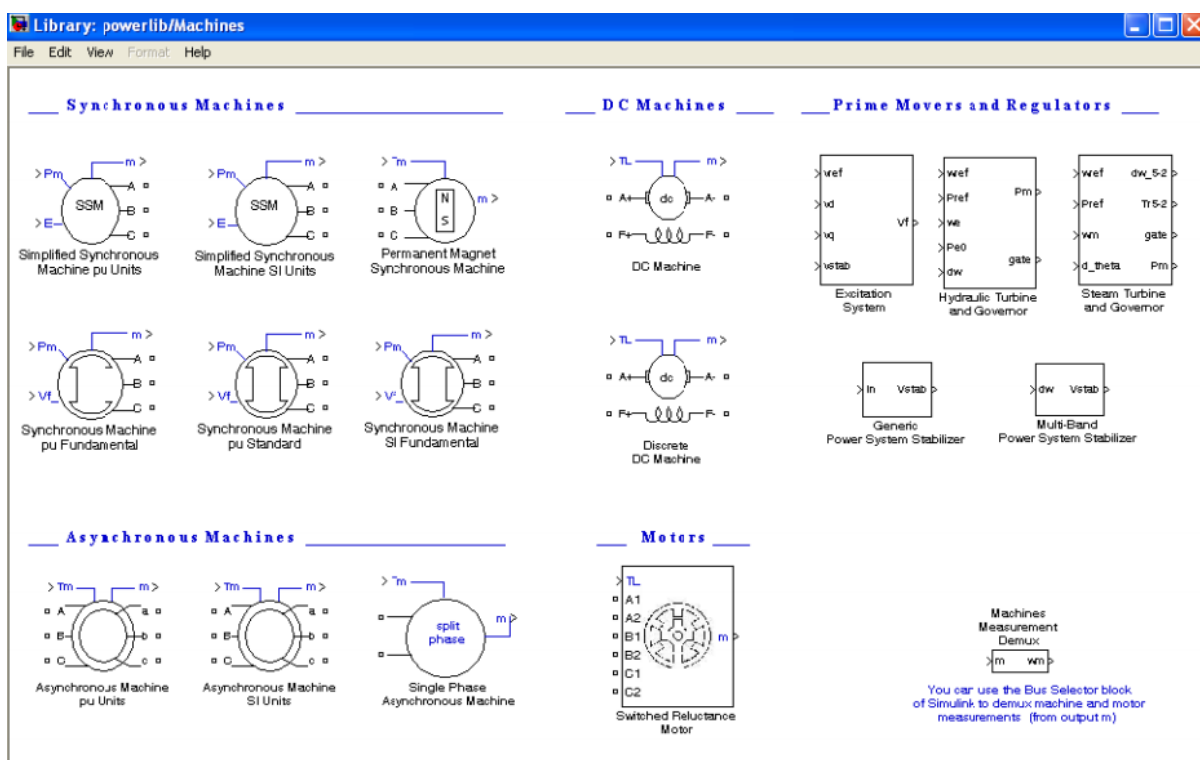
Mô tả



Là tải RLC ba pha cân bằng, mỗi pha có điện trở, cuộn kháng và tụ điện được kết nối nối tiếp. Để loại trừ các phần tử thì nhập các giá trị sau cho chúng tương ứng R, L là (inf), còn C là (0).

Có thể nhập giá trị âm cho điện trở, cảm kháng và điện dung.

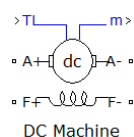
4.1.6.5. Khởi máy điện – Machines



Hình 4.13. Thư viện của Machines (máy điện)

a. Động DC Machine

Mô tả



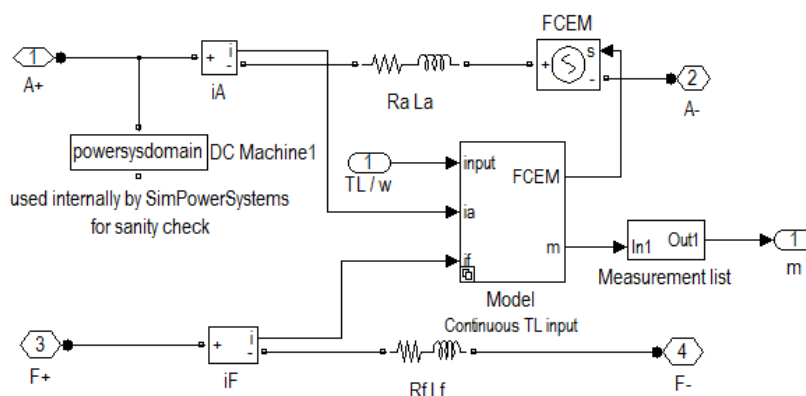
Khối DC machine dùng để mô phỏng động cơ điện một chiều có cuộn dây kích từ độc lập, song song hoặc nối.

Khối có 5 đầu vào (F+, F-, A+, A-, TL) và một đầu ra (m). Các đầu vào A+, A- là các cực của cuộn phần ứng (tương ứng với dấu cộng và dấu trừ). Các đầu vào F+, F- là các cực của cuộn dây kích từ.

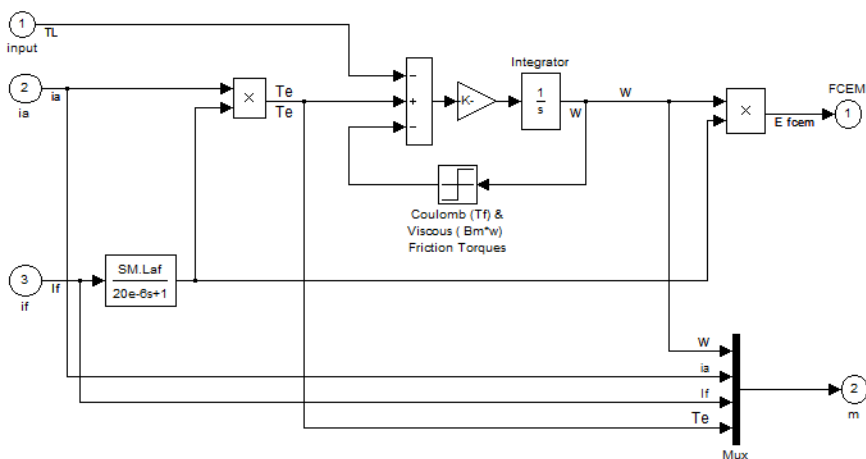
Đầu vào TL là đầu vào đặt độ lớn momen tải, khi TL >0 mô phỏng ở chế độ động cơ, khi TL <0 mô phỏng ở chế độ máy phát.

Đầu ra m là vecto gồm 4 thành phần được định dạng mà cụ thể là:

- w: tần số góc quay của trục động cơ;
- Ia: dòng điện phần ứng;
- If: dòng của cuộn dây kích từ;
- Te: mômen điện từ trên trục.



Hình 4.14. Sơ đồ bên trong khối DC Machine



Hình 4.15. Sơ đồ bên trong khối Model Continuous TL input

Hộp thoại và thông số

Armature resistance and inductance [Ra(ohms) La(H)]: nhập dạng vectơ điện trở phần ứng đo bằng Ω và điện cảm đo bằng H.

Field resistance and inductance [Rf(ohms) Lf(H)]: nhập hệ số điện trở và điện cảm của cuộn dây kích từ tương ứng.

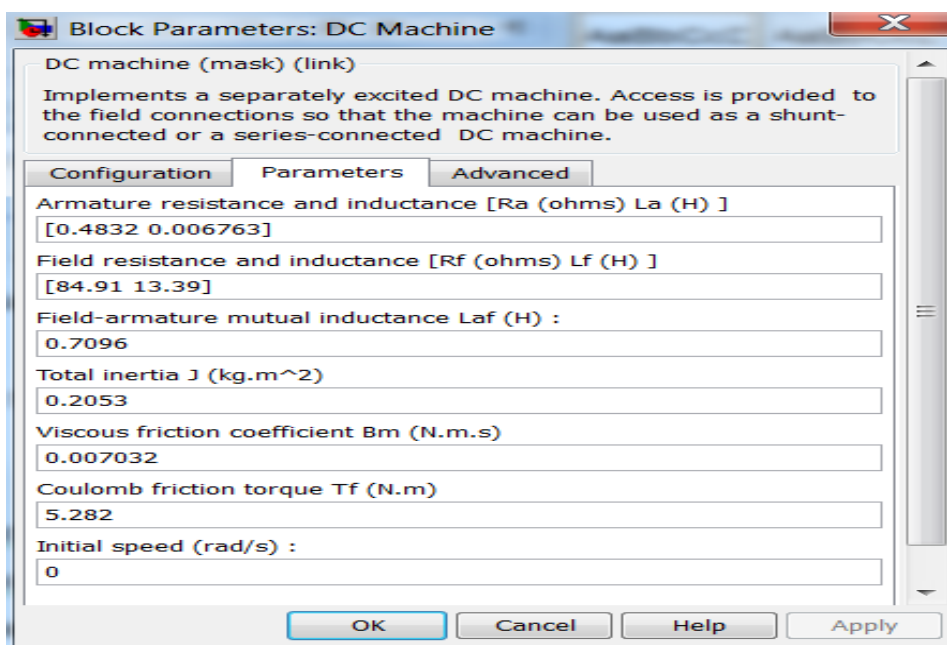
Field armature mutual inductance Laf (H): nhập hệ số điện cảm tương hỗ giữa các cuộn dây kích từ và phần ứng đo bằng H.

*Total inertia J (kg*m²):* Nhập mômen quán tính toàn phần được quy đổi về phần ứng.

*Viscous friction coefficient Bm (N*m*s):* nhập hệ số ma sát nhớt;

*Coulomb friction torque Tf (N*m):* nhập mômen ma sát khô;

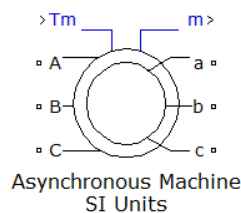
Initial speed (rad/s): Nhập vận tốc ban đầu của trục, nó cần thiết để bắt đầu tính toán và có thể có giá trị rất nhỏ. Theo mặc định giá trị ban đầu là 1 rad/s.



Hình 4.16. Hộp thoại và thông số của khối DC Machine

b. Động cơ không đồng bộ - Asynchronous Machine

Mô tả

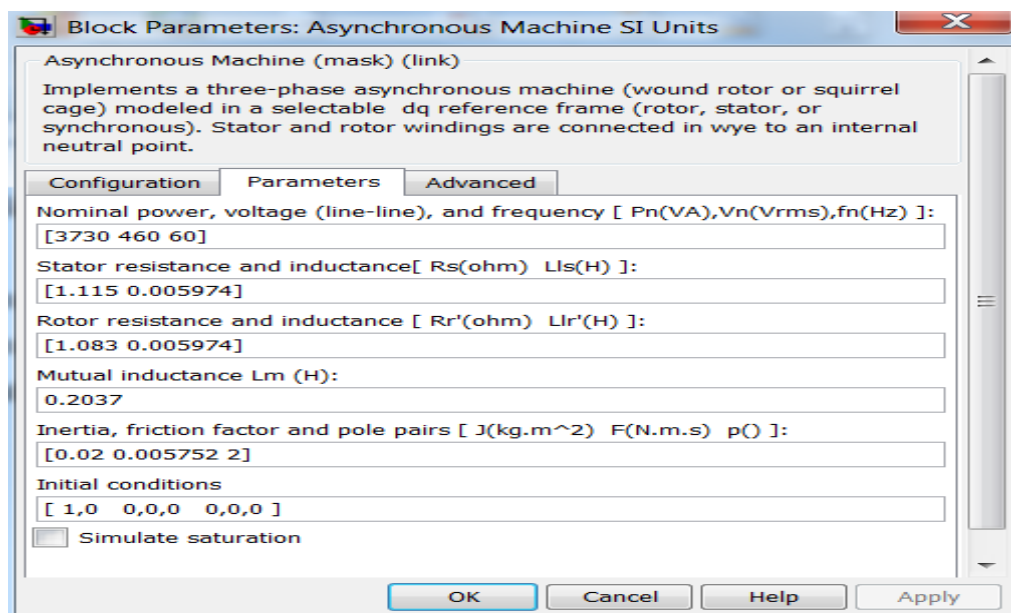


Khối Asynchronous Machine có các cuộn dây stator và rôto được đấu hình sao. Có thể mô phỏng các động cơ với rôto lồng sóc hoặc rôto dây quấn.

Khối động cơ không đồng bộ có thể vận hành ở hai chế độ, phụ thuộc vào dấu của mô men cơ T_m :

- Nếu T_m mang dấu dương động cơ ở chế độ động cơ.
- Nếu T_m mang dấu âm động cơ ở chế độ máy phát.

Hộp thoại và thông số



Hình 4.17. Hộp thoại và thông số của khối Asynchronous Machine

Preset model: Thông số động cơ đã được cài đặt trước như công suất định mức (HP), điện áp dây (V), tần số định mức (Hz), tốc độ định mức (vòng/phút)

Mechanical input – ngõ vào cơ khí: Cho phép lựa chọn những dạng ngõ vào khác nhau. Chọn Torque T_m để chỉ định mômen là đầu vào, tốc độ của động cơ được xác định bằng mômen quán tính J và hiệu số giữa mômen tải và mômen do động cơ sinh ra. Dấu quy đổi: nếu tốc độ mang dấu dương, mômen dương thể hiện động cơ ở chế độ động cơ, mômen âm thể hiện chế độ máy phát. Lựa chọn Speed w để xác định ngõ vào là tốc độ (rad/s). Tốc độ động cơ được cải thiện và mômen quán tính có thể bỏ qua.

Show detailed parameters – nếu chọn cửa sổ sẽ hiện ra thông số chi tiết của động cơ.

Rotor type – dạng rô to;

Reference frame: xác định cấu trúc biến đổi điện áp ngõ vào sang cấu trúc dq và dòng điện ngõ ra (dq) sang dạng (abc);

- Rotor (Park transformation)
- Stationnary (Clarke or $\alpha\beta$ transformation)
- Synchronous

Nominal power, L-L volt, and freq – công suất định mức Pn(VA), trị hiệu dụng điện áp dây – dây Vn (V), tần số định mức fn(Hz).

Stator- điện trở và cảm kháng cuộn stator;

Rotor - điện trở và cảm kháng cuộn rô to;

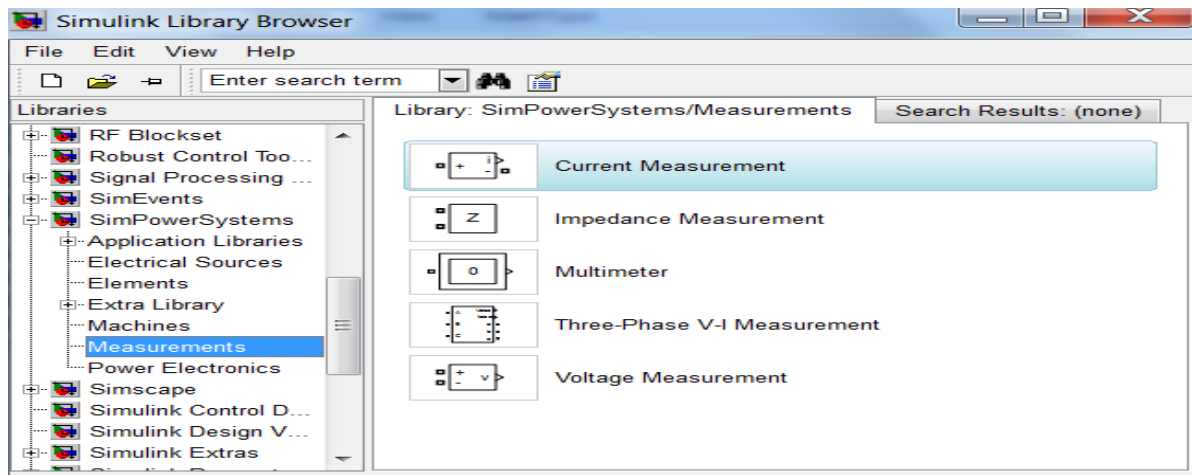
Mutual inductance – cảm kháng mạch từ;

Inertia, friction factor, and pairs of poles: mô men quán tính J (kg.m²), hệ số ma sát F (N.m.s), số đôi cực p. Mô men ma sát (Tf = F.w).

Initial conditions – trạng thái ban đầu: Độ trượt, góc lệch điện, dòng stator, góc pha [slip, th, ias, ibs, ics, phaseas, phasebs, phasecs].

Đối với rô to dây quấn: [slip, th, ias, ibs, ics, phaseas, phasebs, phasecs, iar, ibr, icr, phasear, phasebr, phasecr].

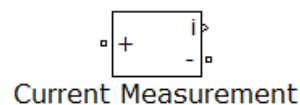
4.1.6.6. Khối đo lường (Measurements s Library)



Hình 4.18. Thư viện khối đo lường

a. Current Measurement – thiết bị đo dòng điện

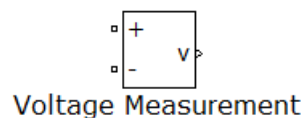
Mô tả



Khối Current Measurement dùng để đo dòng điện tức thời trong một thiết bị hoặc dây dẫn. Ngõ ra của nó là tín hiệu Simulink, vì vậy có thể sử dụng cho bất cứ khối Simulink nào.

b. Voltage Measurement – thiết bị đo điện áp

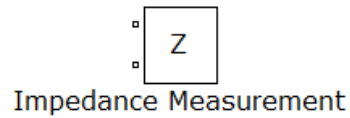
Mô tả



Khối Voltage Measurement dùng để đo điện áp tức thời giữa hai điểm, hai nút của sơ đồ. Ngõ ra của nó là tín hiệu Simulink, vì vậy có thể sử dụng cho bất cứ khối Simulink nào.

c. Impedance Measurement – Thiết bị đo điện trở

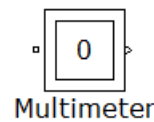
Mô tả



Khối Impedance Measurement dùng để đo tổng trở giữa hai nút trong mạch điện phụ thuộc vào tần số. Bao gồm một nguồn dòng I_z nối giữa đầu vào và đầu thứ hai với khối Impedance Measurement và điện áp đo được V_z giữa hai đầu của nguồn dòng.

d. Multimeter – Đồng hồ vạn năng

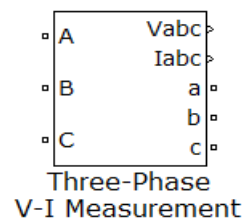
Mô tả



Khối Multimeter dùng để đo điện áp, dòng điện trong các khối của SimPowerSystems

e. Three-Phase V-I Measurement – Thiết bị đo điện áp, dòng điện 3 pha

Mô tả

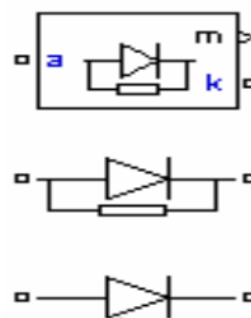


Khối Three-Phase V-I Measurement dùng để đo điện áp, dòng điện trong mạch 3 pha. Khối Three-Phase V-I Measurement được gắn nối tiếp với các phần tử 3 pha, hiển thị điện áp pha và điện áp dây và dòng điện.

4.1.6.7. Khối thư viện điện tử công suất - Power Electronics

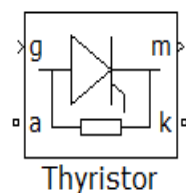
a. Diode

Mô tả



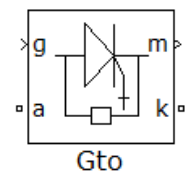
b. Thyristor

Mô tả



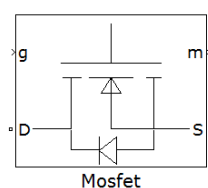
c Tiristor GTO

Mô tả



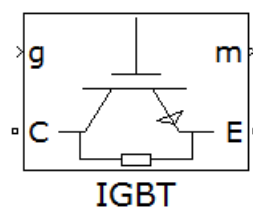
d. MOSFET

Mô tả

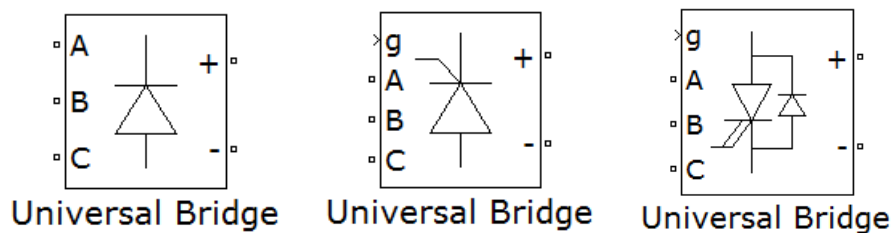


e. IGBT

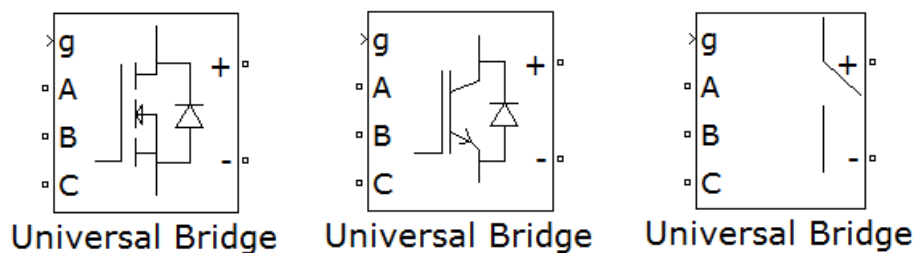
Mô tả



f. Các cầu thông dụng - Universal Bridge



Mô tả



Các khối cầu thông dụng thực hiện một bộ biến đổi công suất 3 pha thông dụng, bao gồm 6 van công suất nối lại với nhau thành một cấu trúc cầu. Loại van công suất và cấu trúc bộ biến đổi được lựa chọn từ hộp hội thoại.

Khối cầu thông dụng cho phép mô phỏng các bộ biến đổi sử dụng cả hai loại thiết bị khóa tự nhiên (diodes hoặc thyristors) và cả thiết bị khóa cưỡng bức (GTO, IGBT, MOSFET).

Các khối cầu thông dụng là các khối cơ bản dành cho việc tạo ra các bộ biến đổi nguồn áp 2 cấp (VSC).

Bài tập cuối bài :

Bài 1: Thực hiện lấy khối nguồn 1 chiều từ thư viện Simulink thay đổi tham số và hiển thị lên khối Scope.

Bài 2: Thực hiện lấy khối nguồn xoay chiều 1 pha từ thư viện Simulink thay đổi tham số và hiển thị lên khối Scope.

Bài 3: Thực hiện lấy khối nguồn xoay chiều 3 pha từ thư viện Simulink thay đổi tham số và hiển thị lên khối Scope.

Bài 4: Mô phỏng hệ thống có hàm truyền đạt

$$W(s) = \frac{10s + 3}{s^2 + 3s + 1}$$

Tín hiệu kích thích là hàm bước nhảy step.

Bài 5: Mô phỏng hệ thống có hàm truyền đạt

$$W(s) = \frac{10s + 3}{s^2 + 3s + 1}$$

Tín hiệu kích thích có dạng Constant.

Câu hỏi trắc nghiệm

Câu 1: Thư viện nào trong Simulink chứa các khối nguồn tín hiệu ?

- A. Model Verification B. Signal Routing C. Sources D. Lookup Tables

Câu 2: Trong hộp thoại và thông số của khối Động DC Machine thì tham số Field armature mutual inductance Laf (H) có ý nghĩa gì?

- A. Nhập dạng vectơ điện trở phản ứng đo bằng Ω và điện cảm đo bằng H.
B. Nhập hệ số điện trở và điện cảm của cuộn dây kích từ tương ứng.
C. Nhập hệ số điện cảm tương hỗ giữa các cuộn dây kích từ và phản ứng đo bằng H.
D. Nhập giá trị điện cảm của cuộn dây kích từ đo bằng H.

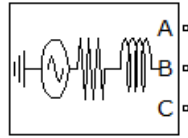
Câu 3: Vai trò của khối Current Measurement có vai trò:

- A. Thiết bị đo dòng điện
B. Thiết bị đo điện áp
C. Thiết bị đo điện trở
D. Đồng hồ vạn năng

Câu 4: Thư mục không nằm trong thư viện Simpower system.

- A. Extras Library
- B. Application Libraries
- C. Signal Attributes
- D. Power Electronics

Câu 5: Khối sau có vai trò gì?



Three-Phase Source

- A. Nguồn xoay chiều 3 pha
- B. Nguồn áp xoay chiều 3 pha lập trình
- C. Nguồn dòng điều khiển
- D. Nguồn dòng 3 pha điều khiển

Bài 5: Mô phỏng hệ thống với matlab/simulink (tiếp) (số tiết: 3 tiết)

4.2. Mô phỏng hệ thống bằng matlab/simulink

4.2.1. Thư viện SIMULINK

4.2.1.1. Sourese – các nguồn tín hiệu

Từ cửa sổ trình duyệt thư viện SIMULINK như hình 6.2, sau đó mở thư viện các khối nguồn như trên như hình 6.6. Toàn bộ thư viện này được chia thành hai phần: Cổng vào của mô hình và hệ thống con (Model and Subsystem Input), Máy phát tín hiệu (Signal Generator)

a. Cổng vào của mô hình và hệ thống con (Model and Subsystem Input)

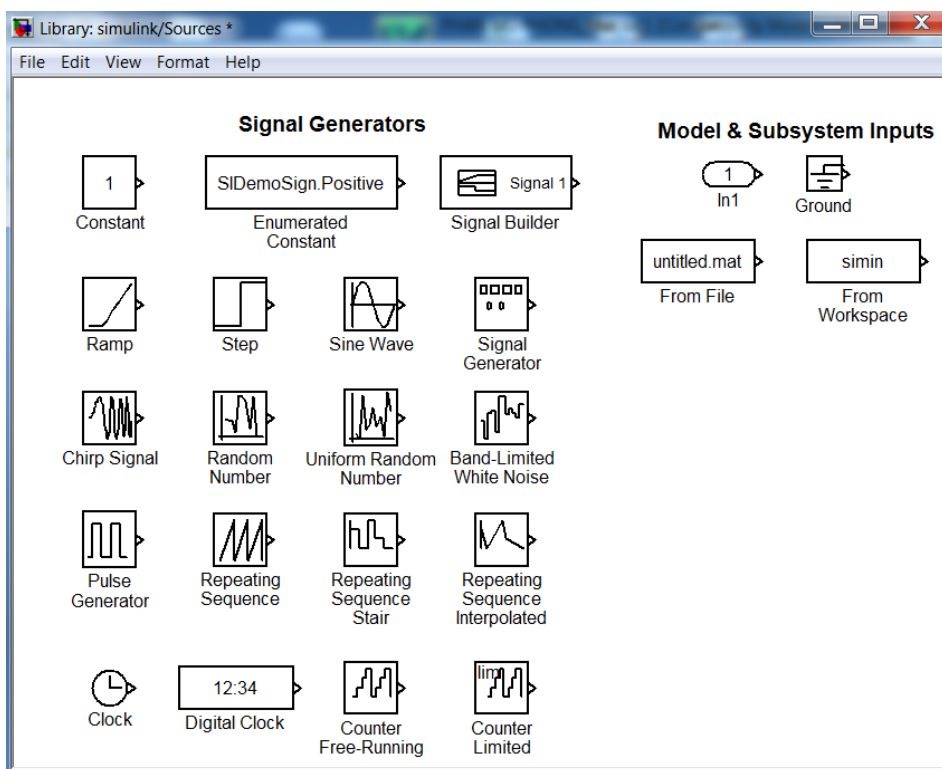
Cổng vào của mô hình và hệ thống con có chứa bốn khối:

* **Inport** – Khối cổng vào

Mô tả



Khối cổng vào tạo ra một cổng đầu vào cho một hệ thống con hoặc mô hình cấp trên của hệ thống.



Hình 4.19: Thư viện nguồn

Trong cửa sổ thiết lập các thông số có quy định cụ thể số cổng (*Port member*), kích thước của cổng vào (*Port dimensions*, bước thời gian mô phỏng và loại dữ liệu tín hiệu đầu vào (*Data type*).

* **Ground** – Khối tín hiệu mức không

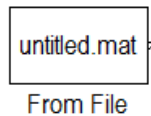
Mô tả



Khối tín hiệu mức không dành cho việc tạo ra một tín hiệu mức không. Nếu bất kì đầu vào nào của khối trong mô hình không được kết nối, thì khi thực hiện mô phỏng trong cửa sổ chính của MATLAB sẽ xuất hiện một tin nhắn cảnh báo. Để khắc phục điều này trên một đầu vào chưa được kết nối có thể gửi một tín hiệu từ khối *Ground*.

* **From File** – Khối đọc dữ liệu từ một tập tin (*file*) bên ngoài

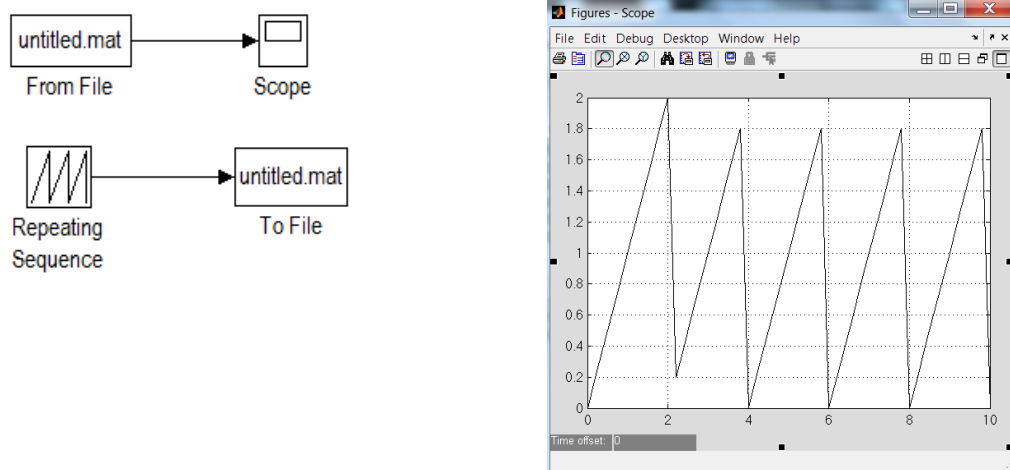
Mô tả



Trong cửa sổ thiết lập các thông số đưa ra tên của tập tin dữ liệu (*File name*), và thời gian trích mẫu (*Sample time*). Dữ liệu trong tập tin phải được thiết lập ở dạng ma trận: Ma trận bao gồm ít nhất hai dòng, giá trị thời gian ghi ở hàng đầu tiên của ma trận, và các hàng còn lại là các giá trị tín hiệu tương ứng với những thời điểm đã cho. Thời gian phải được ghi lại theo thứ tự tăng dần. Tín hiệu ra của khối chỉ chứa các giá trị tín hiệu và không có giá trị của biến thời gian.

SIMULINK cho phép người sử dụng thuận tiện tạo ra một mat-file với sự giúp đỡ của khối To File (Thư viện Sinks).

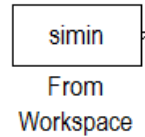
Ví dụ 4.1: Hình 4.8 cho thấy một ví dụ về cách sử dụng khối này, đọc các giá trị của tín hiệu răng cưa từ tập tin untitled.mat.(*FileTo_From_File*).



Hình 4.20: Mô hình các khối To File và From File

* **From Workspace**– Khối đọc dữ liệu từ không gian làm việc MATLAB

Mô tả

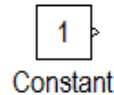


b. Máy phát tín hiệu (Signal Generator)

Máy phát tín hiệu chứa số lượng lớn các khối, dưới đây mô tả một số thiết bị thường dùng để mô phỏng các hệ thống điện cơ.

* **Constant** – Bộ tạo một tín hiệu liên tục không đổi

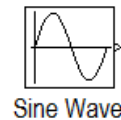
Mô tả



Giá trị không đổi có thể là số thực hoặc số phức, biểu thức tính, véc tơ hoặc ma trận.

* **Sine Wave** – Nguồn tín hiệu hình sin

Mô tả



Nguồn tín hiệu hình sin tạo ra một tín hiệu hình sin với một tần số, biên độ nhất định, pha và độ dịch pha bằng không. Khi hình thành tín hiệu đầu ra theo giá trị thời gian thì tín hiệu ra được xác định bởi biểu thức:

$$y = Amplitude * \sin(frequency * time + phase) + bias \quad (6.1)$$

Amplitude – Biên độ.

Bias – Thành phần không đổi của tín hiệu.

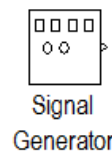
Frequency (rad/s) – Tần số (radian/s).

Phase (rads) – Pha ban đầu (rad).

Sample time – Bước thời gian mô phỏng. Được sử dụng để đồng bộ sự làm việc của các nguồn và các thành phần khác của mô hình theo thời gian.

* **Signal Generator** – Máy phát tín hiệu

Mô tả



Máy phát tín hiệu tạo ra một trong bốn loại tín hiệu tuần hoàn:

Sine – tín hiệu hình sin.

Square – tín hiệu vuông.

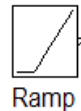
Sawtooth – tín hiệu dạng răng cưa.

Random – tín hiệu ngẫu nhiên.

Biên độ tín hiệu (*Amplitude*) và tần số (*Frequency*) được xác định trong các trường của số cài đặt các thông số.

* **Ramp** – nguồn tác động thay đổi tuyến tính

Mô tả



Nguồn tác động thay đổi tuyến tính tạo ra tín hiệu tuyến tính dạng $y = Slope * time + Initial\ output$. Trong cửa sổ cài đặt thông số ta xác định độ dốc của tín hiệu ra (*Slope*), thời gian bắt đầu tạo tín hiệu (*Start time*), giá trị đầu tiên của tín hiệu đầu ra của khối (*Initial output*).

Độ dốc được xác định bằng $tg\alpha = \frac{y}{t} = Slope$. Với y là tín hiệu ra, t là thời gian

* **Step** – máy phát tín hiệu bậc thang

Mô tả



Máy phát tín hiệu bậc thang tạo tín hiệu bậc thang có độ lớn (*Final value*) vào thời điểm (*Step time*).

* **Repeating Sequence** – khối tín hiệu tuần hoàn

Mô tả



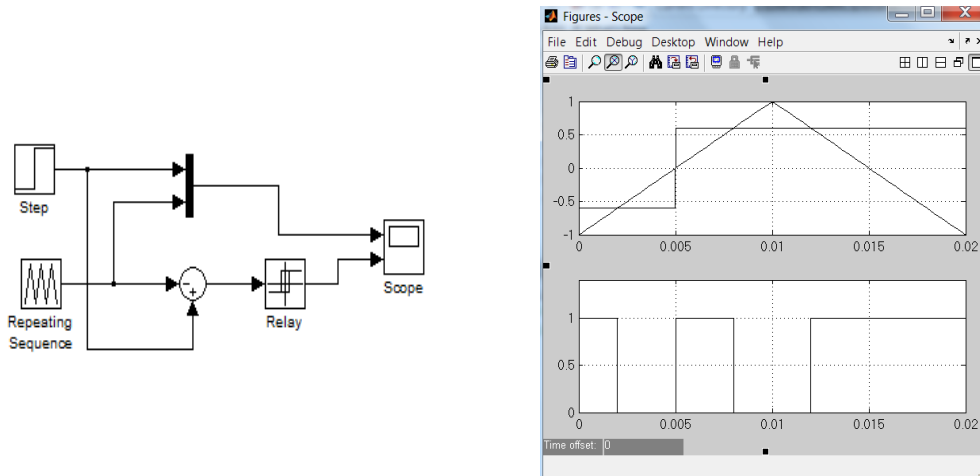
Khối tín hiệu tuần hoàn tạo ra tín hiệu tuần hoàn, trong cửa sổ cài đặt nhận thông số theo dạng hai véc tơ:

Time values – véc tơ các giá trị thời gian mô phỏng.

Output values – véc tơ các giá trị tín hiệu ở các thời điểm được chỉ ra bởi véc tơ Time values.

Ví dụ 4.2: Trên hình 4.21 trình bày một ví dụ về việc sử dụng khối *Step* và *Repeating Sequence* để xây dựng mô hình chức năng bộ điều chỉnh độ rộng xung dòng một chiều (PWM). Các thông số được xác định của khối *Step*: *Step time* = 0.005 (s), *Initial value* = -

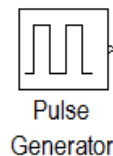
0.6, Final value=0.6. Các giá trị của thời gian mô phỏng trong khối *Repeating Sequence* được thiết lập bởi véc tơ [0 0.01 0.01 0.02], và các giá trị của tín hiệu đầu ra được tạo bởi véc tơ [-1 1 1 -1].



Hình 4.21: Mô hình điều chế độ rộng xung

* **Pulse Generator** – nguồn tín hiệu xung

Mô tả



Nguồn tín hiệu xung tạo ra một tín hiệu xung có biên độ nhất định (*Amplitude*), chu kỳ (*Period*) và độ dài của xung tính theo phần trăm của chu kỳ (*Pulse width*).

4.2.1.2. Sinks – Các khối thu tín hiệu

Toàn bộ thư viện được chia thành ba phần: Các đầu ra của mô hình và các hệ thống con (Model & Subsystem Output); khối quan sát dữ liệu (Data Viewers) và Các khối điều khiển thời gian mô phỏng (Simulation Control).

a. Các đầu ra của mô hình và các hệ thống con (Model & Subsystem Output)

Gồm có 4 khối con:

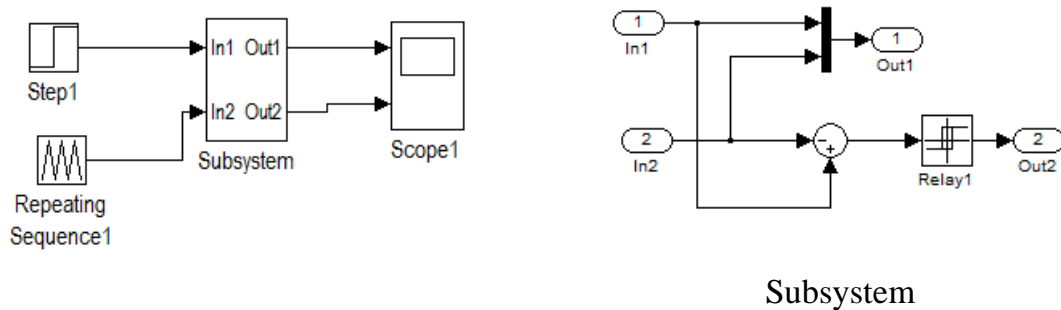
* **Out**– khối cổng đầu ra

Mô tả



Tạo một cổng đầu ra cho một hệ thống con hoặc cho mô hình cấp cao. Khi tạo ra một hệ thống con bằng cách sử dụng lệnh Edit/Create subsystem các cổng đầu ra được hình thành và đánh số tự động.

Ví dụ 4.3: Hình 4.22 cho thấy một ví dụ mô hình bộ điều biến độ rộng xung. Đây là một phần của chương trình bằng cách sử dụng lệnh Edit/Create chuyển đổi thành một hệ thống con. Sơ đồ hệ thống con này với các cổng đầu vào và đầu ra ở phía bên phải của hình 4.10.



Hình 4.22: Mô hình với hệ thống con, các cổng đầu vào và đầu ra

Cổng đầu ra trong mô hình cấp cao được sử dụng để truyền tín hiệu vào không gian làm việc MATLAB.

* **Terminator** – thiết bị nhận tín hiệu đầu cuối

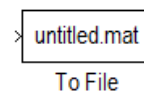
Mô tả



Trong trường hợp đầu ra của khối không kết nối với đầu vào của khối khác, SIMILINK sẽ hiển thị một thông điệp cảnh báo trong cửa sổ lệnh MATLAB. Để ngăn chặn điều này phải sử dụng một khối Terminator.

* **To File** – khối lưu trữ dữ liệu vào một tập tin

Mô tả

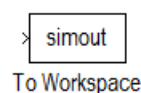


Khối ghi các dữ liệu nhận được đưa tới đầu vào của nó trong một file. Trong cửa sổ xác định các thông số của khối cần nhập tên của file (Filename) và tên của một biến chứa dữ liệu được ghi (Variable name). Nếu không chỉ định đường dẫn đầy đủ của tập tin, thì tập tin được lưu trong thư mục làm việc hiện thời.

Mức độ lược bỏ dữ liệu ghi vào file của tín hiệu vào được xác định bởi tham số - *Decimation*. Thời gian trích mẫu (*Sample time*) xác định mức độ rời rạc của việc ghi dữ liệu. Ví dụ về sử dụng khối này đã được xem xét trước đó (hình 6.8).

* **To Workspace** – khối lưu trữ dữ liệu vào không gian làm việc.

Mô tả



Khởi ghi các dữ liệu đưa đến đầu vào của nó vào trong không gian làm việc MATLAB. Tham số *Limit data points to last* chỉ ra số điểm trích mẫu dữ liệu tối đa cần lưu giữ. Nếu mô phỏng phát sinh các điểm dữ liệu nhiều hơn số điểm tối đa đã chỉ ra, thì chỉ lưu lại những điểm trích mẫu gần nhất. Để lưu giữ toàn bộ dữ liệu, ta phải đặt giá trị này bằng vô cùng (inf).


Định dạng lưu trữ dữ liệu được chọn từ danh sách thả xuống (Save format). Để đọc dữ liệu được lưu giữ trong không gian làm việc MATLAB có thể sử dụng các khối From Workspace (Thư viện Sources).

b. Phần để quan sát dữ liệu (Data Viewers) bao gồm bốn khối:

* **Scope** – máy hiện sóng

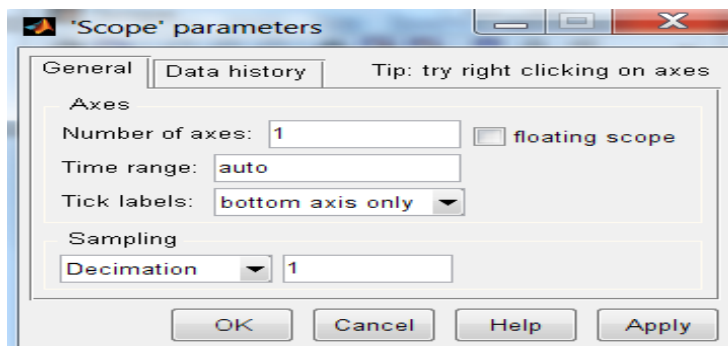


Xây dựng đồ thị của các tín hiệu được khảo sát là hàm phụ thuộc theo thời gian. Cho phép ta quan sát những thay đổi của tín hiệu trong quá trình mô phỏng. Để mở một cửa sổ xem tín hiệu cần nháy đúp chuột trái lên các khối hình ảnh. Trong trường hợp đó, nếu đưa đến đầu vào của khối này tín hiệu vectơ, thì các đường cong cho mỗi phần tử của véc tơ được xây dựng với một màu sắc riêng biệt.

Các thông số của khối được thiết lập trong cửa sổ Scope parameters, ta có thể mở nó bằng biểu tượng Parameters  trên thanh công cụ. Cửa sổ các thông số có hai thành phần:

General – các thông số chung.

Data history – các tham số lưu trữ tín hiệu trong không gian làm việc MATLAB. Tab các tham số chung mở một hộp thoại như trên hình 4.11.



Hình 4.23: Các tham số cơ bản mục General

Trên mục General, có các thông số sau:

Number of axes – số lượng đầu vào của máy hiện sóng. Khi thay đổi tham số này trên hình ảnh khối sẽ xuất hiện các cổng đầu vào bổ sung.

Time range – giá trị của khoảng thời gian hiển thị đồ họa. Nếu có thời gian tính toán của mô hình vượt quá thời gian định trước bởi tham số Time range, thì đầu ra của đồ thị được tạo ra thành từng phần, khi đó khoảng cách cho hiển thị từng phần của đồ thị bằng khoảng thời gian xác định đã cho Time range.

Tick labels – hiển thị / ẩn các trục và nhãn trục.

Sampling – sử dụng điều khiển này để lựa chọn hoặc là hệ số lược bỏ dữ liệu (Decimation factor) hoặc là khoảng thời gian trích mẫu (Sample time interval). Sau khi đã lựa chọn, cần phải nhập một số tương ứng vào trường dữ liệu ở bên cạnh.

Floating scope – chuyển máy hiện sóng sang chế độ “thả nổi” khi kích vào ô lựa chọn này.

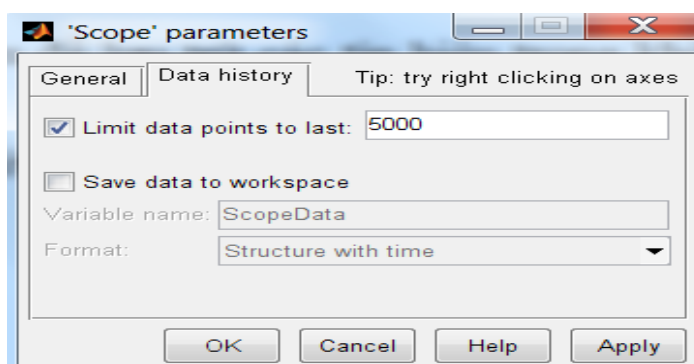
Trên tab Data history (Hình 4.12) xác định các thông số sau:

Limit data points to last – Số lượng tối đa điểm dữ liệu được thể hiện trên đồ thị. Nếu vượt quá con số này thì phần đầu tiên của đồ thị bị cắt bỏ. Trong trường hợp thông số Limit data points to last không được thiết lập, SIMULINK sẽ tự động tăng giá trị của thông số này để hiển thị tất cả các điểm tính toán.

Save data to workspace – lưu các giá trị tín hiệu vào không gian làm việc của MATLAB.

Variable name – tên biến để lưu trữ các tín hiệu trong không gian làm việc MATLAB.

Format – định dạng dữ liệu khi lưu giữ trong không gian làm việc MATLAB.




Hình 4.24: Tab Data history


* Floating Scope - Máy hiện sóng thả nổi

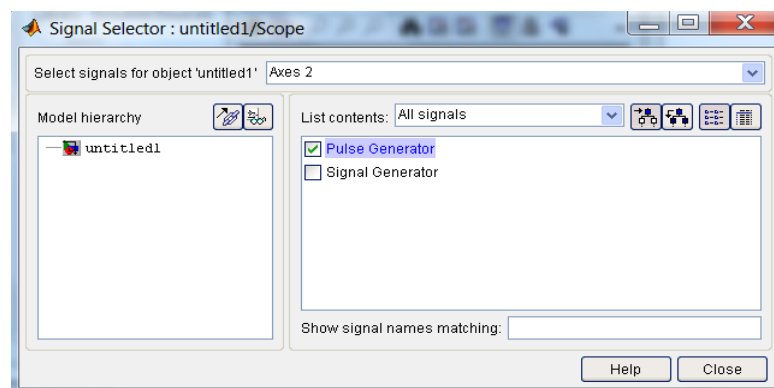
Mô tả



Floating
Scope

Máy hiện sóng thả nổi thực chất là một máy hiện sóng bình thường *Scope* được chuyển vào chế độ “tự do”. Trong chế độ này, khối máy hiện sóng không có đầu vào, đồng thời sự lựa chọn tín hiệu hiển thị được thực hiện bằng cách sử dụng công cụ  (*Signal selection*) trên thanh công cụ. Để chọn các tín hiệu cần thực hiện theo các bước sau:

1. Chọn hệ thống tọa độ để hiển thị đồ họa. Điều này đạt được bằng cách nhấn nút trái của chuột trên hệ thống mong muốn. Hệ thống được lựa chọn sẽ được tô đậm bởi màu xanh.
2. Mở hộp thoại *Signal Selector* bằng công cụ  như hình 4.13.
3. Đánh dấu tên các khối có các tín hiệu đầu ra cần khảo sát.

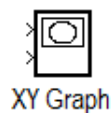


Hình 4.25: Cửa sổ hộp thoại *Signal Selector*

Sau khi thực hiện các tính toán trong cửa sổ khối *Floating Scope* các tín hiệu đã chọn sẽ được hiển thị.

* ***XY Graph*** – khối vẽ đồ thị

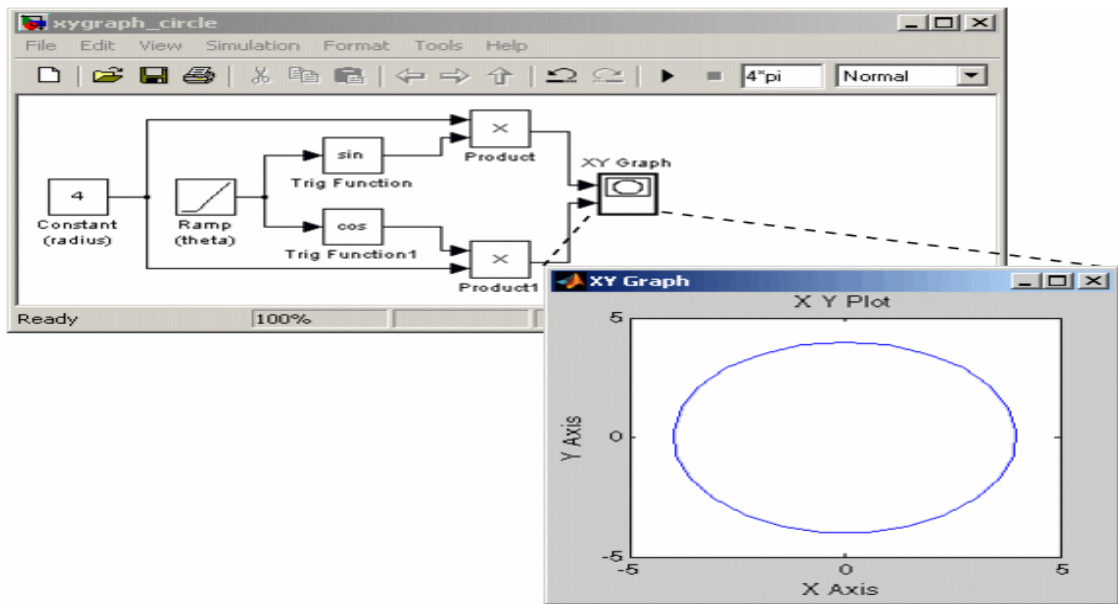
Mô tả



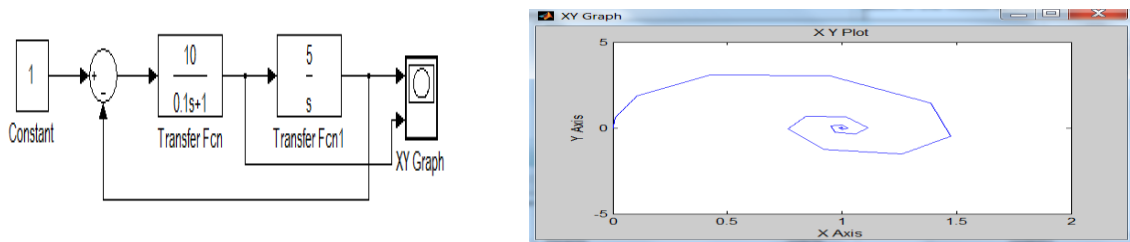
Khối vẽ đồ thị dùng để vẽ đồ thị một tín hiệu dưới dạng hàm của một tín hiệu khác (đồ thị có dạng $Y(X)$).

Khối có hai đầu vào, đầu vào phía trên được thiết kế để cung cấp tín hiệu là một đối số (X), đầu dưới – để cung cấp các giá trị của hàm số (Y).

Ví dụ 4.4 : Cách sử dụng máy vẽ đồ thị *XY Graph*



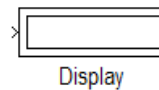
Hình 4.26: Ví dụ về cách sử dụng máy vẽ đồ thị XY Graph



Hình 4.27: Ví dụ về cách sử dụng máy vẽ đồ thị XY Graph

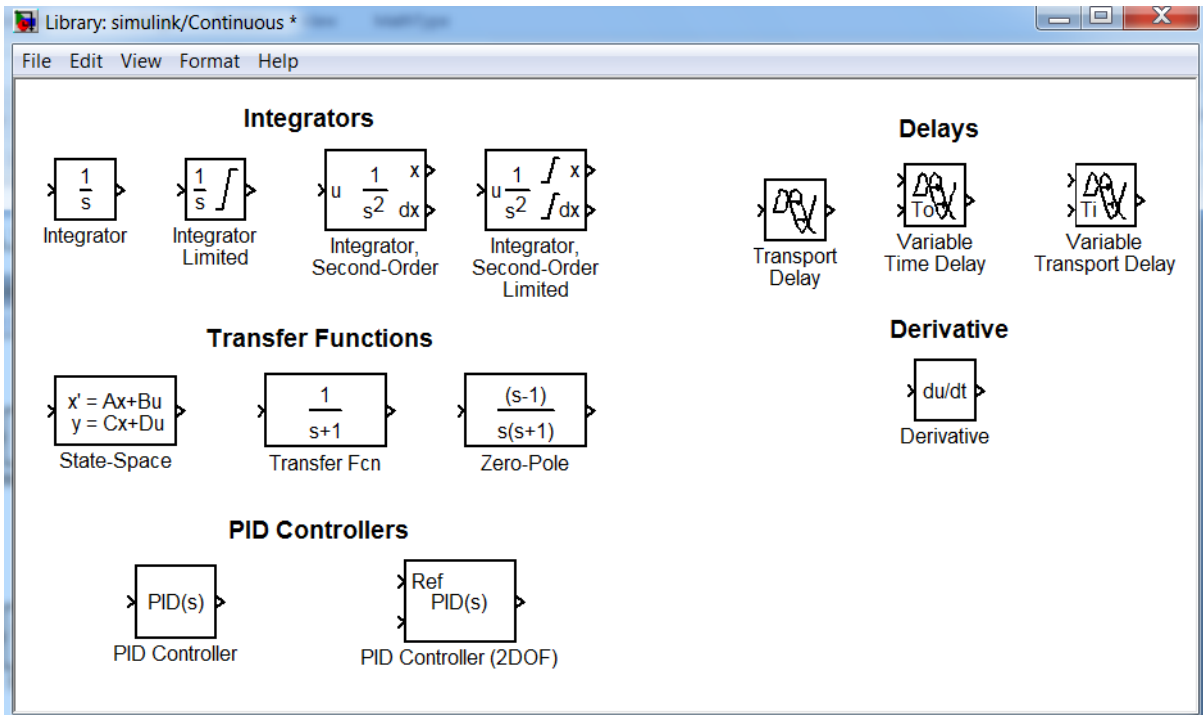
* **Display** – hiển thị số.

Mô tả



Hiển thị giá trị của tín hiệu ở dạng số. Khối Display có thể được sử dụng để hiển thị không chỉ những tín hiệu vô hướng, mà cả những tín hiệu véc tơ, ma trận và số phức.

4.2.1.3. Continuous – các khối liên tục



Hình 4.28: Thư viện của các khối liên tục

Tất cả các khối của thư viện được chia thành hai nhóm. Nhóm thứ nhất bao gồm các khối động học để phân tích và tổng hợp các hệ thống liên tục. Những khối này chứa cách ghi kí hiệu tượng trưng cho các phương trình vi phân mô tả các hệ thống điều khiển tuyến tính liên tục. Nhóm thứ hai bao gồm các khối trễ thời gian tín hiệu.

Các khối này, ngoài các chức năng cốt lõi của nó, còn cho phép làm giảm đáng kể thời gian mô phỏng trong các hệ thống có phản hồi (liên kết ngược) âm. Đối với khối đã cho cần phải đưa vào mạch phản hồi. Khi đó thời gian trễ sẽ được chọn sao cho sự có mặt của khối không ảnh hưởng đến các quá trình động học trong hệ thống.

a. Integrator – khối tích phân



Khối tích phân thực hiện tích phân tín hiệu đầu vào với các khả năng tác động bên ngoài lên tín hiệu đang tăng lên (*rising*), hoặc tín hiệu giảm (*falling*), hay lên tín hiệu tăng và giảm (*either*), hoặc một tín hiệu khác không (*level*). Khối cho phép đặt các điều kiện ban đầu (*Initial condition*) và thiết lập giới hạn tín hiệu ra (*Upper saturation limit. Lower saturation limit*).

b. Derivative – khối tính đạo hàm



Khối tính đạo hàm thực hiện phép vi phân số tín hiệu đầu vào. Khối này sử dụng cho các tín hiệu liên tục.

c. State-Space – Khối không gian trạng thái

Mô tả $\left. \begin{matrix} x' = Ax + Bu \\ y = Cx + Du \end{matrix} \right\} \text{State-Space}$

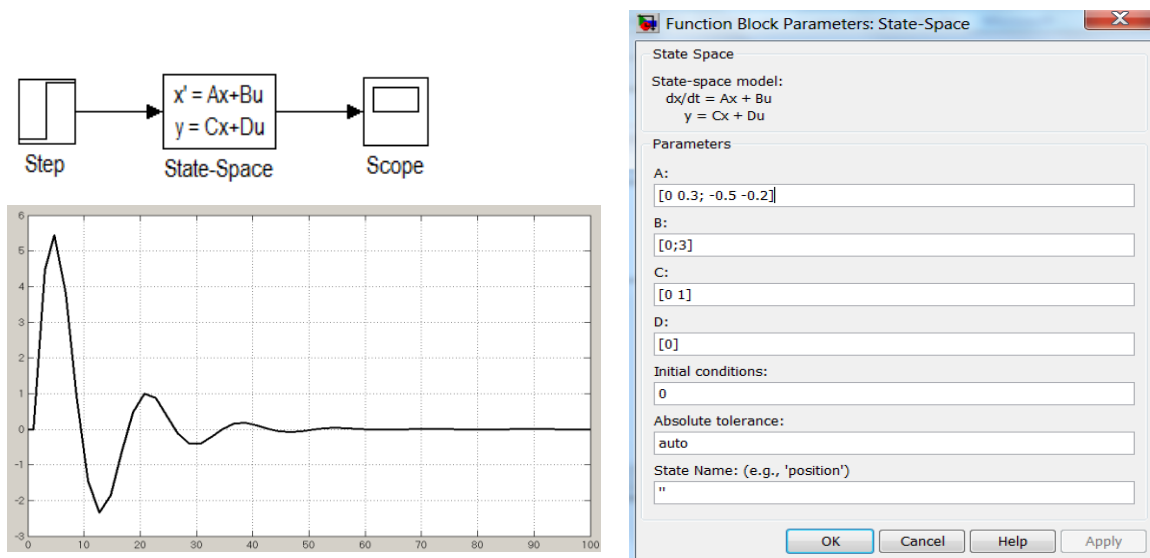
Khối State-Space tạo ra đối tượng động học được mô tả bởi các phương trình trong không gian trạng thái:

$$\begin{aligned} x' &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Trong đó x – véc tơ trạng thái, u – véc tơ các tác động đầu vào, y – véc tơ các tín hiệu đầu ra, A, B, C, D – tương ứng là các ma trận: hệ thống, đầu vào, đầu ra và ma trận liên hệ trực tiếp đầu ra với đầu vào. Khối cho phép đặt véc tơ các điều kiện đầu cho các biến trạng thái (Initial condition).

Ví dụ 4.3: Cho đối tượng có mô hình không gian trạng thái, với các ma trận A, B, C, D được cho như sau:

$$\begin{aligned} A &= \begin{bmatrix} 0 & 0,3 \\ -0,5 & -0,2 \end{bmatrix} & B &= \begin{bmatrix} 0 \\ 3 \end{bmatrix} \\ C &= [0 \quad 1] & D &= [0] \end{aligned}$$



Hình 4.29: Ví dụ sử dụng khối State-Space

d. Transfer Fcn – Khối hàm truyền

Mô tả $\left. \frac{1}{s+1} \right\} \text{Transfer Fcn}$

Transfer Fcn – khối hàm truyền, đặt làm hàm truyền là tỷ lệ của các đa thức

$$W(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}$$

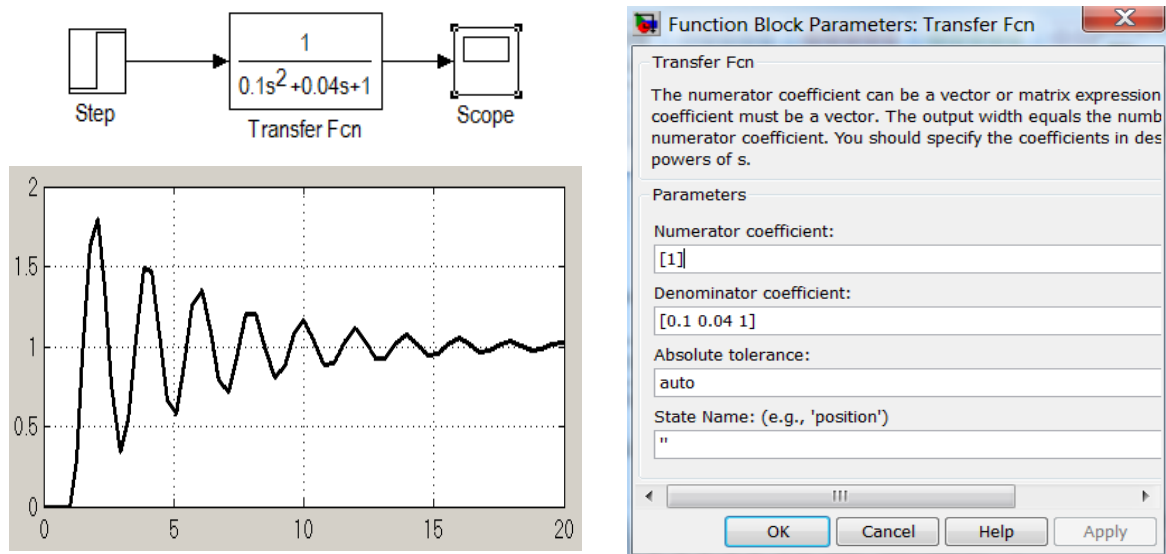
Khi đó bậc của đa thức tử số không được vượt quá bậc của mẫu số. Trong cửa sổ cài đặt tham số của khối đưa ra véc tơ các hệ số của đa thức tử số (Numerator) và véc tơ các hệ số của đa thức mẫu số (Denominator). Hình 4.30 trình bày một ví dụ về mô phỏng khâu dao động bằng khối Transfer Fcn.

Nếu hệ số của tử số được cho bởi ma trận, thì khối Transfer Fcn mô phỏng hàm truyền dạng véc tơ. Điều này có thể được hiểu là thực hiện đồng thời một số hàm truyền với các đa thức mẫu số giống nhau, nhưng các đa thức tử số khác nhau. Trong trường hợp này tín hiệu ra của khối này là một véc tơ và số lượng hàng của ma trận tử số xác định kích thước của tín hiệu đầu ra.

Ví dụ 4.4: Mô phỏng hệ thống có hàm truyền đạt

$$W(s) = \frac{1}{0,1s^2 + 0,004s + 1}$$

Tín hiệu kích thích là hàm bước nhảy step.



Hình 4.30: Ví dụ sử dụng khối Transfer Fcn

e. Zero-pole – Các định hàm truyền với các điểm cực và điểm không

Mô tả $\left[\frac{(s-1)}{s(s+1)} \right]$
Zero-Pole

Zero-pole – các định hàm truyền với các điểm cực và điểm không:

$$H(s) = K \frac{(s - z_1) \cdot (s - z_2) \dots (s - z_m)}{(s - p_1) \cdot (s - p_2) \dots (s - p_m)}$$

Trong đó:

z_1, z_2, \dots, z_m – nghiệm của đa thức tử số (số điểm không),

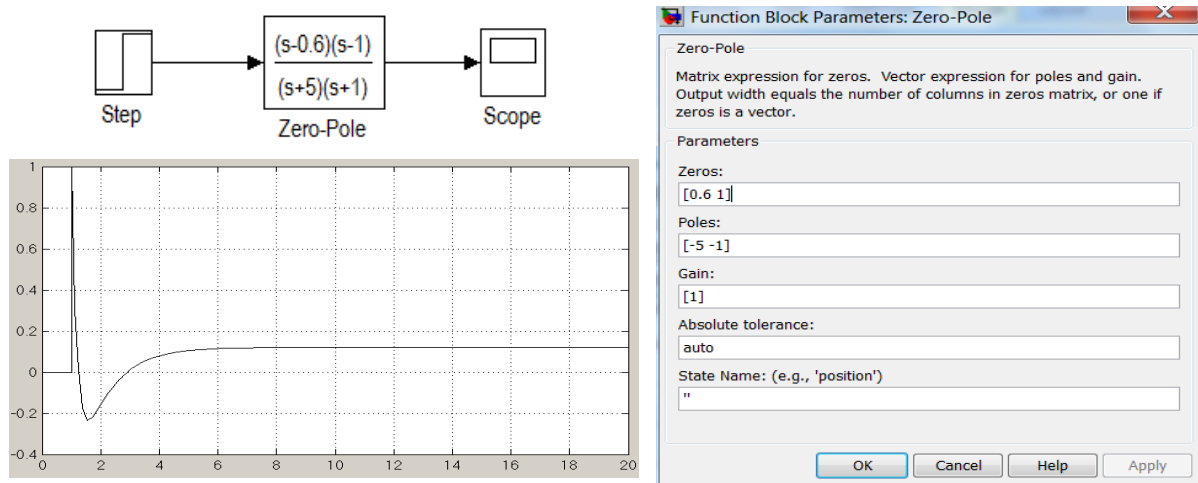
p_1, p_2, \dots, p_m – nghiệm của đa thức mẫu số (điểm cực),

K – hệ số hàm truyền. Trong cửa sổ cài đặt tham số của khối ta cần xác định rõ các véc tơ của nghiệm đa thức tử (Zeros), véc tơ nghiệm của đa thức mẫu số (Poles), véc tơ vô hướng hay véc tơ hệ số của hàm truyền (Gain).

Số lượng điểm không phải không được vượt quá số lượng điểm cực của hàm truyền. Zeros hoặc poles có thể được cho ở dạng số phức.

Ví dụ 4.5: Mô phỏng hệ thống có

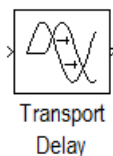
$$H(s) = \frac{(s - 0,6)(s - 1)}{(s + 5)(s + 1)}$$



Hình 4.31: Ví dụ về cách sử dụng các khối Zero-Pole

f. Transport Delay – Khối độ trễ của tín hiệu

Mô tả:

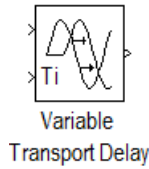


Transport Delay – khối độ trễ của tín hiệu. Đảm bảo độ trễ của tín hiệu đầu vào theo thời gian cho trước. Trong cửa sổ cài đặt thông số của khối cho phép xác định thời gian trễ của tín hiệu (*Time Delay*), giá trị ban đầu của tín hiệu vào (*Initial input*), kích thước của bộ nhớ, phân bổ cho việc lưu trữ tín hiệu trễ (*Buffer size* – cho ở dạng số byte, bội số của 8).

Khi thực hiện mô phỏng giá trị của tín hiệu vào, giá trị thời gian mô phỏng tương ứng với nó được lưu trữ ở bộ đệm trong của khối Transport Delay. Độ trễ giá trị của các tín hiệu được tách ra từ bộ đệm và chuyển giao cho đầu ra của khối.

g. Variable Transport Delay – khối điều khiển sự trễ của tín hiệu

Mô tả

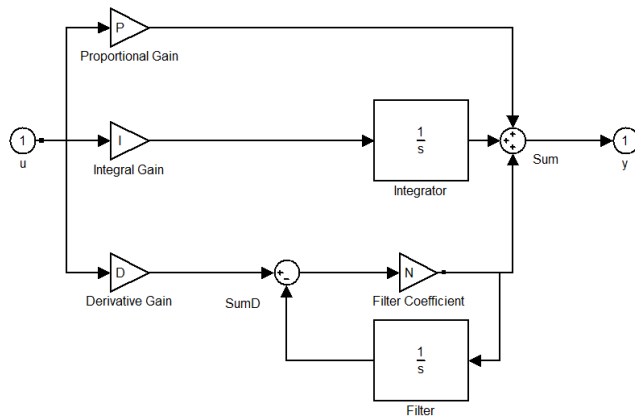
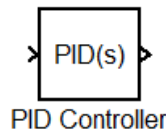


Variable Transport Delay – khối điều khiển sự trễ của tín hiệu. Thực hiện sự trễ của tín hiệu đầu vào cho bởi độ lớn của tín hiệu điều khiển.

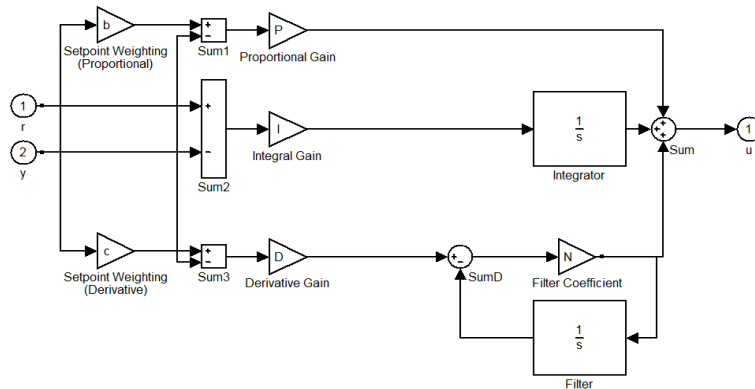
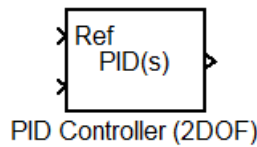
Trong cửa sổ cài đặt thông số của khối cho thời gian trễ tối đa của tín hiệu (*Maximum delay*), giá trị ban đầu của tín hiệu đầu vào (*Initial input*), kích thước của bộ nhớ phân bổ cho việc lưu trữ tín hiệu trễ (*Buffer size* – cho ở dạng số byte, bội số của 8). Khối điều khiển trễ *Variable Transport Delay* làm việc tương tự như khối *Transport Delay*.

h. PID controller

Mô tả



Hình 4.32: Cấu trúc bên trong của khối PID controller



Hình 4.33: Cấu trúc bên trong của khối PID controller (2DOF)

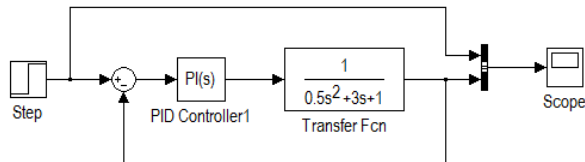
Cách tìm các thông số bộ điều khiển PID

Ví dụ 4.6: Ta có hàm truyền của đối tượng

$$W(s) = \frac{1}{0,5s^2 + 3s + 1}$$

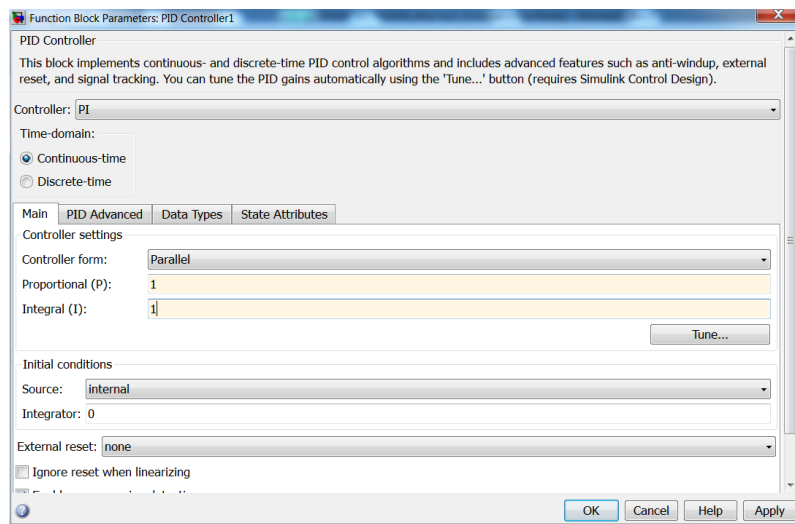
Ta sử dụng bộ điều khiển PID để điều khiển đối tượng trên.

Bước 1: Vẽ mô hình trên simulink như hình 4.34



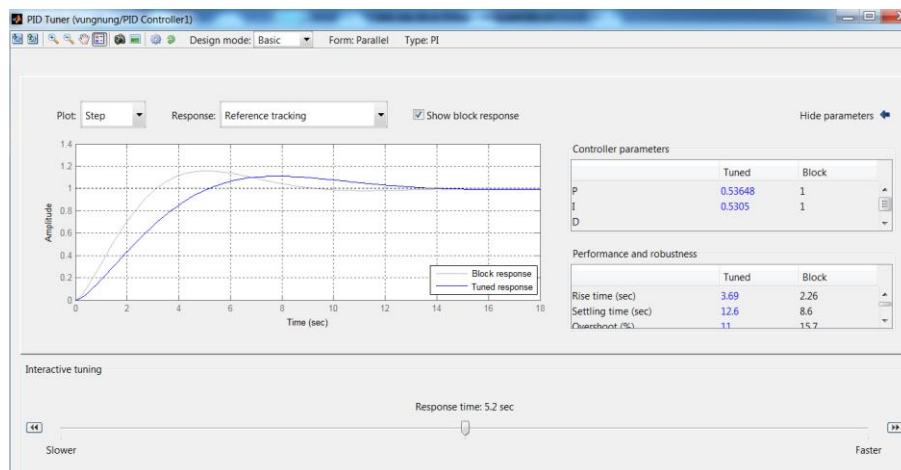
Hình 4.34: Mô hình điều khiển sử dụng bộ PID

Bước 2: Kích đúp chuột trái vào khối PID Controller1, sẽ xuất hiện như hình 4.35



Hình 4.35: Bảng thông số bộ điều khiển PID

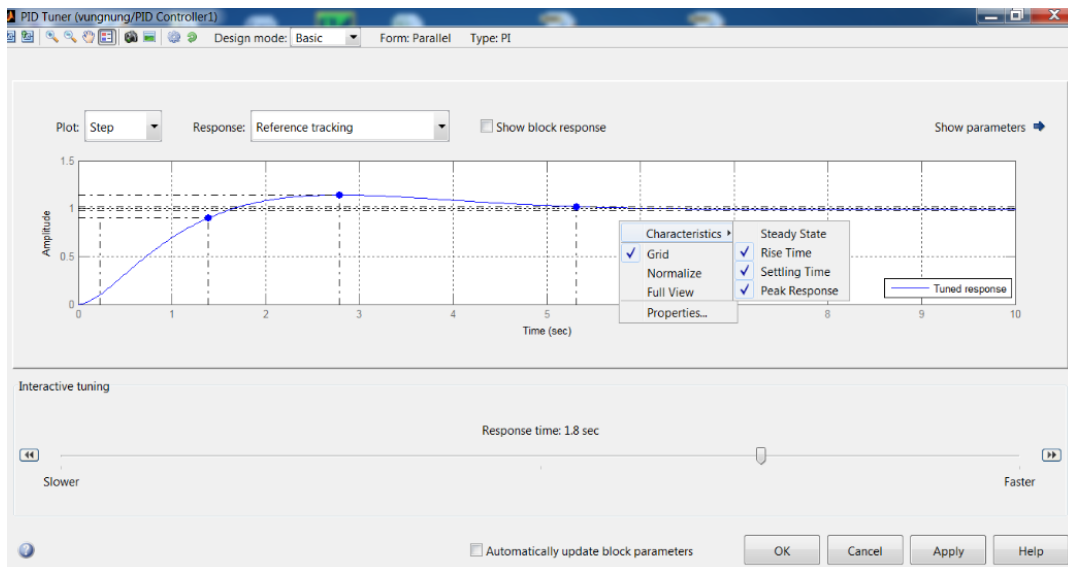
Bước 3: Lựa chọn bộ điều khiển (PID, PI, PD, P, I) ở mục Controller → Apply → Tune... sẽ xuất hiện bảng như hình 4.36



Hình 4.36: PID tuner của Matlab

Kết quả này cho ta các tham số P,I,D và N(khâu lọc) của bộ điều khiển trong bảng *controller parameters* (hình 4.36). Bảng dưới là *Performance and robustness* cho các thông số để đánh giá chất lượng của bộ điều khiển như: thời gian quá độ, độ quá điều chỉnh...

Có thể đánh dấu các điểm đặc biệt trên đồ thị bằng cách kích chuột phải vào màn hình của đồ thị rồi chọn characteristic như hình 4.37

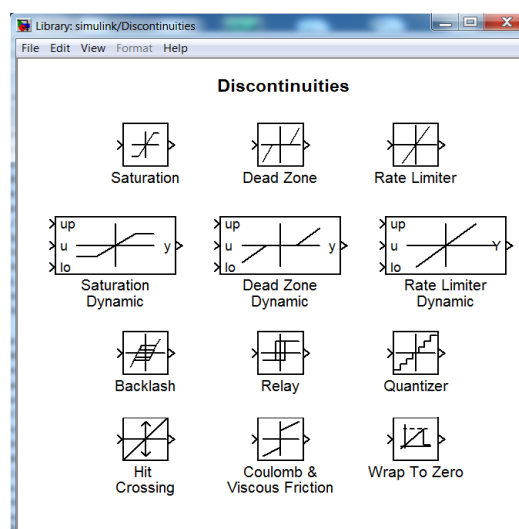


Hình 4.37: Đánh dấu các điểm đặc biệt trên đồ thị đáp ứng của PID tuner

Chú ý tích vào ô *Automatically update parameter* để bộ điều khiển tự động cập nhật các tham số mới.

Bước 4: Tinh chỉnh các tham số của bộ điều chỉnh thành trượt *Response time* (thay đổi thời gian thiết lập). Sau đó vào ô *Design mode* trên cùng, chọn *Extended*. Di chuyển hai thanh trượt *Bandwidth* và *Phase margin* để được chất lượng tốt nhất.

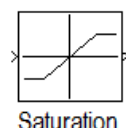
4.2.1.4. Discontinuities – Các khối phi tuyến



Hình 4.38: Thư viện các khối phi tuyến

a. Saturation – khối giới hạn

Mô tả

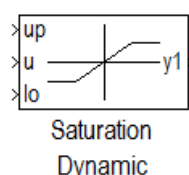


Khối Saturation thực hiện giới hạn giá trị tín hiệu đầu ra. Trong cửa sổ cài đặt thông số khối quy định giới hạn ngưỡng trên và dưới (Upper limit, Lower limit).

Tín hiệu đầu ra của khối bằng đầu vào nếu giá trị của nó không vượt quá ngưỡng giới hạn. Khi tín hiệu ra của khối đạt đến cấp độ giới hạn bởi tín hiệu đầu vào sẽ không còn thay đổi và tương đương với ngưỡng.

b. Saturation Dynamic

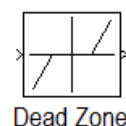
Mô tả



Khối Saturation Dynamic tương tự như khối Saturation. Trong khối này giới hạn trên và dưới của tín hiệu đầu ra được cho theo đầu vào up, io và có thể thay đổi trong khi mô phỏng.

c. Dead Zone

Mô tả



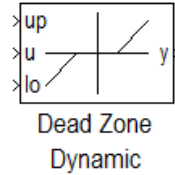
Dead Zone – khối với cùng không nhạy. Thực hiện một sự phụ thuộc phi tuyến dạng “vùng không cảm nhận (vùng chết)”.

Trong cửa sổ cài đặt thông số của khối quy định điểm bắt đầu vùng chết (Start of dead zone – ngưỡng dưới), và điểm kết thúc vùng chết (End of dead zone – ngưỡng trên).

Nếu giá trị của tín hiệu đầu vào nằm bên trong vùng chết, thì tín hiệu đầu ra của khối bằng không. Nếu tín hiệu đầu vào lớn hơn ngưỡng trên của vùng chết, thì tín hiệu đầu ra là đầu vào trừ đi ngưỡng.

d. Dead Zone Dynamic

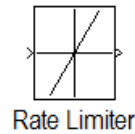
Mô tả



Khối Dead Zone Dynamic cũng có chức năng tương tự. Trong khối này mức trên và dưới của vùng chết tín hiệu đầu ra quy định theo đầu vào u_p , u_l và có thể thay đổi khi mô phỏng.

e. Rate Limiter

Mô tả

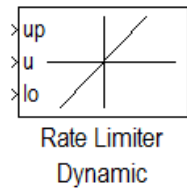


Rate Limiter – khối giới hạn tốc độ thay đổi của tín hiệu. Khối này đảm bảo giới hạn tốc độ thay đổi của tín hiệu (đạo hàm bậc nhất).

Trong cửa sổ cài đặt thông số của khối quy định mức giới hạn tốc độ tăng tín hiệu (Rising slew rate), mức giới hạn tốc độ suy giảm tín hiệu (Falling slew rate).

f. Rate Limiter Dynamic

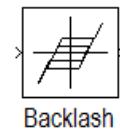
Mô tả



Khối Rate Limiter Dynamic cũng có chức năng tương tự như trên. Trong khối này mức trên và dưới của giới hạn tốc độ thay đổi tín hiệu đầu ra được hạn định theo các đầu vào u_p , u_l và có thể thay đổi khi mô phỏng.

g. Backlash

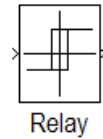
Mô tả



Backlash – khối khe hở trong bộ truyền cơ khí. Mô phỏng tính phi tuyến của khe hở. Trong cửa sổ cài đặt thông số khối quy định chiều rộng của khe hở (Deaband width), giá trị ban đầu của tín hiệu đầu ra (Initial output).

h. Relay – khối role

Mô tả

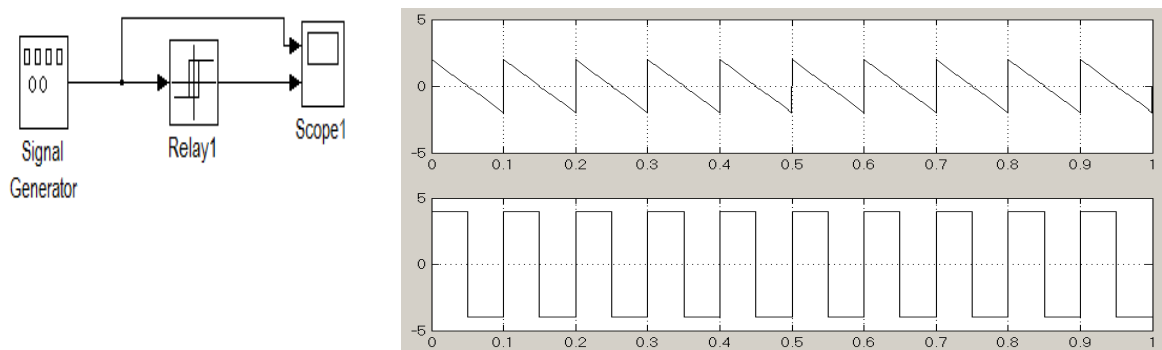


Khối role thực hiện tính phi tuyến role. Trong cửa sổ cài đặt thông số khối quy định ngưỡng mở và tắt (Switch on point – giá trị của tín hiệu đầu vào làm role đóng mạch, Switch off point – giá trị tín hiệu đầu vào làm role cắt mạch).

Giá trị tín hiệu đầu ra trong trạng thái đóng mạch (Output when on), và tín hiệu đầu ra ở trạng thái cắt mạch (Output when off).

Việc chuyển đổi từ một trạng thái này sang trạng thái khác xảy ra đột ngột khi tín hiệu đầu vào đạt tới các ngưỡng đóng và cắt mạch.

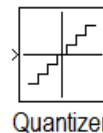
Ví dụ 4.8: Sử dụng khối Relay



Hình 4.39 Ví dụ sử dụng khối Relay

i. Quantizer – khối lượng tử theo mức

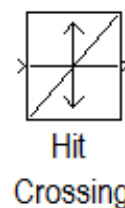
Mô tả



Quantizer – khối lượng tử theo mức. Khối này đảm bảo lượng tử hóa tín hiệu đầu vào với bước như nhau dạng bậc thang. Trong cửa sổ cài đặt thông số của khối cần chỉ rõ bước lượng tử hóa (Quantization interval).

j. Hit Crossing – khối xác định thời điểm giao cắt ngưỡng

Mô tả



Hit Crossing – khối xác định thời điểm giao cắt ngưỡng. Xác định thời điểm khi tín hiệu đầu vào giao cắt ngưỡng giá trị cho trước. Trong cửa sổ cài đặt thông số khối

cần nhập vào giá trị ngưỡng giao cắt (Hit Crossing offset) và hướng giao cắt (Hit crossing direction):

Rising – tăng.

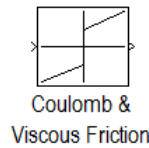
Falling – giảm

Either – cả hai hướng.

Tại thời điểm giao cắt ngưỡng khối tạo ra một tín hiệu bằng đơn vị với độ vào là một bước theo thời gian mô phỏng.

k. Coulomb and Viscous Friction – khối ma sát khô và nhớt

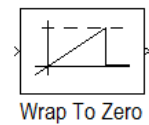
Mô tả



Coulomb and Viscous Friction – khối ma sát khô và nhớt. Mô phỏng hiệu ứng của ma sát khô và nhớt trong bộ truyền cơ. Trong cửa sổ cài đặt thông số khối cần nhập vào giá trị ma sát khô (Coulomb Friction value (Offset)) và hệ số ma sát nhớt (Coefficient of viscous friction (Gain)).

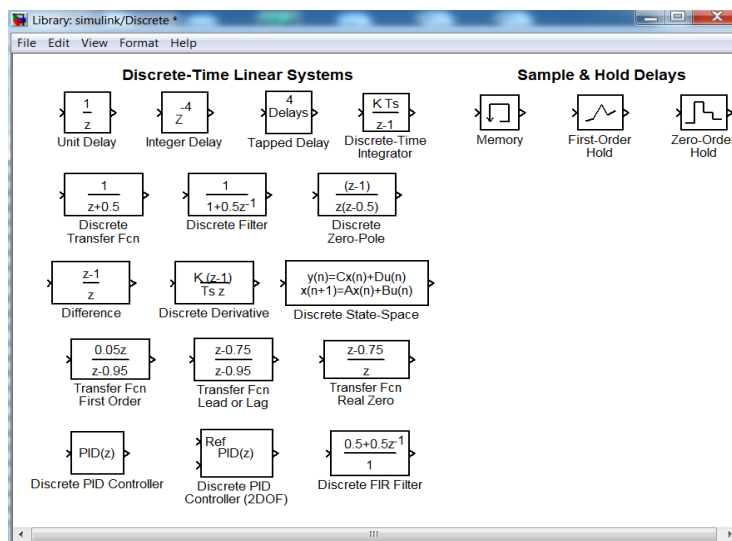
l. Wrap to Zero

Mô tả



Wrap to Zero – Nếu tín hiệu vào cao hơn mức ngưỡng, thì tín hiệu ra bằng không. Ngược lại, tín hiệu ra bằng tín hiệu vào.

4.2.1.5. Discrete – Các khối rời rạc



Hình 4.40: Thư viện khối Discrete

Tất cả các khối của thư viện này được chia thành hai nhóm. Nhóm thứ nhất bao gồm các khối độ trễ khác nhau và các khối động học dùng để phân tích và tổng hợp các hệ thống rời rạc. Những khối này đại diện cho các phương trình khác nhau mô tả các hệ thống điều khiển rời rạc.

Nhóm thứ hai (Sample and Hold Delays) bao gồm khối bộ nhớ và các khối ngoại suy.

Unit Delay, Integer Delay, Tapped Delay – độ trễ đơn vị, độ trễ nguyên, độ trễ phân nhánh. Thực hiện độ trễ tín hiệu đầu vào theo một (Unit Delay) hoặc một số bước (Integer Delay) thời gian mô phỏng. Tín hiệu đầu ra của khối Tapped Delay là một vectơ mà số lượng các phần tử của nó được chỉ ra trong trường Number of Delays. Trong cửa sổ cài đặt thông số của khối ta định rõ giá trị ban đầu của tín hiệu đầu ra (Initial condition), bước thời gian lấy mẫu (Sample time).

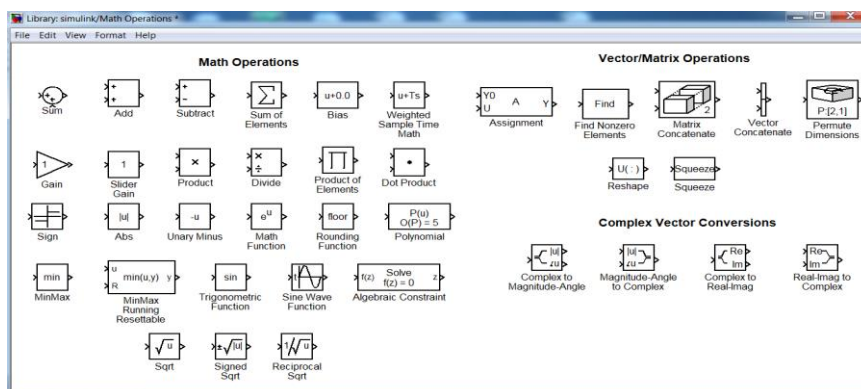
Các khối tích phân rời rạc (Discrete-Time Integrator), các khối sai phân, các hàm rời rạc (Difference, Diskrete Derivative), khối hàm truyền rời rạc (Discrete Transfer Fcn), hàm truyền rời rạc với các điểm cực và điểm không (Discrete Zero-pole), khối đối tượng động học rời rạc theo biến trạng thái (Discrete State-Space) có chức năng tương tự như các khối tương ứng dùng cho các hệ thống liên tục. Trong các khối này sử dụng các phương pháp tính toán số khác nhau.

1. Forward Euler – phương pháp Euler trực tiếp.
2. Backward Euler – phương pháp Euler ngược.
3. Tutsim – Phương pháp hình thang.

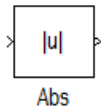
Mỗi khối đều có một tham số - bước rời rạc hóa theo thời gian (Sample time).

Diskrete filter – bộ lọc rời rạc. Là một khâu động học mà hàm truyền của nó nhận được khi chia hàm truyền rời rạc cho z^n . Bộ lọc rời rạc được sử dụng rộng rãi trong các hệ thống truyền thông và thông tin liên lạc. Chúng cũng được sử dụng trong các hệ thống điều khiển kỹ thuật số khi lọc tiếng ồn và những sai lệch.

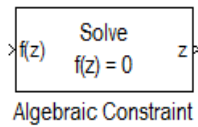
4.2.1.6 Math Operations – Các phép tính toán học



Thư viện khối Math Operations



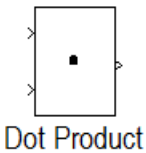
Đầu ra của khối Abs là giá trị tuyệt đối của tín hiệu đầu vào.



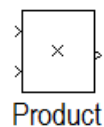
Khối Algebraic Constraint sẽ đưa tín hiệu đầu vào $f(z)$ về 0 và đưa ra giá trị z . Giá trị của z sẽ được hồi tiếp về đầu vào.



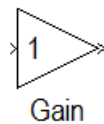
Đầu ra của khối Sum là tổng của các tín hiệu đầu vào. Nếu tín hiệu đầu vào là scalar thì tín hiệu tổng cũng là scalar. Khối Sum tính tổng của từng phần tử. Nếu đầu vào chỉ có 1 véctơ thì đầu ra sẽ là tổng của tất cả các phần tử của véctơ đó



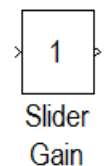
Khối Dot Product tính tích scalar (vô hướng) của các véctơ đầu vào.



Khối Product thực hiện phép nhân hay chia giữa các tín hiệu đầu vào (1-D hay 2-D).

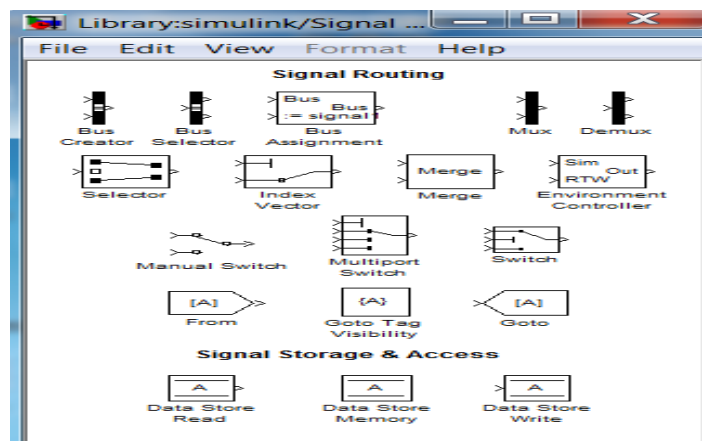


Khối Gain có tác dụng khuếch đại tín hiệu đầu vào. Thông số khuếch đại được thông báo tại ô Gain trong Block Parameter.

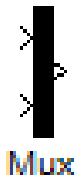

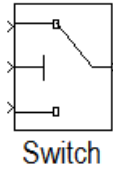


Khối Slider Gain có tác dụng khuếch đại tín hiệu đầu vào. Thông số khuếch đại có thể được thay đổi trong quá trình mô phỏng.

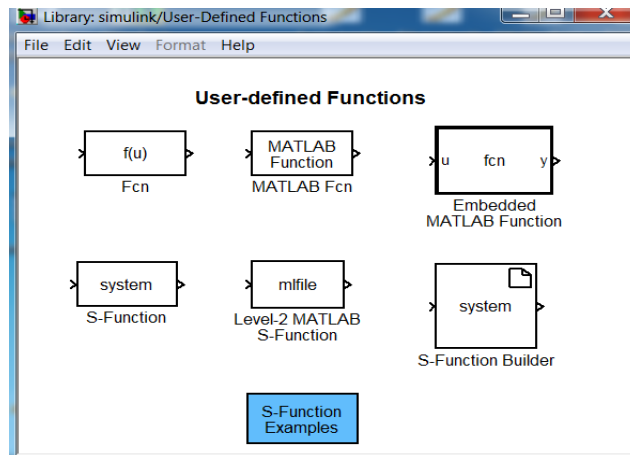
4.2.1.7. Signal Routing – khối định tuyến tín hiệu



Hình 4.41: Thư viện của Signal Routing

 <p>Mux</p>	<p>Khối hợp tín hiệu - Mux .</p> <p>Khối Mux có tác dụng giống như một bộ dồn kênh (multiplexer), có tác dụng ghép các tín hiệu 1-D riêng lẻ thành một tín hiệu mới. Tại ô tham số Number of inputs ta có thể khai báo tên, số lượng tín hiệu ngõ vào.</p>
 <p>Demux</p>	<p>Khối phân tín hiệu- Demux</p> <p>Khối Demux (phân kênh) có tác dụng ngược với Mux: Tách tín hiệu được ghép thành nhiều tín hiệu riêng rẽ trở lại.</p>
 <p>Switch</p>	<p>Khối chuyển mạch -Switch</p> <p>Khối Switch có tác dụng chuyển mạch, đưa tín hiệu từ ngõ vào 1 hoặc 3 tới ngõ ra. Tín hiệu điều khiển chuyển mạch được đưa tới ngõ vào 2 (ngõ vào giữa). Ngưỡng giá trị điều khiển chuyển mạch được khai báo bằng tham số Threshold. Khi tín hiệu điều khiển \geq Threshold, ngõ ra được nối với ngõ vào 1. Khi tín hiệu điều khiển \leq Threshold, ngõ ra được nối với ngõ vào 3.</p>

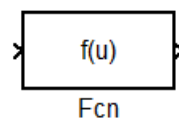
4.2.1.8. User – Defined Function



Hình 4.42: Thư viện User – Defined Function

a. Fcn (Function)

Mô tả

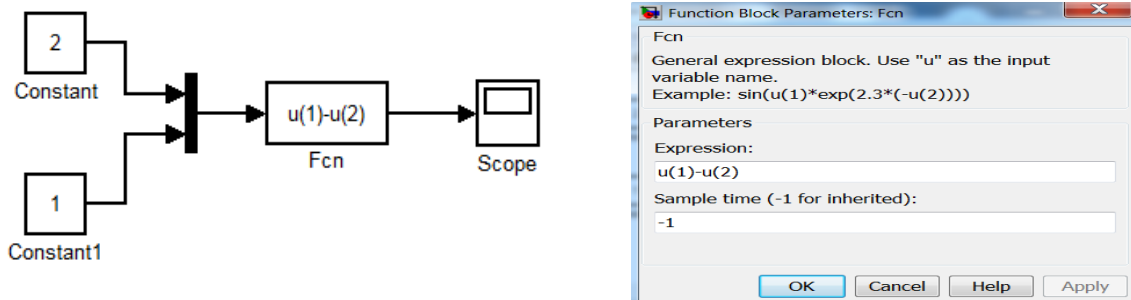


Khối Fcn có thể khai báo một hàm của biến vào dưới dạng một biểu thức. u là tín hiệu vào của khối Fcn và nó là một vectơ đầu vào.

$u(i)$ hoặc $u[i]$: phần tử thứ i của tín hiệu vào dạng vectơ.

$u(1)$ hay viết một mình u : phần tử đầu tiên của vectơ tín hiệu vào.

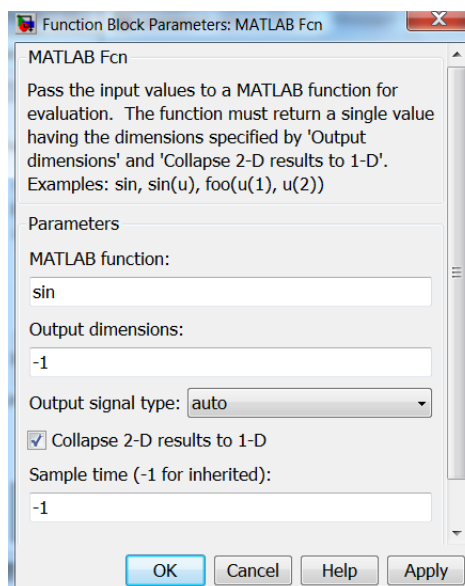
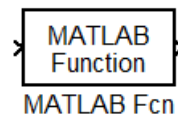
Ví dụ 4.9: Tín hiệu vào $u(2,1)$. Ta thực hiện tính biểu thức $y=u(1)-u(2)$. Ở đây $u(1)$ là phần tử đầu tiên của vectơ vào, nó có giá trị bằng 2; $u(2)$ là phần tử thứ 2 của vectơ vào, nó có giá trị bằng 1



Hình 4.43: Mô tả ví dụ 4.9

b. MATLAB Fcn (Matlab Function)

Mô tả



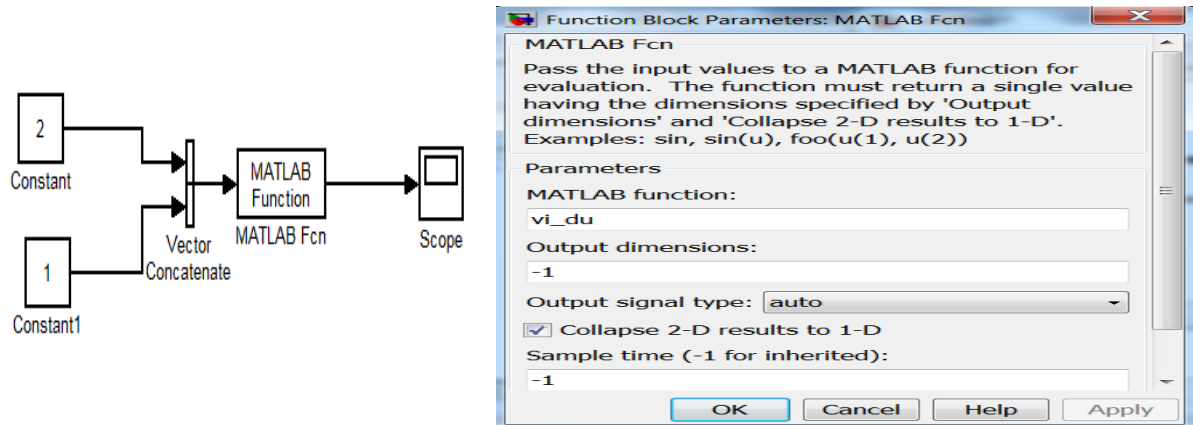
Hình 4.44: Hộp thoại của khối MATLAB Fcn

Khối *MATLAB Fcn* là một dạng mở rộng của khối Fcn. Tại ô *MATLAB function* ta có thể khai báo một biểu thức toán hay một hàm MATLAB (viết dưới dạng *m-File*) của biến đầu vào.

Output dimensions: Kích thước đầu ra. Nếu kích thước đầu ra giống kích thước đầu vào thì ta để là -1.

Output signal type: Loại tín hiệu ra

Ví dụ 4.10 : Cách sử dụng khối MATLAB Fcn



Hình 4.45: Ví dụ sử dụng khối MATLAB Fcn

Tạo m-file, sau đó lưu file có tên là *vi_du.m*

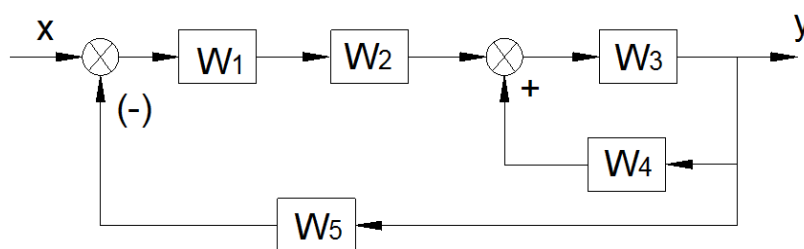
```
function y = vi_du(input)
    a = input(1);
    b = input(2);
    c = a+b;
    d = a-b;
    y = [c d];
end
```

Bài tập cuối bài

Bài 1: Cho hệ thống có sơ đồ khối như sau:

$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$



Sử dụng bộ điều khiển PID trong Matlab/Simulink để điều khiển hệ thống trên.

Bài 2: Cho lò nhiệt điện trở có hàm truyền đạt như sau:

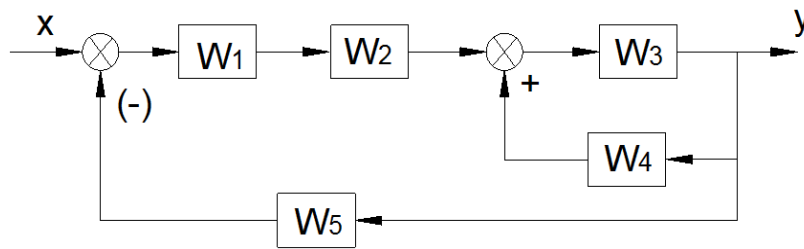
$$W(s) = \frac{K.e^{-T_2s}}{T_1s + 1}$$

Với $K=100, T_1=250, T_2=50$. Sử dụng bộ điều khiển PID trong Matlab/Simulink để điều khiển đối tượng trên.

Bài 3: Cho hệ thống có sơ đồ khối như sau:

$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$

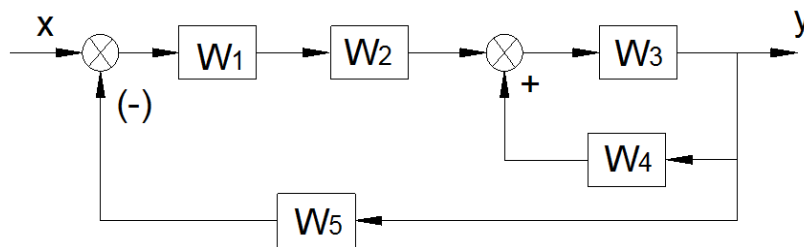


Sử dụng bộ điều khiển PID trong Matlab/Simulink để điều khiển hệ thống trên.

Bài 4: Cho hệ thống có sơ đồ khối như sau:

$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$

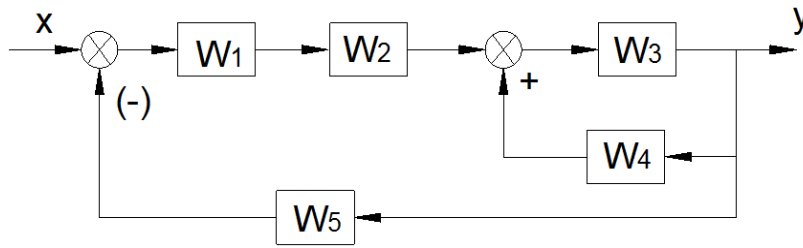


Xác định đáp ứng của hệ thống trên với tín hiệu vào có dạng Step.

Bài 5: Cho hệ thống có sơ đồ khối như sau:

$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$



Xác định đáp ứng của hệ thống trên với tín hiệu vào có dạng Constant.

Câu hỏi trắc nghiệm

Câu 1: Khối Terminator có vai trò gì ?

- A. Thiết bị nhận tín hiệu đầu cuối
- B. Khối lưu trữ dữ liệu vào một tập tin
- C. Khối lưu trữ dữ liệu vào không gian làm việc
- D. Máy hiện sóng

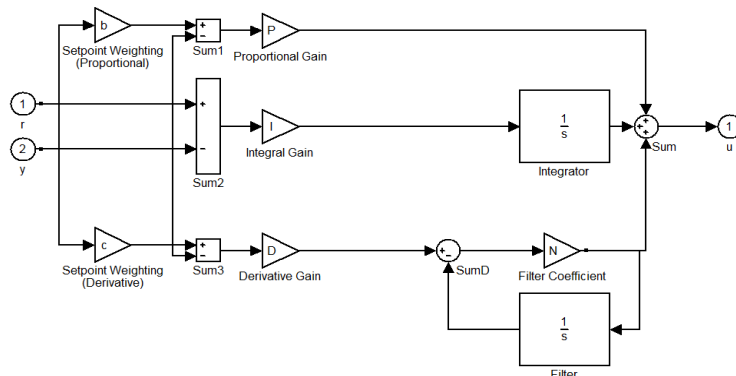
Câu 2: Máy phát tín hiệu Signal Generator tạo ra tín hiệu nào sau đây

- A. Tín hiệu hình sin.
- B. Tín hiệu vuông.
- C. Tín hiệu dạng răng cưa.
- D. Tín hiệu liên tục không đổi.

Câu 3: : Cho hệ thống có hàm truyền e^{-Ts} sẽ được biến đổi tương đương thành

- A. $e^{-Ts} = \frac{T}{Ts + 1}$
- B. $e^{-Ts} = \frac{1}{Ts + 1}$
- C. $e^{-Ts} = \frac{1}{s + 1}$
- D. $e^{-Ts} = \frac{1}{s + T}$

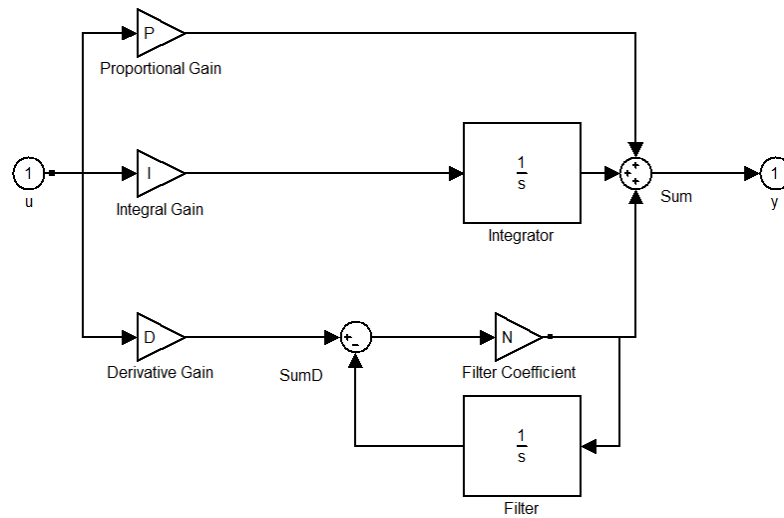
Câu 4: Sơ đồ sau là cấu trúc bên trong của khâu nào?



- A. Khối PID controller (2DOF)

- B. Khối quán tính
- C. Khối đạo hàm
- D. Khối tích phân

Câu 5: Sơ đồ sau là cấu trúc bên trong của khâu nào?



- A. Khối PID controller
- B. Khối quán tính
- C. Khối đạo hàm
- D. Khối tích phân

Bài tập cuối chương

Bài 1: Cho đối tượng có mô hình không gian trạng thái, với các ma trận A, B, C, D được cho như sau :

$$A = \begin{bmatrix} 0 & 0,3 \\ -0,5 & -0,2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$C = [0 \quad 1] \quad D = [0]$$

Xây dựng mô hình mô phỏng đối tượng trên simulink với tín hiệu vào có dạng Step

Bài 2: Cho đối tượng có mô hình không gian trạng thái, với các ma trận A, B, C, D được cho như sau :

$$A = \begin{bmatrix} 0 & 0,3 \\ -0,5 & -0,2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$C = [0 \quad 1] \quad D = [0]$$

Xây dựng mô hình mô phỏng đối tượng trên simulink với tín hiệu vào có dạng Constant.

Bài 3: Cho đối tượng có mô hình không gian trạng thái, với các ma trận A, B, C, D được cho như sau :

$$A = \begin{bmatrix} 0 & 0,3 \\ -0,5 & -0,2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$C = [0 \quad 1] \quad D = [0]$$

Xây dựng mô hình mô phỏng hệ thống điều khiển đối tượng trên sử dụng bộ điều khiển PID trong simulink với tín hiệu vào có dạng Step

Bài 4: Cho đối tượng có mô hình không gian trạng thái, với các ma trận A, B, C, D được cho như sau :

$$A = \begin{bmatrix} 0 & 0,3 \\ -0,5 & -0,2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$C = [0 \quad 1] \quad D = [0]$$

Xây dựng mô hình mô phỏng hệ thống điều khiển đối tượng trên sử dụng bộ điều khiển PID 2 bậc trong simulink với tín hiệu vào có dạng Step

Bài 5: Cho đối tượng có mô hình hàm truyền như sau :

$$W(s) = \frac{10s + 3}{s^2 + 3s + 1}$$

Xây dựng mô hình mô phỏng hệ thống điều khiển đối tượng trên sử dụng bộ điều khiển PID trong simulink với tín hiệu vào có dạng Step

Bài 6: Cho đối tượng có mô hình hàm truyền như sau :

$$W(s) = \frac{10s + 3}{s^2 + 3s + 1}$$

Xây dựng mô hình mô phỏng hệ thống điều khiển đối tượng trên sử dụng bộ điều khiển PID 2 bậc trong simulink với tín hiệu vào có dạng Step

Bài 7: Cho lò nhiệt điện trở có hàm truyền đạt như sau:

$$W(s) = \frac{K.e^{-T_2s}}{T_1s + 1}$$

Với K=150, T₁=3000, T₂=50. Sử dụng bộ điều khiển PID trong Matlab/Simulink để điều khiển đối tượng trên.

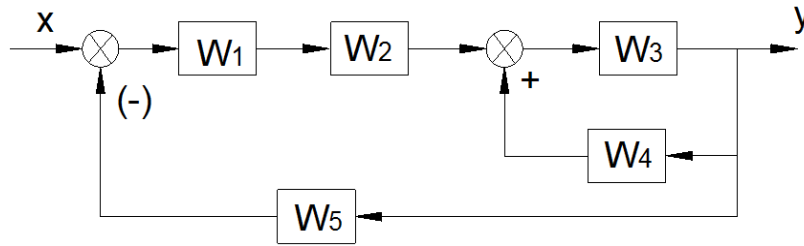
Bài 8: Cho hệ thống có sơ đồ khối như sau:

$$W1 = s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = \frac{10s + 3}{s^2 + 3s + 1}$$

$$W4 = 1/(s+2)$$

$$W5 = 4$$



Xác định đáp ứng của hệ thống trên với tín hiệu vào có dạng Constant.

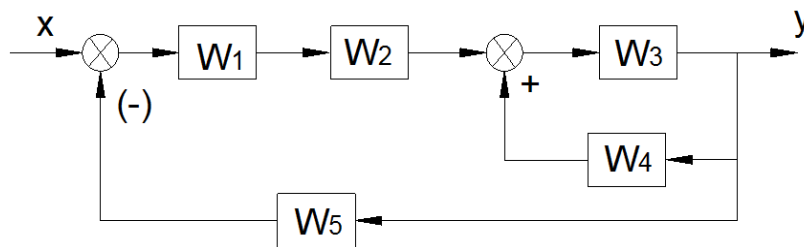
Bài 9: Cho hệ thống có sơ đồ khối như sau:

$$W1 = s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = \frac{10s + 3}{s^2 + 3s + 1}$$

$$W4 = 1/(s+2)$$

$$W5 = 4$$



Sử dụng bộ điều khiển PID trong Matlab/Simulink để điều khiển hệ thống trên.

Bài 10 : Cho lò nhiệt điện trở có hàm truyền đạt như sau:

$$W(s) = \frac{K.e^{-T_2s}}{T_1s + 1}$$

Với $K=150, T_1=3000, T_2=50$. Sử dụng bộ điều khiển PID 2 bậc trong Matlab/Simulink để điều khiển đối tượng trên.

Chương V: KIẾN THỨC CHUNG VỀ MÔ HÌNH HOÁ VÀ MÔ PHỎNG HỆ THỐNG

Nội dung chính của chương

Chương 5 giúp sinh viên nắm được những kiến thức chung về mô hình hóa và mô phỏng hệ thống.

Mục tiêu cần đạt được của chương

Sinh viên hiểu được các khái niệm và phân biệt các loại mô hình, đối tượng và hệ thống nắm được phương pháp thực hiện nghiên cứu một đối tượng hay hệ thống bằng phương pháp mô phỏng.

Bài 6: Kiến thức chung về mô hình hoá và mô phỏng hệ thống (số tiết: 3 tiết)

5.1. Mô hình hóa hệ thống

5.1.1. Khái niệm của mô hình hóa

Khái niệm mô hình hóa được phát biểu như sau:

“Mô hình hóa là phương pháp xây dựng mô hình toán học của hệ thống bằng cách dựa vào các quy luật vật lý chi phối hoạt động của hệ thống”.

Các phương pháp xây dựng mô hình của hệ thống chia ra làm hai loại:

Phương pháp lý thuyết;

Phương pháp thực nghiệm (nhận dạng).

Phương pháp lý thuyết là phương pháp thiết lập mô hình dựa trên các định luật có sẵn về quan hệ vật lý bên trong và quan hệ giao tiếp bên ngoài giữa hệ thống với môi trường bên ngoài của hệ thống. Các quan hệ được mô tả theo quy luật lý – hóa, quy luật cân bằng ... dưới dạng các phương trình toán học.

Mô hình hóa là thay thế đối tượng gốc bằng một mô hình để thay thế nhằm các thu nhận thông tin quan trọng về đối tượng bằng cách tiến hành các thực nghiệm trên mô hình. Lý thuyết xây dựng mô hình và nghiên cứu mô hình để hiểu biết về đối tượng gốc gọi lý thuyết mô hình hóa.

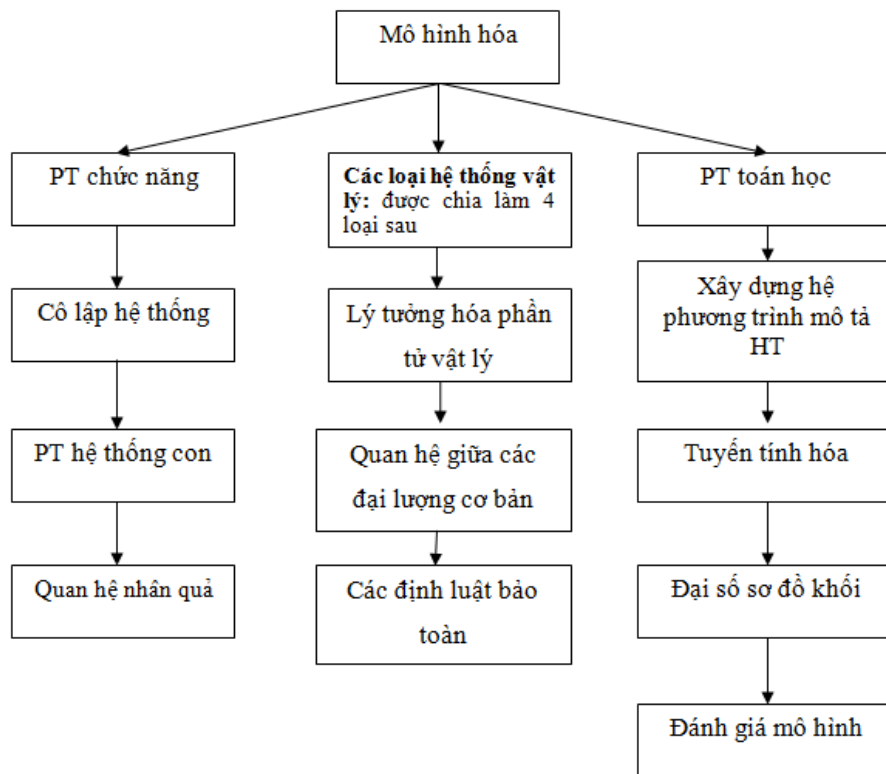
Vậy phương pháp mô hình hóa chính là phương pháp thuộc loại phương pháp lý thuyết và nó chỉ được áp dụng khi ta đã biết rõ cấu trúc của hệ thống và các quy luật vật lý chi phối hoạt động của hệ thống.

Trong một số trường hợp mà ở đó sự hiểu biết về các quy luật bên trong hệ thống và mối quan hệ giữa hệ thống với môi trường bên ngoài không được đầy đủ để có thể xây dựng được mô hình hoàn chỉnh, nhưng ít nhất nó có thể cho ta biết được các thông

tin ban đầu về dạng mô hình để khoanh vùng các mô hình thích hợp cho hệ thống. Để xây dựng được một mô hình hoàn chỉnh trong trường hợp này thì tiếp theo người ta áp dụng phương pháp thực nghiệm để hoàn thiện nốt quá trình xây dựng mô hình hệ thống.

Nếu các quá trình xảy ra trong mô hình đồng nhất (theo các chỉ tiêu định trước) với các quá trình xảy ra trong đối tượng gốc thì người ta nói rằng mô hình đồng nhất với đối tượng. Lúc này người ta có thể tiến hành các thực nghiệm trên mô hình để thu nhận thông tin về đối tượng. Lý thuyết cũng như thực nghiệm cũng đã chứng minh được rằng chúng ta chỉ xây dựng được mô hình gần đúng với đối tượng thực mà thôi, vì trong quá trình mô hình hóa đối tượng thực bao giờ chúng ta cũng phải chấp nhận một số giả thiết nhằm làm giảm bớt độ phức tạp của mô hình, để mô hình có thể ứng dụng thuận tiện trong thực tế. Mặc dù vậy, mô hình hóa vẫn là phương pháp hữu hiệu để con người nghiên cứu đối tượng, nhận biết các quá trình, các quy luật tự nhiên. Đặc biệt, ngày nay với sự trợ giúp đắc lực của khoa học kỹ thuật, nhất là khoa học máy tính và công nghệ thông tin, người ta đã phát triển các phương pháp mô hình hóa cho phép xây dựng các mô hình ngày càng gần với đối tượng nghiên cứu, đồng thời việc thu nhận, lựa chọn, xử lý các thông tin về mô hình rất thuận tiện, nhanh chóng và chính xác. Chính vì vậy, mô hình hóa là một phương pháp nghiên cứu khoa học mà tất cả những người làm khoa học, đặc biệt là các kỹ sư đều phải nghiên cứu và ứng dụng vào thực tiễn hoạt động của mình.

Quá trình mô hình hóa một hệ thống vật lý hay hệ thống thực tế được thực hiện qua một số bước như sau: phân tích chức năng, phân tích vật lý và phân tích toán học. Trong đó các bước mô hình hóa được mô tả cụ thể như sau:



Hình 5.1. Mô hình hóa

Bước 1: Phân tích chức năng

Phân tích chức năng là bước mà phân tích hệ thống thành các khối chức năng, trong đó mô hình toán học của các khối chức năng đã biết hoặc có thể rút ra được dựa vào các quá trình vật lý.

Bước 2: Phân tích vật lý

Phân tích vật lý là quá trình rút ra phương trình toán học (mô hình toán học) của các khối chức năng dựa vào các quá trình vật lý.

Bước 3: Phân tích toán học

Phân tích toán học chính là quá trình liên kết các khối chức năng lại với nhau thông qua các liên kết toán học, từ đó ta có thể đi xây dựng được mô hình của hệ thống.

Phương pháp mô hình hóa chỉ được áp dụng khi ta đã biết rõ cấu trúc của hệ thống và các quy luật vật lý chi phối tới hoạt động của hệ thống.

5.1.2. Vai trò của phương pháp mô hình hóa hệ thống

Mô hình hóa là một trong những phương pháp nghiên cứu khoa học đang phát triển và có triển vọng. Ở giai đoạn thiết kế, mô hình hóa giúp người thiết kế lựa chọn cấu trúc, các thông số của hệ thống để tổng hợp hệ thống. Ở giai đoạn chế tạo, mô hình

hóa giúp cho việc lựa chọn vật liệu và công nghệ chế tạo. Ở giai đoạn vận hành hệ thống, mô hình hóa giúp cho người điều khiển giải các bài toán điều khiển tối ưu, dự đoán các trạng thái của hệ thống ...

Vậy phương pháp mô hình hóa thường được dùng trong các trường hợp sau đây:

1/ Khi nghiên cứu trên hệ thống thực gặp nhiều khó khăn do nhiều nguyên nhân gây ra như sau:

- Giá thành nghiên cứu trên hệ thống thực quá đắt.

VD: Nghiên cứu kết cấu tối ưu, độ bền, khả năng chống dao động của ô tô, tàu thủy, máy bay,... người ta phải tác động vào đối tượng nghiên cứu các lực đủ lớn đến mức có thể phá hủy đối tượng để từ đó đánh giá các chỉ tiêu kỹ thuật đã đề ra. Như vậy, giá thành nghiên cứu sẽ rất đắt. Bằng cách mô hình hóa trên máy tính ta dễ dàng xác định được kết cấu tối ưu của các thiết bị nói trên.

- Nghiên cứu trên hệ thống thực đòi hỏi thời gian quá dài.

VD: Nghiên cứu đánh giá độ tin cậy, đánh giá tuổi thọ trung bình của hệ thống kỹ thuật (thông thường tuổi thọ trung bình của hệ thống kỹ thuật khoảng $30 \div 40$ năm), hoặc nghiên cứu quá trình phát triển dân số trong khoảng thời gian $20 \div 50$ năm,... Nếu chờ đợi quãng thời gian dài như vậy mới có kết quả nghiên cứu thì không còn tính thời sự nữa. Bằng cách mô phỏng hệ thống và cho “hệ thống” vận hành tương đương với khoảng thời gian nghiên cứu người ta có thể đánh giá được các chỉ tiêu kỹ thuật cần thiết của hệ thống.

- Nghiên cứu trên hệ thực ảnh hưởng đến sản xuất hoặc gây nguy hiểm cho người và thiết bị.

VD: Nghiên cứu quá trình cháy trong lò hơi của nhà máy nhiệt điện, trong lò luyện clanhke của nhà máy xi măng... người ta phải thay đổi chế độ cấp nhiên liệu (than, dầu), tăng giảm sản lượng gió cấp, thay đổi áp suất trong lò,... Việc làm các thí nghiệm như vậy sẽ cản trở việc sản xuất bình thường, trong nhiều trường hợp có thể xảy ra cháy, nổ gây nguy hiểm cho người và thiết bị. Bằng cách mô phỏng hệ thống, người ta có thể cho hệ thống “vận hành” với các bộ thông số, các chế độ vận hành khác nhau để tìm ra lời giải tối ưu.

- Trong một số trường hợp không cho phép làm thực nghiệm trên hệ thống thực.

VD: Nghiên cứu các hệ thống làm việc ở môi trường độc hại, nguy hiểm, dưới hầm sâu, dưới đáy biển, hoặc nghiên cứu trên cơ thể người,... Trong những trường hợp này dùng phương pháp mô phỏng là giải pháp duy nhất để nghiên cứu hệ thống.

2/ *Phương pháp mô hình hóa cho phép đánh giá độ nhạy của hệ thống* khi thay đổi tham số hoặc cấu trúc của hệ thống cũng như đánh giá phản ứng của hệ thống khi thay đổi tín hiệu điều khiển. Những số liệu này dùng để thiết kế hệ thống hoặc lựa chọn thông số tối ưu để vận hành hệ thống.

3/ *Phương pháp mô hình hóa cho phép nghiên cứu hệ thống ngay cả khi chưa có hệ thống thực.*

Trong trường hợp này, khi chưa có hệ thống thực thì việc nghiên cứu trên mô hình là giải pháp duy nhất để đánh giá các chỉ tiêu kỹ thuật của hệ thống, lựa chọn cấu trúc và thông số tối ưu của hệ thống... đồng thời mô hình cũng được dùng để đào tạo và huấn luyện.

Trong những trường hợp này dùng phương pháp mô phỏng mô hình hóa là giải pháp duy nhất để nghiên cứu hệ thống.

5.2. Đối tượng, hệ thống, mô hình toán học

5.2.1. Đối tượng của hệ thống

Khái niệm: Đối tượng (*object*) là bao gồm tất cả các phần tử có thông số cố định, là tất cả những sự vật, sự kiện mà hoạt động của con người có liên quan tới. Ví dụ như động cơ điện một chiều, động cơ xoay chiều (động cơ không đồng bộ, động cơ đồng bộ), máy biến áp...

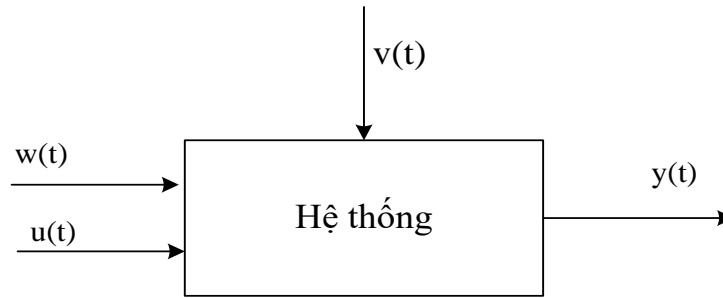
Mô hình toán học: là một hàm hợp thực chặt (đa thức chặt là đa thức có bậc của tử số nhỏ hơn bậc của mẫu số).

$$G(s) = K_0 \frac{a_m s^m + a_{m-1} s^{m-1} + \dots + a_1 s + a_0}{b_n s^n + b_{n-1} s^{n-1} + \dots + b_1 s + b_0} \quad (m \leq n)$$

5.2.2. Hệ thống

Hệ thống là tập hợp các đối tượng trong đó các biến có những mối quan hệ nhất định, chúng tương tác với nhau tạo thành tín hiệu quan sát được.

Ta xét mô hình sau:



Hình 5.2. Mô hình tổng quát hệ thống

Trong đó: $w(t)$ là nhiễu đo được

$u(t)$ là tín hiệu vào

$v(t)$ là nhiễu không đo được

$y(t)$ là tín hiệu ra

Từ ví dụ trên, ta thấy rằng các tín hiệu vào/ra quan hệ với nhau thông qua khối hệ thống. Hệ thống này có thể được mô tả bởi hàm truyền đạt hoặc phương trình trạng thái mà chúng ta đã được học trong môn lý thuyết điều khiển tự động.

Cấu trúc của một hệ thống được xét dựa trên hai phương diện: từ phía ngoài và từ phía trong. Từ phía ngoài tức là xét hệ thống bằng cách dựa vào các phần tử cấu thành lên hệ thống và mối quan hệ giữa chúng với nhau, đây gọi là *phương pháp tiếp cận cấu trúc*. Từ phía trong tức là phân tích hệ thống dựa trên các đặc tính chức năng của các phần tử cấu thành cho phép hệ thống đạt tới mục tiêu đã định trước, đó gọi là *phương pháp chức năng*.

Hệ thống được phân loại theo nhiều cách như là phân loại theo thời gian, theo tính chất, theo mục đích điều khiển.....Phân loại theo thời gian có hệ liên tục và hệ rời rạc; theo tính chất hệ thống có hệ thống động và hệ thống tĩnh, ngoài ra còn có hệ có tham số và hệ không tham số...

Theo tính chất của hệ thống chia thành 2 loại, đó là hệ thống động và hệ thống tĩnh. Trong đó các hệ thống được khái niệm như sau:

Hệ thống tĩnh là hệ thống mà tín hiệu ra chỉ phụ thuộc vào tín hiệu vào ở thời điểm hiện tại mà không phụ thuộc bất cứ tín hiệu nào khác.

Hệ thống động (Dynamic System) là hệ thống trong đó tín hiệu ra không chỉ phụ thuộc vào tín hiệu vào ở hiện tại mà còn phụ thuộc vào tín hiệu ra và tín hiệu vào trong quá khứ.

Như vậy, đối với hệ thống động thì tín hiệu ra không chỉ phụ thuộc vào tín hiệu vào mà còn phụ thuộc vào tín hiệu nhiễu hoặc tín hiệu vào ở quá khứ.

Như đã được nêu ở trên, các tín hiệu vào và ra của một hệ thống quan hệ với nhau, quan hệ vào ra của hệ thống có thể mô tả được bằng phương trình vi phân (hệ liên tục) hoặc phương trình sai phân (*hệ rời rạc*), hàm truyền đạt, phương trình trạng thái.

Đối với hệ thống liên tục theo thời gian, mối quan hệ giữa tín hiệu vào $u(t)$ và tín hiệu ra $y(t)$ được mô tả bởi phương trình vi phân:

$$\begin{aligned} b_n \frac{d^n y}{dt^n} + b_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + b_{n-2} \frac{d^{n-2} y}{dt^{n-2}} + \dots + b_1 \frac{d^1 y}{dt^1} \\ = a_m \frac{d^m x}{dt^m} + a_{m-1} \frac{d^{m-1} x}{dt^{m-1}} + \dots + a_1 \frac{dx}{dt} + a_0 \end{aligned}$$

Đối với hệ thống rời rạc theo thời gian, được mô tả bởi phương trình sai phân:

$$b_n y(t-n) + b_{n-1} y(t-n+1) + \dots + b_1 y(t) = a_m x(t-m) + a_{m-1} x(t-m+1) + \dots + a_1 x(t-1) + x(t)$$

Đối với một hệ thống thực, có hai phương pháp cơ bản để nghiên cứu thực nghiệm: Nghiên cứu trên hệ thực và nghiên cứu trên mô hình. Nếu nghiên cứu thực nghiệm trên hệ thực thì cho ta các số liệu khách quan, trung thực nhưng phải tiến hành các quá trình lấy mẫu thống kê, ước lượng tham số, xử lý dữ liệu,... Tuy nhiên, việc nghiên cứu một số hệ thống thực còn gặp khó khăn như là: giá thành nghiên cứu cao, đòi hỏi thời gian nghiên cứu dài, có khả năng gây nguy hiểm cho người và thiết bị khi nghiên cứu, ngoài ra một số trường hợp không cho phép nghiên cứu... Vì vậy, trong trường hợp này thì nghiên cứu thực nghiệm dựa trên các mô hình là phương pháp có nhiều triển vọng nhất và các hệ thống phổ biến hiện nay hệ thống động.

5.2.3. Mô hình, mô hình toán học

5.2.3.1. Mô hình của hệ thống

a. Khái niệm về mô hình (Model):

Mô hình của một hệ thống là quan hệ giả thiết (*assumed relationship*) giữa các tín hiệu quan sát được của hệ thống đó.

Nhìn chung các hệ thống thực có tính phức tạp hay các đối tượng thực có cấu trúc phức tạp mà thuộc loại hệ thống lớn thì mô hình của chúng cũng được liệt vào loại hệ thống lớn và có những đặc điểm cơ bản sau:

1. *Tính mục tiêu*: tùy theo yêu cầu nghiên cứu có thể có mô hình chỉ có một mục tiêu là để nghiên cứu một nhiệm vụ cụ thể nào đó hoặc mô hình đa mục tiêu nhằm khảo sát một số chức năng, đặc tính của đối tượng thực.

2. *Độ phức tạp*: Độ phức tạp thể hiện ở cấu trúc phân cấp của mô hình, các mối quan hệ qua lại của các hệ thống con với nhau và giữa hệ thống với môi trường làm việc của hệ thống.

3. *Hành vi của mô hình*: Hành vi của mô hình là con đường để mô hình đạt tới mục tiêu đề ra. Tùy thuộc vào hành vi của hệ thống, người ta có thể phân ra hệ thống liên tục hoặc gián đoạn. Khi nghiên cứu hành vi của mô hình người ta biết được xu thế vận động của đối tượng thực.

4. *Tính thích nghi*: Tính thích nghi chỉ có ở hệ thống cấp cao, hệ thống có thể thích nghi với sự thay đổi của các tác động vào hệ thống. Tính thích nghi thể hiện ở khả năng mô hình phản ánh được các tác động của môi trường tới hệ thống và khả năng giữ ổn định của mô hình khi các tác động thay đổi.

5. *Tính điều khiển được*: Ngày nay nhiều phương pháp tự động hóa đã được ứng dụng trong mô hình hóa hệ thống. Sử dụng các phương pháp lập trình người ta có thể điều khiển mô hình theo một mục tiêu định trước, thực hiện giao tiếp giữa người với mô hình để thu nhận thông tin và ra quyết định điều khiển.

6. *Khả năng phát triển của mô hình*: Khi tiến hành mô hình hóa hệ thống bao giờ cũng xuất hiện bài toán nghiên cứu sự phát triển của hệ thống trong tương lai. Vì vậy mô hình phải có khả năng mở rộng, thu nạp thêm các hệ thống con, có thể thay đổi cấu trúc cho phù hợp với sự phát triển của hệ thống thực.

7. *Độ tin cậy – Độ chính xác*: Mô hình hóa là thay thế đối tượng thực bằng mô hình của nó để thuận tiện trong việc nghiên cứu. Vì vậy mô hình phải phản ánh trung thực được các hiện tượng xảy ra trong đối tượng. Các kết quả thực nghiệm trên mô hình phải có độ tin cậy và độ chính xác thỏa mãn yêu cầu đề ra.

b. Phân loại mô hình hệ thống

Phân loại mô hình dựa vào nhiều yếu tố khác nhau. Có thể liệt kê từng cặp mô hình như sau:

Mô hình tiền định – Mô hình ngẫu nhiên

Mô hình tĩnh – Mô hình động

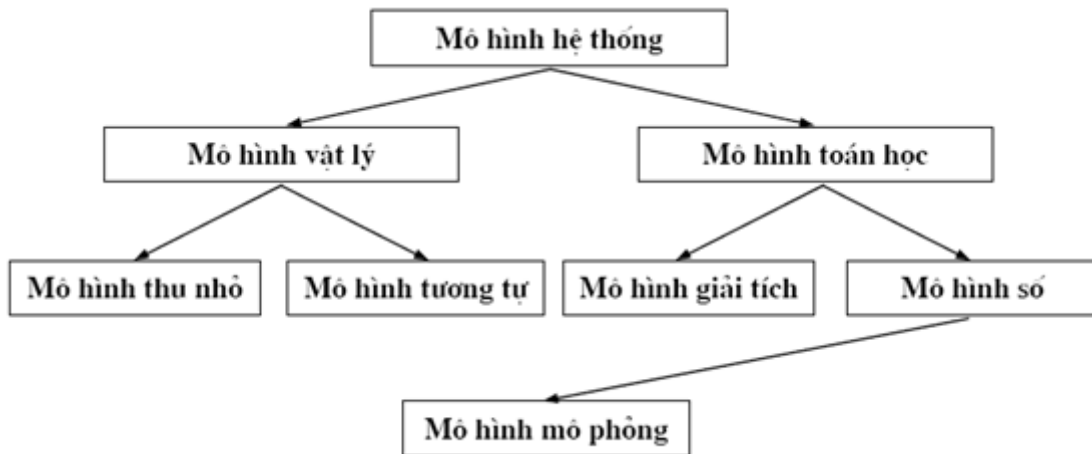
Mô hình tuyến tính – Mô hình phi tuyến

Mô hình liên tục - Mô hình gián đoạn

Mô hình vật lý – Mô hình toán học

Mô hình giải tích – Mô hình mô phỏng

Tuy nhiên, có thể chia các cặp mô hình trên theo hình 1.1. Theo cách này mô hình chia thành hai nhóm: mô hình vật lý và mô hình toán học hay còn gọi là mô hình trừu tượng.



Hình 5.3. Sơ đồ phân loại mô hình

Mô hình vật lý là mô hình được cấu tạo bởi các phần tử vật lý. Các thuộc tính của đối tượng phản ánh các định luật vật lý xảy ra trong mô hình. Nhóm mô hình vật lý được chia thành mô hình thu nhỏ và mô hình tương tự.

Mô hình vật lý thu nhỏ có cấu tạo giống đối tượng thực nhưng có kích thước nhỏ hơn cho phù hợp với điều kiện của phòng thí nghiệm. Ví dụ, người ta chế tạo lò hơi của nhà máy nhiệt điện có kích thước nhỏ đặt trong phòng thí nghiệm để nghiên cứu các chế độ thủy văn của đập thủy điện. Ưu điểm của loại mô hình này là các quá trình vật lý xảy ra trong mô hình giống như trong đối tượng thực, có thể đo lường quan sát các đại lượng vật lý một cách trực quan với độ chính xác cao. Nhược điểm của mô hình vật lý thu nhỏ là giá thành đắt, vì vậy chỉ sử dụng khi thực sự cần thiết.

Mô hình vật lý tương tự được cấu tạo bằng các phần tử vật lý không giống với đối tượng thực nhưng các quá trình xảy ra trong mô hình tương đương với quá trình xảy ra trong đối tượng thực. Ví dụ, có thể nghiên cứu quá trình dao động của con lắc đơn bằng mô hình tương tự là mạch dao động R-L-C vì quá trình dao động điều hòa trong mạch R-L-C hoàn toàn tương tự quá trình dao động điều hòa của con lắc đơn, hoặc người ta có thể nghiên cứu đường dây tải điện bằng mô hình tương tự là mạng bốn cực

R-L-C. Ưu điểm của loại mô hình này là giá thành rẻ, cho phép chúng ta nghiên cứu một số đặc tính chủ yếu của đối tượng thực.

Mô hình toán học thuộc loại mô hình trừu tượng. Các thuộc tính được phản ánh bằng các biểu thức, phương trình toán học. Mô hình toán học được chia thành mô hình giải tích và mô hình số. Mô hình giải tích được xây dựng bởi các biểu thức giải tích. Ưu điểm của loại mô hình là cho ta kết quả rõ ràng, tổng quát. Nhược điểm của mô hình giải tích là thường phải chấp nhận một số giả thiết đơn giản hóa để có thể biểu diễn đối tượng thực bằng các biểu thức giải tích, vì vậy loại mô hình này chủ yếu được dùng cho các hệ tiền định và tuyến tính.

Mô hình số được xây dựng theo phương pháp số tức là bằng các chương trình chạy trên máy tính số. Ngày nay, nhờ sự phát triển của kỹ thuật máy tính và công nghệ thông tin, người ta đã xây dựng được các mô hình số có thể mô phỏng được quá trình hoạt động của đối tượng thực. Những mô hình loại này được gọi là mô hình mô phỏng. Ưu điểm của mô hình mô phỏng là có thể mô tả các yếu tố ngẫu nhiên và tính phi tuyến của đối tượng thực, do đó mô hình càng gần với đối tượng thực. Ngày nay, mô hình mô phỏng được ứng dụng rất rộng rãi.

Có thể căn cứ vào các đặc tính khác nhau để phân loại mô hình như: mô hình tĩnh và mô hình động, mô hình tiền định và mô hình ngẫu nhiên, mô hình tuyến tính và mô hình phi tuyến, mô hình có thông số tập trung, mô hình có thông số dải, mô hình liên tục, mô hình gián đoạn, ...

c. Các tính chất cơ bản của mô hình

1. *Tính đồng nhất:* Mô hình phải đồng nhất với đối tượng mà nó phản ánh theo những tiêu chuẩn định trước.

2. *Tính thực dụng:* Có khả năng sử dụng mô hình để nghiên cứu đối tượng. Rõ ràng, để tăng tính đồng nhất trong mô hình phải đưa vào nhiều yếu tố phản ánh đầy đủ các mặt của đối tượng. Nhưng như vậy nhiều khi mô hình trở nên quá phức tạp và cồng kềnh đến nỗi không thể dùng để tính toán được nghĩa là mất đi tính chất thực dụng của mô hình. Nếu quá chú trọng tính thực dụng, xây dựng mô hình quá đơn giản thì sai lệch giữa mô hình và đối tượng thực sẽ lớn, điều đó sẽ dẫn đến kết quả nghiên cứu không chính xác. Vì vậy, tùy thuộc vào mục đích nghiên cứu mà người ta lựa chọn tính đồng nhất và tính thực dụng của mô hình một cách thích hợp.

Một số nguyên tắc khi xây dựng mô hình của hệ thống:

Việc xây dựng mô hình phụ thuộc vào đặc điểm của hệ thống thực, vì vậy khó có thể đưa ra nguyên tắc chặt chẽ, chỉ có thể đưa ra những nguyên tắc có tính định hướng cho việc xây dựng mô hình. Sau đây là một số nguyên tắc:

1. Nguyên tắc xây dựng sơ đồ khối

Nhìn chung hệ thống thực là một hệ thống lớn phức tạp, vì vậy tìm cách phân ra thành nhiều hệ thống nhỏ, mỗi hệ thống nhỏ đảm nhiệm một số chức năng của hệ thống lớn. Như vậy mỗi hệ thống con được biểu diễn bằng một khối, tín hiệu ra của khối trước chính là tín hiệu vào của khối sau.

2. Nguyên tắc thích hợp

Tùy theo mục đích nghiên cứu mà người ta lựa chọn một cách thích hợp giữa tính đồng nhất và tính thực dụng của mô hình. Chúng ta có thể giảm bớt một số chi tiết không quan trọng để làm giảm tính phức tạp và việc giải các bài toán trên mô hình dễ dàng hơn.

3. Nguyên tắc về tính chính xác

Yêu cầu về độ chính xác phụ thuộc vào mục đích nghiên cứu. Ở giai đoạn thiết kế tổng thể, độ chính xác không đòi hỏi cao nhưng khi thiết kế những phần cụ thể thì độ chính xác của mô hình phải đạt được yêu cầu đề ra cần thiết.

4. Nguyên tắc tổng hợp

Tùy theo mục đích nghiên cứu có thể phân chia hoặc tổ hợp các bộ phận của mô hình lại với nhau. Ví dụ khi mô hình hóa một phân xưởng để nghiên cứu quá trình sản xuất sản phẩm thì ta coi các máy móc là thực thể của nó. Khi nghiên cứu các quá trình điều khiển nhà máy thì ta tổ hợp phân xưởng như một thực thể của toàn nhà máy.

5.2.3.2. Mô hình toán học của hệ thống

a. Khái niệm mô hình toán học

Mô hình toán học là các biểu thức toán học mô tả quan hệ vào ra của hệ thống. Các mô hình toán học được xây dựng bằng các cách sau:

Mô hình hóa (*System Modeling*) hay gọi là phương pháp mô hình hộp trắng (*White-box model*).

Nhận dạng hệ thống (*System Identification*) hay gọi là phương pháp mô hình hộp đen (*Black-box model*).

Kết hợp mô hình hóa và nhận dạng hệ thống hay gọi là phương pháp mô hình hộp xám (*Gray-box model*).

b. Các loại mô hình toán học

Mô hình toán học bao gồm có các loại như sau:

Mô hình trong miền thời gian – Mô hình trong miền tần số

Mô hình liên tục – Mô hình rời rạc

Mô hình tuyến tính - Mô hình phi tuyến

Mô hình tham số - Mô hình không tham số...

Ứng dụng của mô hình: mô hình được ứng dụng trong rất nhiều các lĩnh vực khác nhau với các mục đích khác nhau. Một số các ứng dụng mà mô hình được sử dụng như là:

Thiết kế hệ thống

Mô phỏng

Dự báo

Đo lường

Phát hiện, chuẩn đoán lỗi

Tối ưu hóa

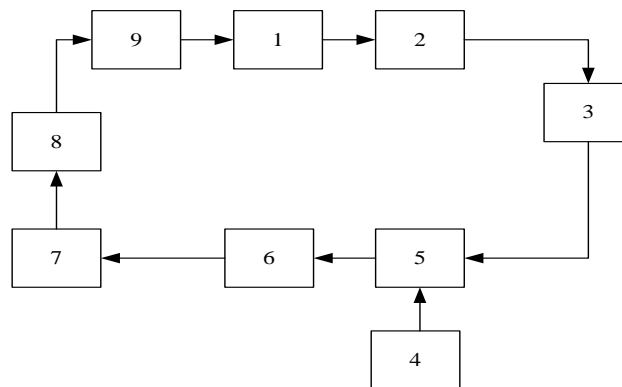
5.2.4. Cấu trúc hệ thống và mối liên hệ ngược

5.2.4.1. Cấu trúc hệ thống

Một hệ thống ĐKTD (điều khiển tự động) bao gồm nhiều phần tử khác nhau và được phân ở nhiều vị trí khác nhau, phụ thuộc vào yêu cầu của thiết bị công nghệ.

Cấu trúc của một hệ thống điều khiển cơ bản gồm có ba thành phần chính: bộ điều khiển, đối tượng điều khiển và khâu phản hồi.

Cấu trúc của một hệ thống điều khiển công nghiệp thông thường gồm có 9 thành phần cơ bản. Sau đây là sơ đồ khối của một hệ thống điều khiển trong công nghiệp thông thường:



Hình 5.4. Sơ đồ khối hệ thống điều khiển công nghiệp

Mô tả chức năng của từng khối:

Thiết bị công nghệ(1): có thể là không gian lò gia nhiệt, một buồng phản ứng, một ống dẫn và cũng có thể là cả một con tàu, buồng lò hơi nhà máy nhiệt điện, một băng tải trong dây chuyền...

Thiết bị đo lường (2): là thiết bị cảm biến dùng để đo các đại lượng vật lí, các đại lượng cần điều chỉnh có thể là một tín hiệu nhiệt độ, áp suất, lưu lượng,....

Bộ chuyển đổi (3): Thường là các bộ chuyển đổi tín hiệu không điện thành tín hiệu điện, tín hiệu điện thành tín hiệu điện hoặc ngược lại. Bộ (3) có thể là bộ chuyển đổi cộng hưởng tín hiệu, tín hiệu ra của nó thường là tín hiệu điện xoay chiều hoặc tín hiệu chuyển điện áp một chiều hoặc tín hiệu chuyển dòng điện: $0 \div 5 \text{ mA}$; $0 \div 20 \text{ mA}$; $4 \div 20 \text{ mA}$ hoặc tín hiệu khí nén chuẩn $0,2 \div 1 \text{ kg / cm}^2$. Nhìn chung tín hiệu ra của bộ (3) phải có cùng thứ nguyên với tín hiệu chủ đạo (4). Giá trị của đại lượng công nghệ cần điều chỉnh được đo bằng cảm biến (2) và được qua bộ chuyển đổi (3) để chuyển đổi tín hiệu đo thành dạng tín hiệu thuận lợi cho việc gia công. Bộ chuyển đổi tín hiệu (3) có thể có hoặc không, nếu tín hiệu ra của bộ (2) phù hợp với việc gia công thì ta không cần phải cho qua bộ (3) nữa.

Tín hiệu chủ đạo (4): Được đặt bằng tay do công nhân vận hành đặt trong quá trình sản xuất.

Thiết bị so sánh (5): Xác định giá trị sai lệch e giữa đại lượng chủ đạo x với đại lượng đo được y của đại lượng cần điều chỉnh: $e = x - y$.

Khối chức năng (6): Có đầu vào là tín hiệu e và đầu ra là tín hiệu u tác động lên cơ cấu chấp hành (8) thông qua bộ khuếch đại công suất (7).

Bộ khuếch đại công suất (7): Trong trường hợp công suất đầu ra không đủ đáp ứng yêu cầu của cơ cấu chấp hành (8) thì người ta dùng bộ khuếch đại công suất. Trường hợp công suất đầu ra u của bộ (6) đáp ứng được công suất của cơ cấu chấp hành thì bộ khuếch đại công suất không cần dùng tới.

Cơ cấu chấp hành (8): Thông thường các cơ cấu chấp hành là các loại động cơ điện như là động cơ điện một chiều, xoay chiều, động cơ servo hoặc các loại động cơ khí nén.

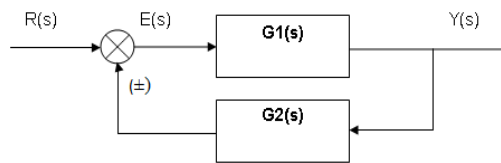
Cơ quan điều chỉnh (9): Thường là các van điều chỉnh dòng năng lượng hoặc vật chất cung cấp cho các thiết bị công nghệ.

Nhiệm vụ môn học là nghiên cứu dây chuyền công nghệ tổng thể, trong đó người kỹ sư thực hiện thiết kế khối (6).

5.2.4.2. Mối liên hệ ngược (phản hồi – Feedback)

Mối liên hệ là bộ phận cấu thành không thể thiếu được của một quá trình điều khiển.

Để điều khiển ổn định cho một đối tượng hay tối ưu cho một quá trình nào đấy thì ngoài việc gia công thông tin và năng lượng truyền tin (thông tin trực tiếp) thì ta phải phản hồi âm (-) hoặc phản hồi dương (+) ở đầu vào.



Hình 5.5. Mô hình phản hồi (feed back)

Ta có công thức như sau:

$$G(s) = \frac{G_1(s)}{1 \pm G_1(s).G_2(s)}$$

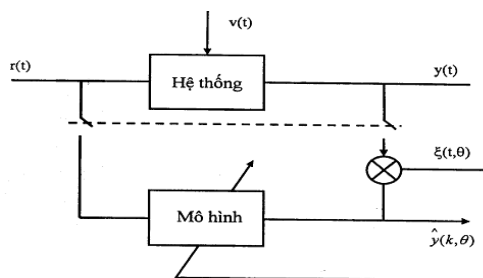
Trong đó: Trường hợp dấu (-) trong biểu thức trên là mạch phản hồi dương; trường hợp dấu (+) trong biểu thức trên là mạch phản hồi âm.

Nếu hàm truyền $G_2(s) = 1$ thì ta có hệ thống điều khiển vòng kín với phản hồi đơn vị. Tùy theo chiều phản hồi mang dấu (+) hay (-) mà ta có phản hồi dương đơn vị hoặc phản hồi âm đơn vị.

5.3. Nhận dạng hệ thống

a. Khái niệm: nhận dạng hệ thống là phương pháp xây dựng mô hình toán của hệ thống dựa vào dữ liệu vào ra quan sát được.

Mô hình như sau:



Hình 5.6. Mô hình tổng quát hệ thống nhận dạng

b. Các bước nhận dạng hệ thống:

Giống như nhiều công việc phát triển hệ thống khác, nhận dạng hầu như bao giờ cũng là một quá trình lặp. Những bước cơ bản trong xây dựng mô hình thực nghiệm cho một quá trình công nghiệp bao gồm các bước sau:

Bước 1: Thu thập, khai thác thông tin ban đầu về quá trình. Ví dụ các quá trình quan tâm, các phương trình mô hình từ phân tích lý thuyết, các điều kiện biên và các giả thiết liên quan.

Bước 2: Lựa chọn phương pháp nhận dạng (trực tuyến/ngoại tuyến, vòng hở hay vòng kín, chủ động/bị động). Thuật toán ước lượng tham số và tiêu chuẩn đánh giá mô hình.

Bước 3: Tiến hành lấy số liệu thực nghiệm cho từng cặp biến vào/ra trên cơ sở phương pháp nhận dạng đã chọn, xử lý thô các số liệu nhằm loại bỏ những giá trị đo kém tin cậy.

Bước 4: Kết hợp yêu cầu về mục đích sử dụng mô hình và khả năng ứng dụng của phương pháp nhận dạng đã chọn, quyết định về dạng mô hình (phi tuyến/ tuyến tính, liên tục/gián đoạn...) đưa ra giả thiết ban đầu về cấu trúc mô hình (bậc tử số/mẫu số của hàm truyền đạt, có hay không có trễ...)

Bước 5: Xác định tham số mô hình theo phương pháp/ thuật toán đã chọn. Nếu tiến hành theo từng mô hình con (ví dụ từng kênh vào/ra, từng khâu trong quá trình) thì sau đó kết hợp chúng lại thành một mô hình tổng thể.

Bước 6: Mô phỏng, kiểm chứng và đánh giá mô hình. Nhận được theo các tiêu chuẩn đã chọn, tốt nhất là trên nhiều tập dữ liệu khác nhau. Nếu chưa đạt yêu cầu thì quay lại một trong các bước 1-4.

Các kiến thức liên quan đến nhận dạng hệ thống:

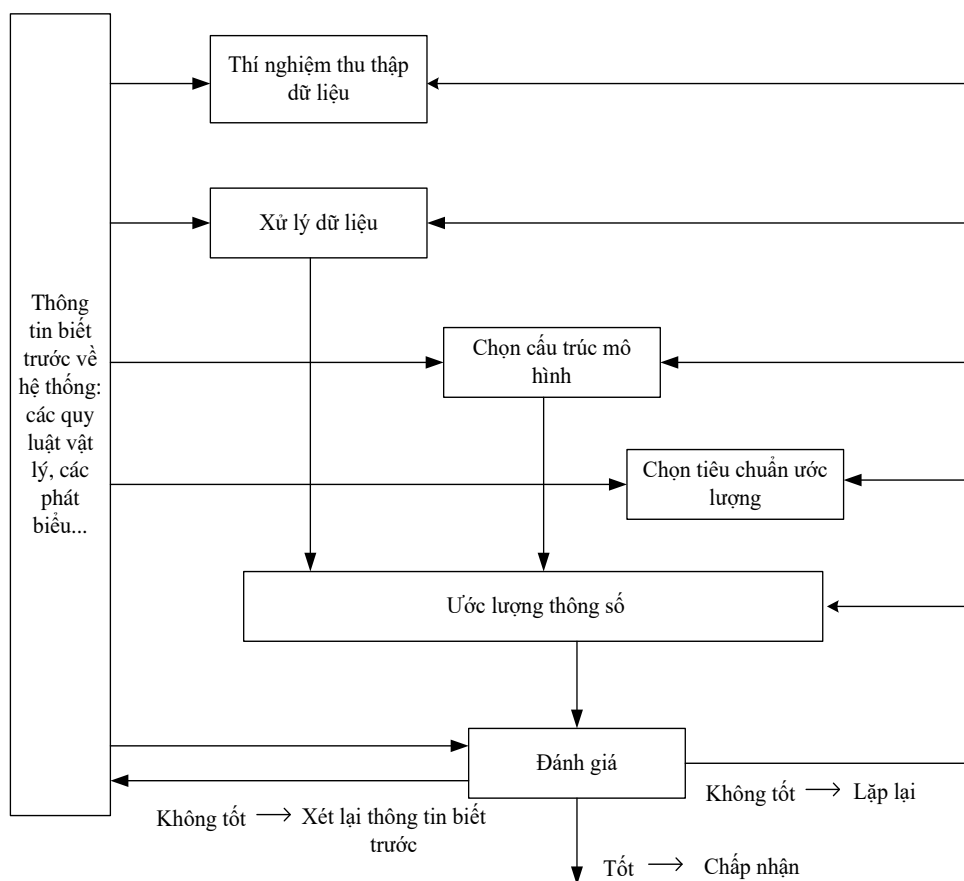
Lý thuyết tín hiệu (tín hiệu, nhiễu, năng lượng, công suất...).

Xác suất thống kê (xác suất, kỳ vọng, phương sai...).

Đại số tuyến tính (các phép toán trên ma trận).

Phương pháp tính (tối ưu hóa dùng phương pháp số).

- **Vòng lặp nhận dạng hệ thống được mô tả như sau:**



Hình 5.7. Quy trình nhận dạng hệ thống

5.4. Mô hình máy tính và phương pháp mô phỏng

5.4.1. Mô hình hoá hệ thống, mô hình máy tính và phương pháp mô phỏng

Khi có một mô hình toán học của một hệ thống thực ta có thể tìm các thông tin về hệ thống bằng nhiều cách. Trong trường hợp mô hình tương đối đơn giản ta có thể dùng phương pháp giải tích, ngược lại thì ta phải dùng phương pháp số.

Trước đây phương pháp giải tích tỏ ra hữu hiệu được dùng để mô hình hóa hệ thống. Tuy nhiên máy tính cũng đã giúp cho việc tính toán thuận lợi như tăng khối lượng tính toán, giảm thời gian tính...nhưng bản thân phương pháp giải tích cũng gặp nhiều khó khăn khi mô tả hệ thống như thường phải chấp nhận nhiều giả thiết để đơn giản hóa mô hình, do đó kết quả nghiên cứu có độ chính xác không cao.

Ngày nay bên cạnh phương pháp giải tích nói trên, phương pháp mô phỏng (*Simulation*) được phát triển mạnh và ứng dụng rất rộng rãi. Các mô hình được xây dựng dựa trên phương pháp mô phỏng được gọi là *mô hình mô phỏng* (hay còn gọi là *mô hình số*). Phương pháp mô phỏng cho phép đưa vào mô hình nhiều yếu tố sát với thực tế. Đồng thời mô hình giải trên máy tính cho tốc độ nhanh, dung lượng lớn, do đó kết quả

thu được có độ chính xác cao. Vì vậy phương pháp mô phỏng đã tạo điều kiện để giải các bài toán phức tạp như bài toán mô hình hóa các hệ thống lớn, hệ thống ngẫu nhiên, hệ thống phi tuyến có các thông số biến thiên theo thời gian.

Phương pháp mô phỏng đặc biệt hữu hiệu khi cần mô hình hóa các hệ thống lớn mà đặc điểm cơ bản của nó là có cấu trúc phân cấp, giữa các hệ thống con và hệ thống điều khiển có tác động qua lại với nhau. Ngoài ra nó cũng hữu hiệu khi mô phỏng các hệ thống có tính ngẫu nhiên, hệ thống phi tuyến.

Phương pháp mô phỏng được ứng dụng để mô hình hóa trong nhiều lĩnh vực khác nhau như: kỹ thuật, kinh tế, xã hội, ... Đặc biệt là các hệ thống lớn phức tạp, có nhiều yếu tố ngẫu nhiên tác động. Ngày nay nhờ sự phát triển của kỹ thuật máy tính và công nghệ thông tin mà phương pháp mô hình hóa và mô phỏng phát triển lên một mức cao, có khả năng:

Mô phỏng các hệ thống lớn phức tạp, đặc biệt là các hệ thống phi tuyến, ngẫu nhiên. Ứng dụng đồ họa kỹ thuật 3D, kỹ thuật tạo hình ảnh sinh động để xây dựng các chương trình mô phỏng sinh động trực quan rất thuận tiện cho việc nghiên cứu và hiển thị các kết quả mô phỏng.

Mô phỏng các hệ thống sản xuất.

Mô phỏng các hệ thống dịch vụ như trạm sửa chữa ô tô, phòng khám bệnh, nhà hàng...

Mô phỏng các hệ thống trò chơi được ứng dụng lĩnh vực giải trí, quân sự, kinh doanh...

Mô phỏng các hệ thống đào tạo như phòng thí nghiệm, lớp học điện tử (*E-learning*), phòng đào tạo lái xe,

Nhìn chung kỹ thuật mô hình hóa và mô phỏng ngày càng phát triển trong lĩnh vực khoa học kỹ thuật mà còn phát triển trong lĩnh vực y tế, kinh tế, giáo dục,....

5.4.2. Bản chất của phương pháp mô phỏng

Phương pháp mô phỏng được định nghĩa như sau:

“Mô phỏng là quá trình xây dựng mô hình toán học của hệ thống thực và sau đó tiến hành tính toán thực nghiệm trên mô hình để mô tả, giải thích và dự đoán hành vi của hệ thống thực”.

Theo định nghĩa này thì mô phỏng phải đạt được ba điểm cơ bản sau: Thứ nhất là phải có mô hình toán học tốt tức là mô hình phải có tính đồng nhất cao với hệ thực đồng thời mô hình được mô tả rõ ràng thuận tiện cho người sử dụng. Thứ hai là có khả năng làm thực nghiệm trên mô hình tức là có khả năng thực hiện các chương trình máy

tính để xác định các thông tin về hệ thực. Cuối cùng là khả năng dự đoán hành vi của hệ thực tức là có thể mô tả sự phát triển của hệ thực theo thời gian.

Phương pháp mô phỏng được đề xuất vào những năm 80 của thế kỷ 20, từ đó đến nay phương pháp mô phỏng đã được nghiên cứu, hoàn thiện, và ứng dụng thành công vào nhiều lĩnh vực khác nhau như lĩnh vực khoa học kỹ thuật, khoa học xã hội, kinh tế, y tế,...

Bản chất của phương pháp mô phỏng chính là xây dựng một mô hình số (*Model Numerically*), tức là xây dựng mô hình thể hiện được bằng các chương trình máy tính, sau đó tiến hành các “*thực nghiệm*” trên mô hình để tìm ra các đặc tính của hệ thống được mô phỏng. Số lần thực nghiệm về lý thuyết được tăng lên vô cùng lớn. Quá trình mô hình hóa được tiến hành như sau: Gọi hệ thống được mô phỏng là S.

Bước thứ nhất: Mô hình hóa hệ thống S với các mối quan hệ nội tại, thông thường bằng cách chia hệ S thành nhiều hệ con theo các tiêu chí nào đó $S = \{S_1, S_2, S_3, \dots, S_n\}$. Tiếp đến mô tả toán học các hệ con cùng các quan hệ giữa chúng, giữa các hệ con có mối quan hệ trao đổi năng lượng và trao đổi thông tin.

Bước thứ hai: Mô hình hóa môi trường xung quanh E, nơi hệ thống S làm việc, với các mối quan hệ tác động qua lại giữa S và E. Khi đã có mô hình của S và E được tiến hành các thực nghiệm trên mô hình, tức là cho S và E làm việc ở một điều kiện xác định nào đó. Kết quả thu được một bộ thông số của hệ thống, hay thường gọi là xác định được một điểm làm việc của hệ thống. Các thực nghiệm đó được lặp lại nhiều lần và kết quả mô phỏng được đánh giá theo xác suất thống kê. Kết quả mô phỏng càng chính xác nếu số lần thực nghiệm càng lớn. Về lý thuyết bước mô phỏng là hữu hạn nhưng phải đủ lớn và phụ thuộc vào yêu cầu của độ chính xác.

Một số lĩnh vực mà phương pháp mô phỏng đã được ứng dụng trong thực tế:

Phân tích và thiết kế hệ thống sản xuất, lập kế hoạch sản xuất.

Đánh giá phần cứng, phần mềm của hệ thống máy tính.

Quản lý và xác định chính sách dự trữ mua sắm vật tư của hệ thống kho vật tư, nguyên liệu.

Phân tích và đánh giá hệ thống phòng thủ quân sự, xác định chiến lược phòng thủ, tấn công.

Phân tích và thiết kế hệ thống thông tin liên lạc, đánh giá khả năng làm việc của mạng thông tin.

Phân tích và thiết kế các hệ thống giao thông như đường sắt, đường bộ, đường hàng không, cảng biển.

Đánh giá, phân tích và thiết kế các cơ sở dịch vụ như bệnh viện, bưu điện, nhà hàng, siêu thị.

Phân tích hệ thống kinh tế, tài chính.

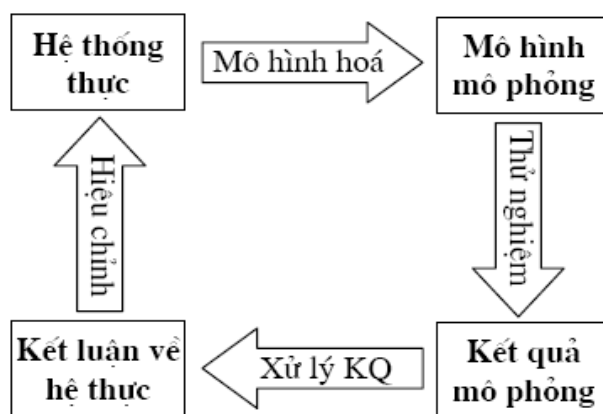
Phương pháp mô phỏng được ứng dụng vào các giai đoạn khác nhau của việc nghiên cứu, thiết kế và vận hành các hệ thống như sau:

a. Phương pháp mô phỏng được ứng dụng vào giai đoạn nghiên cứu, khảo sát hệ thống trước khi tiến hành thiết kế nhằm xác định độ nhạy của hệ thống đối với sự thay đổi cấu trúc và tham số của hệ thống.

b. Phương pháp mô phỏng được ứng dụng vào giai đoạn thiết kế hệ thống để phân tích và tổng hợp các phương án thiết kế hệ thống, lựa chọn cấu trúc hệ thống thỏa mãn các chỉ tiêu cho trước.

c. Phương pháp mô phỏng được ứng dụng vào giai đoạn vận hành hệ thống để đánh giá khả năng hoạt động, giải bài toán vận hành tối ưu, chuẩn đoán các trạng thái đặc biệt của hệ thống.

Quá trình nghiên cứu bằng phương pháp mô phỏng và quan hệ giữa hệ thống thực với kết quả mô phỏng như hình 5.8.

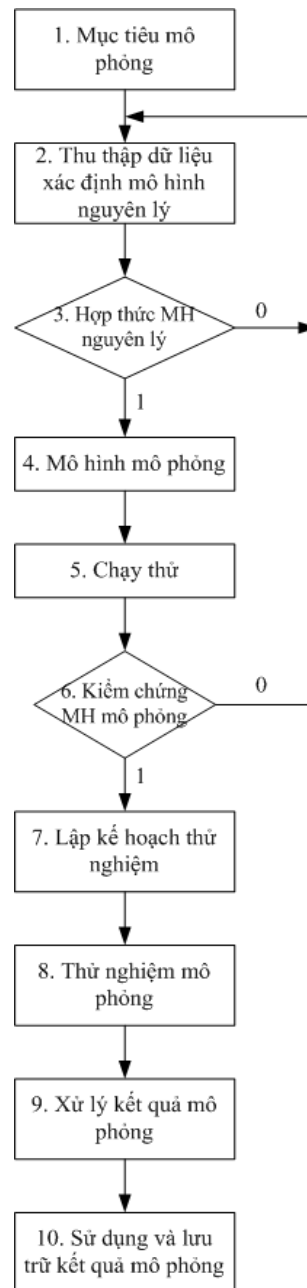


Hình 5.8. Quá trình nghiên cứu bằng phương pháp mô phỏng

Từ hình 5.8, để nghiên cứu hệ thống thực phải tiến hành mô hình hóa tức là xây dựng mô hình mô phỏng; khi có mô hình mô phỏng sẽ tiến hành các thực nghiệm để thu được các kết quả mô phỏng. Thông thường các kết quả mô phỏng có tính trừu tượng nên phải thông qua quá trình xử lý kết quả mô phỏng thì chúng ta mới thu được các thông

tin, từ đó đi kết luận về hệ thống thực. Sau đó dùng các thông tin, các kết luận trên để hiệu chỉnh hệ thống thực theo mục đích nghiên cứu đề ra.

5.4.3. Các bước mô phỏng hệ thống



Hình 5.9. Các bước mô phỏng hệ thống

Mô phỏng hệ thống thực hiện qua 10 bước cơ bản sau:

Bước 1: Xây dựng mục tiêu mô phỏng và kế hoạch nghiên cứu

Điều quan trọng trước khi mô phỏng đó là phải xác định rõ mục tiêu nghiên cứu mô phỏng.

Bước 2: Thu thập dữ liệu và xác định mô hình nguyên lý

Tùy theo mục đích nghiên cứu mà người ta thu thập thông tin cần thiết, các dữ liệu của hệ thống S và môi trường E. Trên cơ sở đó đi xây dựng mô hình nguyên lý M_{nl} , mô hình nguyên lý là mô hình toán học phản ánh bản chất của hệ thống S.

Bước 3: Hợp thức hóa mô hình nguyên lý M_{nl}

Hợp thức hóa mô hình nguyên lý là kiểm tra tính đúng đắn, hợp lý của mô hình. Mô hình nguyên lý phải phản ánh đúng bản chất của hệ thống S và môi trường E nhưng đồng thời cũng phải tiện dụng, không quá phức tạp cồng kềnh. Nếu mô hình nguyên lý M_{nl} không đạt phải thu thập thêm thông tin, dữ liệu để tiến hành xây dựng lại mô hình.

Bước 4: Xây dựng mô hình mô phỏng M_{mp} trên máy tính

Mô hình mô phỏng M_{mp} là những chương trình chạy trên máy tính. Các chương trình này được viết bằng các ngôn ngữ thông dụng như FORTRAN, PASCAL, C++, hoặc các ngôn ngữ chuyên dụng để mô phỏng như GPSS, SIMSCRIPT,...

Bước 5: Chạy thử

Sau khi cài đặt chương trình, người ta tiến hành chạy thử xem mô hình mô phỏng có phản ánh đúng các đặc tính của hệ thống S và môi trường E hay không. Ở giai đoạn này cũng tiến hành sửa chữa các lỗi về lập trình.

Bước 6: Kiểm chứng mô hình

Sau khi chạy thử người ta có thể kiểm chứng và đánh giá mô hình mô phỏng có đạt yêu cầu hay không, nếu không phải quay lại từ bước 2.

Bước 7: Lập kế hoạch thử nghiệm

Ở bước này phải xác định số lần thử nghiệm, thời gian mô phỏng của từng bộ phận hoặc toàn bộ mô hình. Căn cứ vào kết quả mô phỏng (ở bước 9), tiến hành hiệu chỉnh kế hoạch thử nghiệm để đạt được kết quả với độ chính xác theo yêu cầu.

Bước 8: Thử nghiệm mô phỏng

Cho chương trình chạy thử nghiệm theo kế hoạch đã được lập ở bước 7. Đây là bước thực hiện việc mô phỏng, các kết quả lấy ra từ bước này.

Bước 9: Xử lý kết quả

Thử nghiệm mô phỏng thường cho nhiều dữ liệu có tính thống kê xác suất. Vì vậy, để có kết quả cuối cùng với độ chính xác theo yêu cầu, cần phải thực hiện việc xử

lý các kết quả trung gian. Bước xử lý kết quả đóng vai trò quan trọng trong quá trình mô phỏng.

• **Bước 10:** Sử dụng và lưu trữ kết quả

Sử dụng kết quả mô phỏng vào mục đích đã định và lưu giữ dưới dạng các tài liệu để có thể sử dụng nhiều lần.

5.4.4 Một số môi trường mô phỏng thường gặp

- Matrix/ System Build - LabView - Pascal
- Easy 5 - VisSim
- Matlab/ Simulink - Tutsim

Bài tập cuối chương

Bài 1. Cho một số ví dụ về mô hình vật lý tương tự và mô hình vật lý thu nhỏ?

Bài 2. Một số khó khăn thường gặp khi tiến hành nghiên cứu trên các mô hình thực tế là gì? Cho ví dụ minh họa.

Bài 3. Các bước tiến hành mô phỏng hệ thống? Có mấy loại mô hình?

Bài 4. Nêu tầm quan trọng của mô hình hóa? Tại sao phải mô hình hóa hệ thống?

Bài 5. Trình bày bản chất và phương pháp nghiên cứu bằng mô phỏng cho ví dụ minh họa?

Bài 6. Khó khăn khi tiến hành nghiên cứu trên các hệ thống thực sau là gì?

Hệ thống lò hơi trong nhà máy nhiệt điện

Hệ thống tháp chưng cất

Hệ thống đường dây tải điện cao áp

Bài 7: Khi nghiên cứu trên hệ thống thực gặp nhiều khó khăn nào cho ví dụ cụ thể?

Bài 8: Trình bày vai trò của phương pháp mô hình hóa hệ thống.

Bài 9: Nêu đặc điểm và phân loại của mô hình vật lý và mô hình toán học. Loại mô hình nào được sử dụng để mô phỏng vì sao?

Bài 10: Khái niệm về nhận dạng hệ thống trình bày các bước tiến hành nhận dạng hệ thống.

Bài 11: Trình bày các bước khi tiến hành mô phỏng hệ thống.

Câu hỏi trắc nghiệm

Câu 1: Quá trình mô hình hóa một hệ thống vật lý hay hệ thống thực tế được thực hiện qua các bước nào?

- a/ Phân tích chức năng
- b/ Phân tích vật lý
- c/ Phân tích toán học
- d/ Cả 3 phương án trên đều đúng

Câu 2: Các loại mô hình nào sau đây không phải là mô hình toán học?

- a/ Mô hình trong miền thời gian – Mô hình trong miền tần số
- b/ Mô hình liên tục – Mô hình rời rạc
- c/ Mô hình tham số - Mô hình không tham số...
- d/ Mô hình ký tự

Câu 3: Cấu trúc của một hệ thống điều khiển cơ bản gồm ?

- a/ Bộ điều khiển
- b/ Đối tượng điều khiển
- c/ Khâu phản hồi
- d/ Cả 3 phương án trên đều đúng

Câu 4: Các kiến thức nào liên quan đến nhận dạng hệ thống?

- a/ Lý thuyết tín hiệu, xác suất thống kê, đại số tuyến tính, phương pháp tính
- b/ Lý thuyết matlab, xác suất thống kê, đại số tuyến tính, phương pháp tính
- c/ Lý thuyết tín hiệu, xác suất vòng lặp, đại số tuyến tính, phương pháp tính
- d/ Lý thuyết tín hiệu, xác suất thống kê, đại số tổ hợp, phương pháp tính

Câu 5: Có mấy bước khi tiến hành mô phỏng hệ thống?

- a/ 10 bước
- b/ 8 bước
- c/ bước
- d/ 12 bước

Chương VI: MÔ TẢ TOÁN HỌC CHO CÁC ĐỐI TƯỢNG VÀ HỆ THỐNG

Nội dung chính của chương

Chương 6 giúp sinh viên nắm được phương pháp mô tả toán học cho đối tượng và hệ thống.

Mục tiêu cần đạt được của chương

Sinh viên hiểu được các khái niệm và phương pháp xây dựng mô tả toán học cho đối tượng và hệ thống. Áp dụng xây dựng mô tả toán học cho một số đối tượng cụ thể.

Bài 7: Mô tả toán học cho các đối tượng và hệ thống (số tiết: 3 tiết)

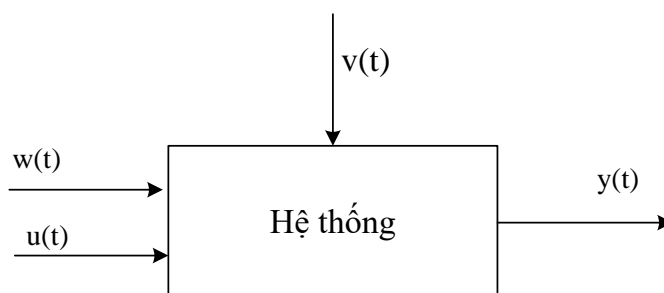
6.1. Hệ thống và các đặc tính của hệ thống

6.1.1. Hệ thống và hệ thống theo thời gian

Hệ thống được định nghĩa khái quát như sau:

“Hệ thống là tập hợp các đối tượng trong đó các biến có những mối quan hệ nhất định, chúng tương tác với nhau tạo thành tín hiệu quan sát được”.

Ta xét mô hình sau:



Hình 6.1. Mô hình tổng quát hệ thống

Trong đó: $w(t)$ là nhiễu đo được

$u(t)$ là tín hiệu vào

$v(t)$ là nhiễu không đo được

$y(t)$ là tín hiệu ra

Khi hệ thống chỉ có một tín hiệu vào và một tín hiệu ra thì hệ thống đó là hệ SISO và mô hình T của hệ thống là mô hình SISO (*Single Input Single Output*). Ngược lại hệ thống có nhiều đầu vào và nhiều đầu ra tương ứng sẽ có mô hình T là loại MIMO (*Multi Input –Multi Output*).

Có rất nhiều cách để phân loại hệ thống, có thể phân loại hệ thống theo thời gian, theo chức năng, theo tính chất động học của hệ thống ...

Một hệ thống được gọi là tuyến tính khi mô hình toán học của hệ thống là ánh xạ tuyến tính tức là tín hiệu vào $u(t)$ của hệ thống thỏa mãn nguyên lý xếp chồng. Nhờ có tính chất này mà việc phân tích hệ thống và khảo sát hệ thống trở nên đơn giản hơn. Để có được những tính chất của hệ thống tuyến tính trên cơ sở phân tích mô hình toán học của nó thì ta chỉ cần phân tích, khảo sát các đáp ứng của hệ thống với một vài kích thích điển hình, ví dụ như tín hiệu kích thích là xung dirac $\delta(t)$ hoặc xung đơn vị $1(t)$. Như vậy từ tính chất của hệ thống tuyến tính có thể suy ra được tính chất của một hệ thống phi tuyến như sau: Hệ phi tuyến là hệ mà trong đó các tín hiệu không thỏa mãn tính chất xếp chồng.

Phân loại hệ thống theo thời gian, ta có hệ thống liên tục theo thời gian và hệ thống gián đoạn theo thời gian (hệ thống rời rạc). Tùy vào dạng tín hiệu là liên tục hay rời rạc mà mô hình T của hệ thống là mô hình liên tục hay mô hình rời rạc.

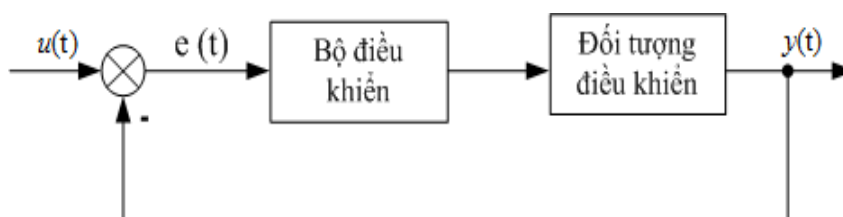
Phân loại hệ thống theo đặc tính làm việc thì người ta phân ra thành hệ thống động và hệ thống tĩnh. Các hệ thống này chúng ta sẽ được tìm hiểu trong mục 2.2.

Để đánh giá được chất lượng hệ thống làm việc tốt hay xấu thì thông thường người ta đi xét hệ thống về mặt thời gian. Vậy để biết được một hệ thống liên tục theo thời gian và hệ thống rời rạc theo thời gian là như thế nào thì chúng ta đi vào khái niệm về hệ liên tục và hệ rời rạc.

*Khái niệm hệ thống liên tục:

“Hệ thống liên tục là hệ có tất cả các tín hiệu tồn tại trong hệ đều liên tục theo thời gian”.

Cấu trúc tổng quát của hệ liên tục như hình dưới đây:

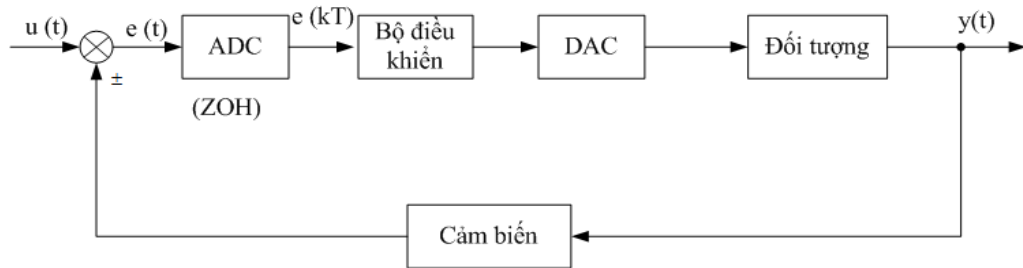


Hình 6.2. Cấu trúc của hệ thống liên tục

*Với Hệ thống gián đoạn:

“Một hệ thống gọi là gián đoạn khi một trong các tín hiệu tồn tại trong hệ thống không liên tục theo thời gian”.

Cấu trúc tổng quát của hệ gián đoạn.



Hình 6.3. Cấu trúc của hệ thống gián đoạn

Vậy hệ thống mà có tất cả các khâu liên tục thì hệ thống đó gọi là hệ thống liên tục, còn hệ thống nào có một khâu động học hoạt động rời rạc hóa thì gọi là hệ rời rạc. Hiện nay, trong công nghiệp thường sử dụng các hệ thống rời rạc vì nó chịu ảnh hưởng của nhiễu ít, nhưng không vì thế mà các hệ thống liên tục không được sử dụng mà các hệ thống liên tục vẫn được sử dụng song song với các hệ thống gián đoạn trong một số lĩnh vực.

Các hệ động lực học hay hệ thống vật lý trên thực tế bao gồm rất nhiều loại, thông thường người ta chia thành 4 loại hệ động lực cơ bản sau:

Hệ thống điện

Hệ thống cơ

Hệ thống nhiệt

Hệ thống lưu chất

Một hệ thống phức tạp có thể gồm nhiều hệ thống con thuộc 4 loại trên. Các phần tử cơ bản của hệ thống gồm:

Phân tử trở

Phân tử dung

Phân tử cảm hay quán tính

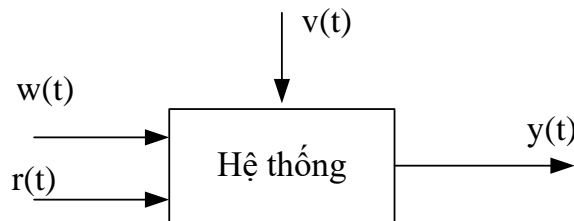
Loại hệ thống	Biến		
	Lượng	Thế	Thời gian
Điện	Điện tích	Điện thế	Giây
Cơ khí	Khoảng cách	Lực	Giây
Lưu chất	Thể tích	Áp suất	Giây

Nhiệt	Nhiệt năng	Nhiệt độ	Giây
-------	------------	----------	------

Chúng ta biết rằng cấu tạo nên hệ thống là các phần tử vật lý mà trên thực tế để xây dựng được các mô hình toán học của hệ thống hay là mô hình hóa hệ thống thì chúng ta phải đơn giản hóa hệ thống hay chính là lý tưởng hóa các phần tử vật lý bên trong hệ thống. Các phần tử vật lý cấu thành nên hệ thống thông thường được lý tưởng hóa theo các nguyên tắc như sau: nguyên tắc thuần hóa, nguyên tắc tập trung hóa, nguyên tắc lý tưởng hóa. Như vậy lý tưởng hóa các phần tử vật lý của hệ thống cũng chính là mô hình hóa các phần tử vật lý đó.

6.1.2. Vấn đề tĩnh và động của hệ thống

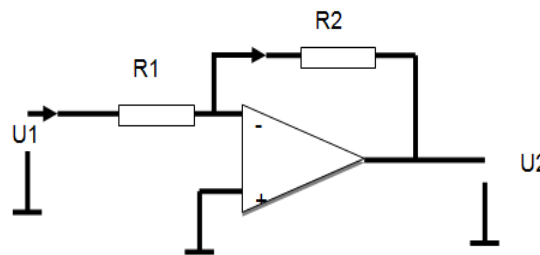
Một hệ thống được gọi là tĩnh khi tín hiệu ra chỉ phụ thuộc vào tín hiệu vào mà không phụ thuộc vào tín hiệu nào khác.



Hình 6.4. Mô hình của hệ thống

- Phương trình mô tả quan hệ: $y = f(r, w, v)$

Ví dụ 6.1: Xét mạch khuếch đại như sau:



Hình 6.5. Minh họa cho ví dụ 6.1 - hệ thống tĩnh/động

Ta có:

$$\dot{U}_1(s) = \dot{I}_1(s) \cdot R_1$$

$$\dot{U}_2(s) = -\dot{I}_2(s) \cdot R_2$$

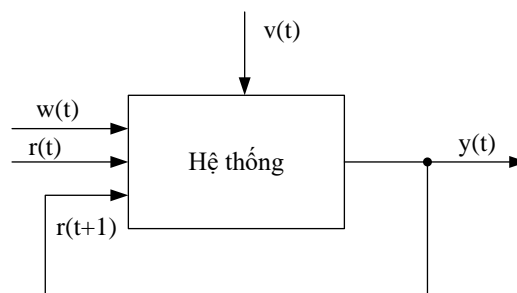
$$\rightarrow G(s) = \frac{U_2(s)}{U_1(s)} = \frac{-I_2(s) \cdot R_2}{I_1(s) \cdot R_1} = \frac{-R_2}{R_1}$$

Ta thấy, tỷ số giữa tín hiệu ra và tín hiệu vào là hằng số vì thế tín hiệu đầu ra chỉ phụ thuộc vào tín hiệu vào. Vậy hệ thống trên là hệ thống tĩnh.

Khái niệm về hệ thống động:

“Một hệ thống gọi là động khi tín hiệu ra không chỉ phụ thuộc vào tín hiệu vào ở hiện tại mà còn phụ thuộc vào tín hiệu vào, tín hiệu ra trong quá khứ”.

- Mô hình tổng quát của hệ thống động như sau:



Hình 6.6. Mô hình tổng quát của hệ thống động

Phương trình mô tả mối quan hệ: $y = f[r(t), r(t+1), w(t), v(t)]$

6.1.3. Vấn đề trễ

Các đối tượng điều khiển có trễ trong công nghiệp thường gặp nhiều trong các lĩnh vực công nghiệp, quân sự, giao thông... Thông thường khi thiết kế bộ điều khiển thường xấp xỉ chúng hoặc có thể dùng các bộ điều khiển dự báo theo các tiêu chuẩn nhất định.

Phương trình mô tả: $y(t) = r(t-\tau)$

Theo phương trình trên, ta có các phương trình xấp xỉ như sau:

1/ Phương trình xấp xỉ theo phương pháp Maclorank :

$$e^{-Ts} = \frac{1}{Ts+1}$$

6.1.4. Vấn đề tuyến tính hóa

Hầu hết mô hình toán học xây dựng bằng phương pháp lý thuyết cho các quá trình thực đều chứa phương trình vi phân phi tuyến. Nhưng cho đến nay, đa số phương

pháp phân tích và thiết kế điều khiển đều dựa trên mô hình tuyến tính. Ngay cả một số phương pháp phi tuyến cũng không loại trừ hoàn toàn sử dụng mô hình tuyến tính, ít ra là để làm cơ sở so sánh và kiểm chứng chất lượng. Vì thế nếu mục đích sử dụng mô hình là sử dụng cho mục đích phân tích hệ thống, thiết kế sách lược và thuật toán điều khiển, thì việc tuyến tính hóa mô hình trước hay sau cũng cần thiết. Có ba phương pháp tuyến tính hóa cơ bản hay được sử dụng là:

- Tuyến tính hóa xung quanh điểm làm việc (phải là một điểm cân bằng) áp dụng phép khai triển Taylor, kết quả là một mô hình tuyến tính xấp xỉ có giá trị sử dụng tại lân cận điểm làm việc.

- Tuyến tính hóa thông qua phép biến đổi đơn thuần, kết quả có thể là một mô hình tuyến tính hoặc một mô hình “ít phi tuyến” hơn, nhưng hoàn toàn tương đương với mô hình ban đầu.

- Tuyến tính hóa chính xác sử dụng phản hồi, kết quả là một mô hình mở rộng tuyến tính.

Trong ba phương pháp trên thì tuyến tính hóa xung quanh điểm làm việc áp dụng phép khai triển Taylor là có khả năng ứng dụng đa năng và đơn giản nhất. Đối với các hệ thống điều khiển quá trình, điểm làm việc ít thay đổi hoặc thay đổi rất chậm, vì vậy phép khai triển Taylor phù hợp cho đại đa số ứng dụng. Trong trường hợp hệ thống có độ phi tuyến lớn, ta có hai cách giải quyết đơn giản:

- Áp dụng một số phép đổi biến quen thuộc (ví dụ lấy tỉ lệ, tích, lũy thừa...) nhằm giảm độ phi tuyến, trước khi áp dụng tuyến tính hóa với phép khai triển Taylor.

- Chia dài làm việc của hệ thống thành nhiều phạm vi nhỏ và thực hiện tuyến tính hóa xung quanh các điểm làm việc tương ứng.

Mô hình của các quá trình công nghiệp mà ta quan tâm thông thường chỉ bao gồm phương trình vi phân cấp 1 và phương trình đại số, nên những biến trạng thái được chọn cũng chính là các biến ra cần điều khiển. Việc tuyến tính hóa sẽ đơn giản hơn nếu tiến hành với từng phương trình mô hình, đặc biệt khi một số phương trình đã tuyến tính. Chú ý rằng, phép tuyến tính hóa xung quanh điểm làm việc cho ta biểu diễn với các biến chênh lệch. Vì thế, từ mô hình tuyến tính hóa ta có thể dẫn xuất ngay ra mô hình hàm truyền đạt.

6.2 Các định luật vật lý áp dụng trong hệ thống

6.2.1. Nhắc lại các phần tử cơ bản

6.2.1.1. Các phần tử cơ bản của hệ động lực học bao gồm:

- Phần tử điện trở
- Phần tử điện dung
- Phần tử điện cảm

6.2.1.2. Các biến cơ bản bao gồm:

- Điện lượng q [C]
- Điện thế u [V]
- Cường độ dòng điện I [A]

Ta có bảng sau mô tả các biến trong các loại hệ thống khác nhau:

Loại hệ thống	Biến		
	Lượng	Thế	Thời gian
Điện	Điện tích	Điện thế	Giây
Cơ khí	Khoảng cách	Lực	Giây
Lưu chất lỏng	Thể tích		Giây
Nhiệt	Nhiệt năng	Nhiệt độ	Giây

6.2.1.3. Các biến khác nhau được định nghĩa trên các biến cơ bản:

- *Cường độ dòng*: là biến thiên lượng trong một đơn vị thời gian (hay gọi tắt là sự biến thiên lượng)

- *Công suất*: công suất = thế * cường độ dòng

- Định nghĩa phần tử trở: trở là đại lượng đặc trưng cho khả năng chống lại sự dịch chuyển cơ học hay dòng vật chất năng lượng. “*Trở được đo bằng thế cần thiết để chuyển 1 đơn vị lượng trong một đơn vị thời gian*”.

- Định nghĩa phần tử dung: “Dung biểu diễn mối quan hệ giữa lượng và thế. Dung được đo bằng lượng cần thiết cho thế biến thiên một đơn vị”.

$$dung = \frac{lượng}{thế}$$

$$Thế = \frac{1}{dung} \int (\text{cường độ dòng điện}) dt$$

- Định nghĩa phần tử cảm: cảm hay quán tính là đại lượng đặc trưng cho khả năng chống lại sự thay đổi trạng thái chuyển động cơ học của dòng vật chất, năng lượng.

“Cảm được đo bằng thế cần thiết để làm tốc độ biến thiên của lượng thay đổi một đơn vị”.

$$cảm = \frac{thế}{\frac{d(cường\độ\đòng)}{dt}}$$

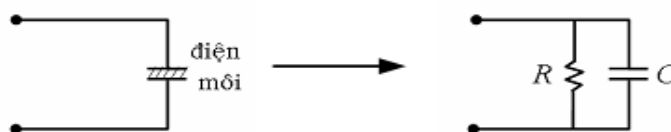
$$Suy\ ra: thế = cảm \times \frac{d(cường\độ\đòng)}{dt}$$

6.2.2. Lý tưởng hóa các phần tử vật lý

Cấu tạo nên hệ thống là các phần tử vật lý mà trên thực tế để xây dựng được các mô hình toán học của hệ thống hay là mô hình hóa hệ thống thì người ta phải đơn giản hóa hệ thống hay chính là lý tưởng hóa các phần tử vật lý bên trong hệ thống. Các phần tử vật lý cấu thành nên hệ thống thông thường được lý tưởng hóa theo các nguyên tắc như sau: nguyên tắc thuần hóa, nguyên tắc tập trung hóa, nguyên tắc lý tưởng hóa. Như vậy lý tưởng hóa các phần tử vật lý của hệ thống cũng chính là mô hình hóa các phần tử vật lý đó, các nguyên tắc lý tưởng hóa các phần tử vật lý như sau:

6.2.2.1. Nguyên tắc thuần hóa

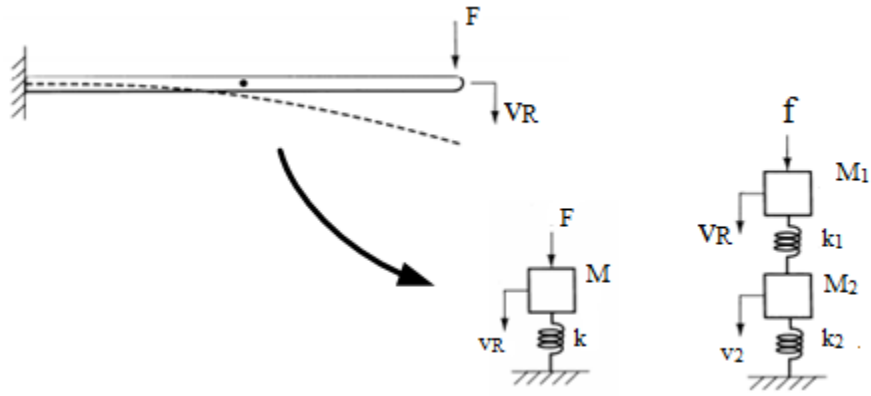
Khi có một phần tử vật lý thực tế nào đó, việc đầu tiên để chúng ta đi mô hình hóa được đối tượng vật lý thì chúng ta phải nhận ra được ảnh hưởng vật lý cơ bản chi phối hoạt động của đối tượng và sau đó dùng các phần tử thuần để biểu diễn đối tượng đó. Ví dụ chúng ta đi xét đối tượng là tụ điện được ứng dụng trong các hệ thống mạch điện thì ta thấy tụ điện thực tế bao giờ cũng gồm có thành phần điện trở trong và thành phần dung kháng cấu thành nên. Vậy có thể thay thế tụ điện thực bằng 2 thành phần cơ bản là thành phần điện trở và tụ điện để quá trình mô hình hóa đối tượng tụ điện được đơn giản.



Hình 6.7. Lý tưởng hóa phần tử tụ điện

6.2.2.2. Nguyên tắc tập trung hóa

Các ảnh hưởng vật lý thực luôn phân bố trong một miền hay không gian nhất định (dù nhỏ). Các ảnh hưởng phân bố này có thể lý tưởng hóa bằng cách mô hình hóa tập trung.



Hình 6.8. Lý tưởng hóa hệ thống cơ học

6.2.2.3. Nguyên tắc tuyến tính hóa

Thông thường tất cả các hệ thống thực đều là hệ phi tuyến hay hệ ngẫu nhiên nên để xây dựng được mô hình toán học của hệ thống hay mô hình hóa được hệ thống thì chúng ta phải *lý tưởng hóa* các phần tử đó bằng cách *tuyến tính hóa*.

6.2.3. Các phương trình cân bằng

Các định luật bảo toàn khối lượng, năng lượng và xung lượng là các định luật cơ bản được sử dụng khi mô hình hóa các phần tử vật lý, các đối tượng, các hệ thống. Trong các hệ thống điện, người ta dựa vào các định luật Kiffhop về dòng và áp, các định luật về vòng và nút, các định luật về năng lượng từ trường.... Trong hệ thống cơ học thì người ta thường dựa vào các định luật bảo toàn cơ năng, định luật bảo toàn động năng và bảo toàn thế năng... Ngoài ra còn một số hệ thống khác nữa sẽ được đi tìm hiểu rõ hơn trong các mục dưới đây.

Phương trình cân bằng cơ bản có dạng tổng quát như sau:

$$\text{Dòng tích lũy} = \text{dòng vào} - \text{dòng ra}$$

Phương trình trên được phát biểu như sau: “Năng lượng dòng tích lũy chính bằng tổn hao năng lượng giữa dòng vào và dòng ra”.

Nếu hệ thống không có các phần tử tích trữ khối lượng, năng lượng và xung lượng thì phương trình trên trở thành:

$$0 = \text{dòng vào} - \text{dòng ra}$$

Nếu hệ thống có phần tử tích trữ khối lượng hay xung lượng thì sự tích trữ làm thay đổi trạng thái của hệ thống:

$$\frac{d(\text{biến trạng thái})}{dt} = \text{dòng vào} - \text{dòng ra}$$

6.2.4. Phân tích vật lý hệ thống điện

Trong mạch điện, các phương trình cân bằng vật lý của hệ thống điện như sau:

1/ Định luật Kirchoff về dòng được phát biểu bởi công thức như sau:

“Tổng dòng điện đi vào bằng tổng dòng điện đi ra tại nút đó hay tổng đại số dòng điện tại một nút bằng 0”.

$$\sum_{k=1}^n i_k = 0$$

2/ Định luật Kirchoff về áp được phát biểu bởi công thức như sau:

“ Trong một vòng kín thì tổng đại số các sụt áp bằng 0”

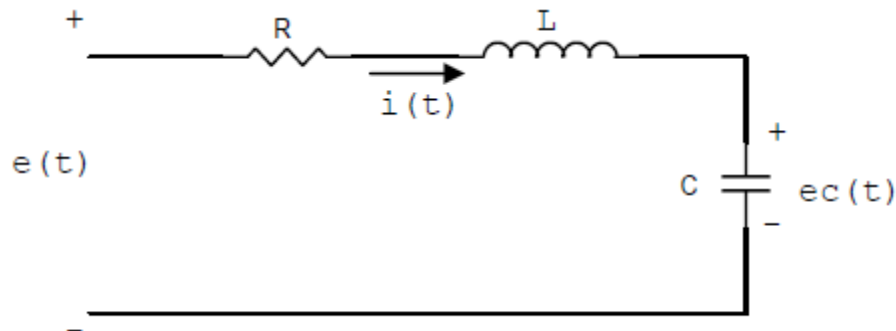
$$\sum_{k=1}^n u_k = 0$$

Dựa trên hai định cơ bản về mạch điện, từ đó ta có hai phương pháp giải tích mạch điện:

- Phương pháp điện thế đỉnh
- Phương pháp dòng vòng

Bằng cách dựa trên các định luật và các phương pháp giải tích trong mạch điện.

Ví dụ 6.2: Cho mạch điện sau



Hình 6.9. Minh họa cho ví dụ 6.2

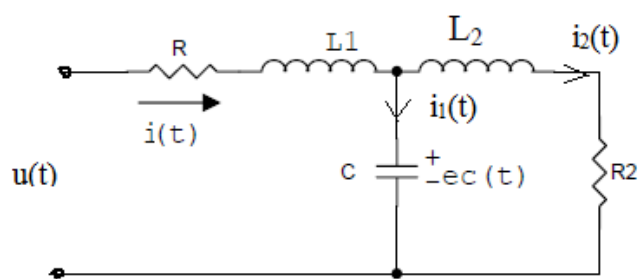
Viết các phương trình mạch điện cho mạch trên.

Giải:

Các phương trình trong mạch điện là:

Theo định luật Kiffhop 2 cho mạch vòng ta có: $e(t) = R \cdot i(t) + L \frac{di(t)}{dt} + \frac{1}{C} \int i(t) dt$

Ví dụ 6.3: Xét mạch điện dưới đây



Hình 6.10. Minh họa cho ví dụ 6.3

Theo các định luật Kiffhop 1 về dòng và định luật Kiffhop 2 về áp trong mạch điện, ta có các phương trình cân bằng sau:

Áp dụng định luật Kiffhop 2 cho mạch vòng 1 gồm (R, L₁, C), ta có:

$$u(t) = R \cdot i(t) + L_1 \frac{di(t)}{dt} + \frac{1}{C} \int di_1(t)$$

Áp dụng định luật Kiffhop 2 cho mạch vòng 2 gồm (R₂, L₂, C), ta có:

$$\frac{1}{C} \int di_1(t) = R_2 \cdot i_2(t) + L_2 \frac{di_2(t)}{dt}$$

Áp dụng định luật Kiffhop 1 cho nút ta có:

$$i(t) = i_1(t) + i_2(t)$$

6.3 Mô hình cơ bản của các hệ động lực học

6.3.1. Mô hình hàm truyền của hệ thống liên tục

Mô hình cho hệ thống liên tục SISO có tín hiệu vào là $u(t)$ và tín hiệu ra $y(t)$. Hàm truyền đạt $G(s)$ được định nghĩa như là tỷ số giữa ảnh Laplace $Y(s)$ của đáp ứng xung tín hiệu ra $y(s)$ cho ảnh Laplace $U(s)$ của tín hiệu vào khi hệ được kích thích từ trạng thái 0 tức là có các điều kiện đầu bằng 0.

$$G(s) = \frac{Y(s)}{U(s)}$$

Việc cho các điều kiện đầu bằng 0 thì không ảnh hưởng tới bản chất tính động học của mô hình.

Để đưa ra được phương trình hàm truyền đạt của hệ thống, người ta thực hiện chuyển đổi từ phương trình vi phân theo toán tử Laplace.

Xuất phát từ phương trình:

$$a_0 \frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_{n-1} \frac{d y}{dt} + a_n y = b_0 \frac{d^m u}{dt^m} + b_1 \frac{d^{m-1} u}{dt^{m-1}} + \dots + b_{m-1} \frac{d u}{dt} + b_m u$$

Theo biến đổi Laplace ta có:

$$\begin{aligned} a_0 s^n Y(s) + a_1 s^{n-1} Y(s) + \dots + a_{n-1} s Y(s) + a_n Y(s) \\ = b_0 s^m U(s) + b_1 s^{m-1} U(s) + \dots + b_{m-1} s U(s) + b_m U(s) \end{aligned}$$

Tương đương:

$$(a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n) Y(s) = (b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m) U(s)$$

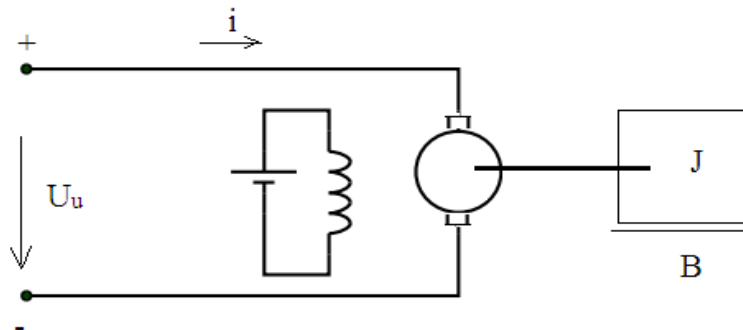
$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n}$$

Vậy hàm truyền đạt của hệ thống liên tục là:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n}$$

Chú ý: Hàm truyền đạt của hệ trên chỉ có nghĩa cho hệ liên tục tuyến tính và có các trạng thái đầu bằng 0.

Ví dụ 6.4: Cho hệ thống động cơ điện một chiều như hình vẽ



Hình 6.11. Minh họa cho ví dụ 6.4 về động cơ điện một chiều

Với J là mô men quá tính quy đổi về trục động cơ; B là hệ số ma sát ở trục; giả sử từ thông không đổi.

Xác định hàm truyền đạt của động cơ với điện áp phần ứng U_u là tín hiệu vào và tín hiệu ra là tốc độ động cơ ω ?

Dựa vào các định luật Kinhhoff trong mạch điện phần ứng động cơ điện một chiều và mạch điện kích từ của động cơ:

Ta có các phương trình cân bằng điện sau

$$U_u = i.R + L \frac{di}{dt} + E_u$$

$$U_{KT} = i_{kt}.R_{kt} + L_k \frac{di_{kt}}{dt}$$

$$M = J \frac{d\omega}{dt} + B.\omega = K\phi i ;$$

$$E_u = K\phi\omega ;$$

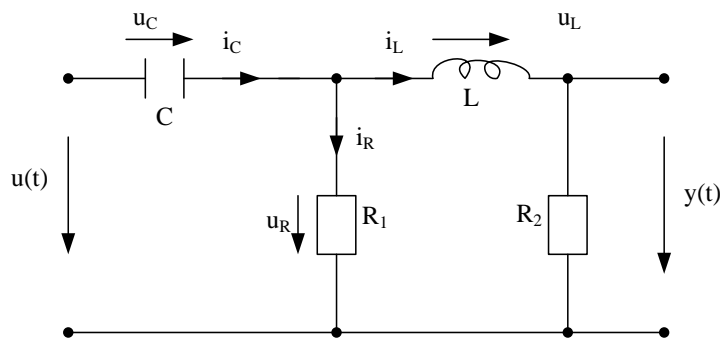
Suy ra được:

$$G(s) = \frac{\omega(s)}{U_u(s)} = \frac{K}{(Ls+R)(J.s+B) + K^2}$$

So với mô hình toán học dạng phương trình vi phân thì mô hình toán học dạng hàm truyền có tính ưu việt hơn hẳn là vì quan hệ giữa tín hiệu vào và ra được mô tả dưới dạng tuyến tính. Điều đó giúp xác định đáp ứng đầu ra $y(t)$ khi có kích thích đầu vào $u(t)$ cho trước được đơn giản hơn nhiều so với việc giải các phương trình vi phân bậc cao. Với hàm truyền trên giúp cho việc khảo sát các đặc tính động học cũng đơn giản và nhanh chóng.

Hàm truyền đạt được dẫn ra từ phương trình vi phân mô tả quan hệ vào – ra của hệ thống song có một số trường hợp không nhất thiết là để có hàm truyền đạt thì phải có phương trình vi phân. Ta đi xét một số ví dụ sau:

Ví dụ 6.5: Cho một mạch điện biết trước các giá trị của C của tụ điện, L của cuộn dây, R_1 và R_2 của điện trở là các phần tử trong mạch điện. Hãy xác định hàm truyền đạt mô tả quan hệ giữa tín hiệu vào là điện áp $u(t)$ và tín hiệu ra $y(t)$ là điện áp trên R_2 của hệ thống mạch điện trên.



Hình 6.12. Minh họa cho ví dụ 6.5

Trong ví dụ trên đã xây dựng các phương trình vi phân dựa trên các quy luật về điện, từ đó có thể đưa ra được hàm truyền của hệ thống. Nhưng ở ví dụ này có thể lấy ảnh Laplace trực tiếp của chúng mà không cần thông qua các phương trình vi phân phức tạp.

Giả sử gọi $U_C(s)$ là ảnh của u_C , $U_L(s)$ là ảnh của u_L ; $U_R(s)$ là ảnh của u_R ; $I_C(s)$ là ảnh của i_C ; $I_L(s)$ là ảnh của i_L ; $I_R(s)$ là ảnh của i_R . Khi đó ta có các mối quan hệ như sau:

$$\begin{cases} I_C(s) = C.s.U_C(s) \\ U_L(s) = L.s.I_L(s) \\ U_R(s) = R_1.I_R(s) \\ U(s) = U_C(s) + U_R(s) \end{cases}$$

Theo định luật Kirchhoff 1, 2 ta có:

$$\begin{aligned} U_R(s) &= Y(s) + L.s.I_L(s) \\ U(s) &= U_C(s) + U_R(s) \\ I_C(s) &= I_R(s) + I_L(s) \end{aligned}$$

Từ các phương trình trên, rút ra ta được phương trình sau:

$$\begin{aligned} CR_1R_2.s.U(s) &= CR_1R_2.s.U_R(s) + CR_1R_2.s.U_C(s) \\ CR_1R_2.s.U(s) &= [CLR_1.s^2 + (CR_1R_2 + L).s + (R_1 + R_2)]Y(s) \end{aligned}$$

Suy ra hàm truyền của hệ trên là:

$$\begin{aligned} U(s) &= [CLR_1.s^2 + .s. + (R_1 + R_2)]Y(s) \\ G(s) &= \frac{CR_1R_2.s.}{CLR_1.s^2 + (CR_2R_1 + L)s + (R_1 + R_2)} \end{aligned}$$

Bài tập cuối chương.

Bài 1: Cho ví dụ về các hệ thống điều khiển liên tục trong thực tế sản xuất?

Bài 2: Cho các hệ thống sau:

Hệ thống đo áp suất

Hệ thống lưu lượng

Hệ thống đo nhiệt độ

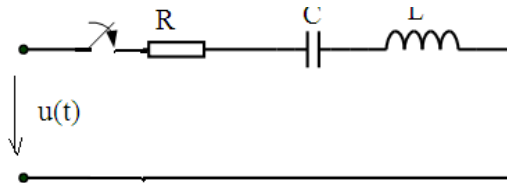
Các hệ thống trên thuộc loại hệ thống liên tục hay hệ thống rời rạc?

Bài 3: Cho ví dụ về hệ thống động? Cho ví dụ về hệ thống tĩnh trong thực tế sản xuất?

Bài 4: Cho một mạch dao động điện R-L-C. Hỏi mạch trên thuộc hệ thống động hay hệ thống tĩnh?

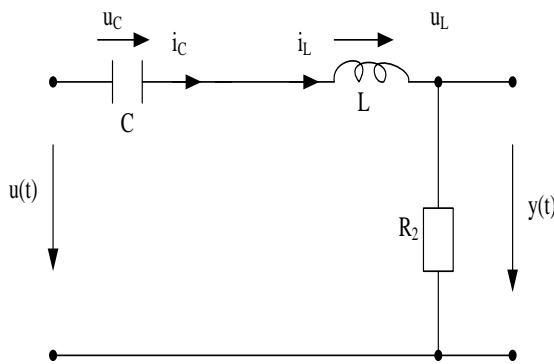
Bài 5: Cho một hệ thống điều khiển liên tục. Để tiến hành mô hình hóa hệ thống đó thì người ta sử dụng mô hình nào?

Bài 6: Cho mạch điện như hình vẽ:

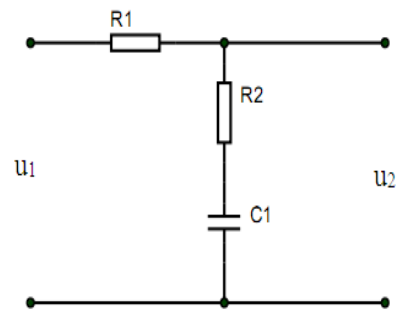


Hỏi hệ thống mạch điện trên thuộc loại hệ thống điều khiển nào? Vì sao?

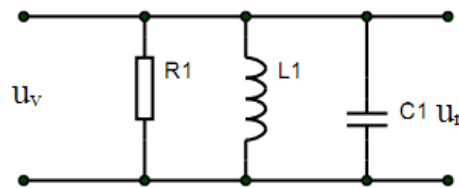
Bài 7: Cho mạch điện sau



a)



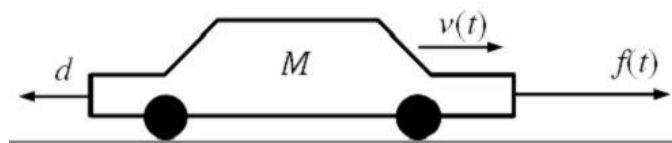
b)



c)

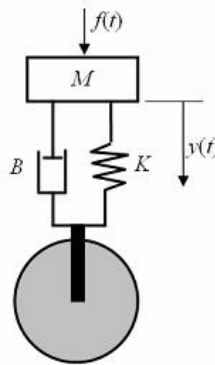
Xác định hàm truyền đạt của hệ thống $G(s)$.

Bài 8: Tìm hàm truyền của ô tô sau



Với: M - khối lượng xe, B - hệ số ma sát; $f(t)$ - lực kéo; $v(t)$ - tốc độ xe

Bài 9: Tìm hàm truyền của hệ thống giảm xóc của ô tô, xe máy sau



Với: M - khối lượng tác động lên bánh xe, B hệ số ma sát, K độ cứng lò xo; $f(t)$: lực do xóc, $y(t)$: dịch chuyển của thân xe

Câu hỏi trắc nghiệm

Câu 1: Thế nào là hệ thống liên tục?

- a/ Hệ thống liên tục có tất cả các tín hiệu tồn tại trong hệ đều liên tục theo thời gian
- b/ Hệ thống liên tục có tất cả các tín hiệu tồn tại trong hệ đều gián đoạn theo thời gian
- c/ Hệ thống liên tục có tất cả các tín hiệu tồn tại trong hệ vừa liên tục và gián đoạn theo thời gian
- d/ Hệ thống liên tục có tất cả các tín hiệu tồn tại trong hệ nối tiếp nhau.

Câu 2: Các hệ động lực học thường được chia thành mấy loại?

a/ 4 loại

b/ 5 loại

c/ 6 loại

e/ 3 loại

Câu 3: Hàm truyền đạt của hệ thống liên tục có cấu trúc?

a/

b/

c/

d/

Câu 4: Các phần tử cơ bản của hệ động lực học bao gồm:

a/ Phần tử điện trở, điện cảm, điện dung

b/ Phần tử điện trở, hồ cảm, điện dung

c/ Phần tử điện trở, điện thế, dòng điện

d/ Phần tử điện trở, điện cảm, điện thế

Câu 5: Mô hình hóa các phần tử vật lý bao gồm các nguyên tắc?

a/ Nguyên tắc thuần hoá, nguyên tắc tập trung hoá, nguyên tắc tuyến tính hoá

b/ Nguyên tắc thuần hoá, nguyên tắc liên tục hoá, nguyên tắc tuyến tính hoá

c/ Nguyên tắc rải rác hoá, nguyên tắc tập trung hoá, nguyên tắc tuyến tính hoá

d/ Nguyên tắc thuần hoá, nguyên tắc tập trung hoá, nguyên tắc phi tuyến hoá

Chương VII: ỨNG DỤNG CỦA MATLAB TRONG KỸ THUẬT

Nội dung chính của chương

Chương 7 cung cấp cho sinh viên những ứng dụng cơ bản của Matlab trong kỹ thuật điều khiển, mô phỏng các mạch điện tử công suất và mô phỏng các hệ cơ điện.

Mục tiêu cần đạt được của chương

Chương 7 giúp sinh viên có thể ứng dụng Matlab vào việc phân tích, tính toán, mô phỏng các quá trình trong hệ thống và giải quyết các bài toán kỹ thuật.

Bài 8: Ứng dụng Matlab trong điều khiển (số tiết: 3 tiết)

7.1 Ứng dụng Matlab trong điều khiển

7.1.1 Các lệnh kết nối hệ thống

Lệnh feedback

a) Công dụng:

Kết nối hai hệ thống hồi tiếp.

b) Cú pháp:

`[num,den] = feedback(num1,den1, num2,den2)`

`[num,den] = feedback(num1,den1, num2,den2,sign)`

c) Giải thích:

sign = 1: Hồi tiếp dương.

sign = -1: Hồi tiếp âm.

Nếu bỏ qua tham số sign thì lệnh sẽ hiểu là hồi tiếp âm.

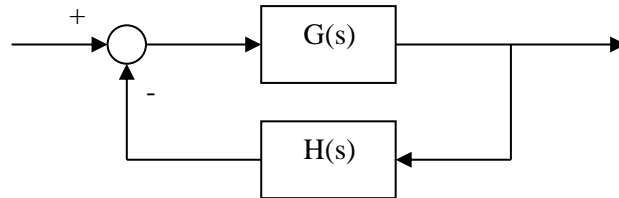
Hàm truyền của hệ thống là:

$$\frac{num(s)}{den(s)} = \frac{G_1(s)}{1 \mp G_1(s)G_2(s)} = \frac{num_1(s)den_2(s)}{den_1(s)den_2(s) \mp num_1(s)num_2(s)}$$

d) Ví dụ:

Kết nối khâu có hàm truyền $G(s) = \frac{2s^2 + 5s + 1}{s^2 + s + 3}$ với khâu hồi tiếp có hàm truyền

$H(s) = \frac{5(s + 2)}{s + 10}$ theo dạng hồi tiếp âm như sau:



numg = [2 5 1];

deng = [1 2 3];

numh = [5 10];

denh = [1 10];

[num,den] = feedback(numg, deng, numh, denh);

Kết quả:

num =

2 25 51 10

den =

11 57 78 40

Lệnh series

a) Công dụng:

ghép nối hai hệ thống nối tiếp nhau.

b) Cú pháp:

[num,den] = series(num1,den1, num2,den2)

$$\frac{num(s)}{den(s)} = G_1(s)G_2(s) = \frac{num_1(s)num_2(s)}{den_1(s)den_2(s)}$$

d) Ví dụ 1:

Kết nối 2 khâu có hàm truyền $G(s)$ và $H(s)$ để tạo thành hệ thống nối tiếp.

$$G(s) = \frac{3}{s + 4}, \quad H(s) = \frac{2s + 4}{s^2 + 2s + 3}$$

Ta thực hiện như sau:

```

num1 = 3;
den1 = [1 4];
num2 = [2 4];
den2 = [1 2 3];
[num,den] = series(num1,den1, num2,den2)
ta được kết quả:
num = [0 0 6 12]
den = [1 6 11 12]

```

Lệnh parallel

a) Công dụng:

Ghép nối hai hệ thống song song.

b) Cú pháp:

```

[a,b,c,d] = parallel(a1,b1,c1,d1,a2,b2,c2,d2)
[num,den] = parallel(num1,den1, num2,den2)

```

c) Giải thích:

```

[num,den] = parallel(num1,den1, num2,den2)

```

d) Ví dụ:

Nối 2 khâu có hàm truyền $G(s)$ và $H(s)$ thành hệ thống song song:

$$G(s) = \frac{3}{s + 4}$$

$$H(s) = \frac{2s + 4}{s^2 + 2s + 4}$$

```

numg = 3;
deng = [1 4];
numh = [2 4];
denh = [1 2 3];
[num,den] = parallel(numg, deng, numh, denh);

```

Ta được hệ thống song song có hàm truyền

$G'(s) = \text{num}(s)/\text{den}(s)$ với các hệ số:

$$\text{num} = [0 \quad 5 \quad 18 \quad 25]$$

$$\text{den} = [1 \quad 6 \quad 11 \quad 12]$$

7.1.2 Chuyển hệ thống từ hàm truyền thành dạng không gian trạng thái

Lệnh TF2SS

a) Công dụng:

Chuyển hệ thống từ dạng không gian hàm truyền thành dạng trạng thái.

b) Cú pháp:

$$[a,b,c,d] = \text{tf2ss}(\text{num},\text{den})$$

c) Ví dụ 1:

Xét hệ thống có hàm truyền:

$$H(s) = \frac{2s+3}{s^2+2s+1}$$

Chuyển hệ thống thành dạng không gian trạng thái :

$$\text{Num} = [0 \quad 2 \quad 3; \quad 1 \quad 2 \quad 3];$$

$$\text{den} = [1 \quad 0.4 \quad 1];$$

$$[a,b,c,d] = \text{tf2ss}(\text{num},\text{den});$$

ta được kết quả:

$$a =$$

$$\begin{matrix} -0.4000 & -1.0000 \\ 1.0000 & 0 \end{matrix}$$

$$$$

$$b =$$

$$1$$

$$0$$

$$c =$$

$$2.0000 \quad 3.0000$$

$$1.0000 \quad 2.0000$$

$$d =$$

$$0$$

$$1$$

Ví dụ 2: Cho hàm truyền: $(s^2+7s+2) / (s^3+9s^2+26s+24)$

Chuyển hệ thống thành dạng không gian trạng thái :

» num=[1 7 2];

» den=[1 9 26 24];

» [A,B,C,D]=tf2ss(num,den)

7.1.3 Chuyển từ mô hình không gian trạng thái sang mô hình hàm truyền đạt

Lệnh SS2TF

a) Công dụng:

Chuyển hệ thống từ mô hình không gian trạng thái sang mô hình hàm truyền thành dạng

b) Cú pháp:

[num,den] = ss2tf(a,b,c,d,iu)

$$H(s) = \frac{NUM(s)}{den(s)} = C(sI - A)^{-1} B + D$$

từ ngõ vào thứ i. Vecto den chứa hệ số của mẫu số theo chiều giảm dần số mũ của s. Ma trận NUM chứa các hệ số tử số với số hàng là ngõ ra

c) Ví dụ 1: Chuyển từ mô hình không gian trạng thái sang mô hình hàm truyền đạt

% Nhập mô hình không gian trạng thái

a = [1 2 3; 4 5 6; 7 8 9];

b = [3 4; 4 5; 7 9]; % co 2 dau vao

c = [0 1 0]; % co 1 dau ra

d = [0 0];

% Chuyển đổi mô hình

[num, den] = ss2tf(a,b,c,d,1)

% Kết quả:

num =

0 4.0000 14.0000 12.0000

den =

1.0000 -15.0000 -18.0000 -0.0000

7.1.4 Đánh giá khả năng điều khiển được, quan sát được của hệ thống

Lệnh GRAM, DGRAM

a) Công dụng:

Đánh giá khả năng điều khiển và khả năng quan sát.

b) Cú pháp:

$$Gc = \text{gram}(a,b)$$

$$Go = \text{gram}(a',c')$$

$$Gc = \text{dgram}(a,b)$$

$$Go = \text{dgram}(a',c')$$

c) Ví dụ:

Xác định khả năng điều khiển của hệ không gian trạng thái ở ví dụ về lệnh `dcgrain`

$$a = [-0.5572 \quad -0.7814; \quad 0.7814 \quad 0];$$

$$b = [1 \quad 0.5379; \quad 0 \quad -0.2231];$$

$$c = [1.9691 \quad 6.4439; \quad 1 \quad 0];$$

$$d = [0 \quad 0; \quad 0 \quad 0];$$

$$Gc = \text{gram}(a,b)$$

Ta nhận được ma trận:

$$Gc = \begin{bmatrix} 1.2016 & -0.0318 \\ -0.0318 & 1.0708 \end{bmatrix}$$

Tìm hạng ma trận bằng lệnh:

$$r = \text{rank}(Gc)$$

Ta được $r = 2$ và bằng kích thước của ma trận đánh giá. Vậy hệ thống này có thể điều khiển được.

7.1.5. Khảo sát hệ thống theo tiêu chuẩn tần số

- Hệ thống ổn định ở trạng thái hở, sẽ ổn định ở trạng thái kín nếu biểu đồ Nyquist không bao điểm $(-1+i0)$ trên mặt phẳng phức.
- Hệ thống không ổn định ở trạng thái hở, sẽ ổn định ở trạng thái kín nếu biểu đồ Nyquist bao điểm $(-1+i0)$ p lần ngược chiều kim đồng hồ (p là số cực GH nằm ở phải mặt phẳng phức).

» `num = [nhập các hệ số của tử số theo chiều giảm dần của số mũ].`

» `den = [nhập các hệ số của mẫu số theo chiều giảm dần của số mũ].`

» `nyquist(num,den)`

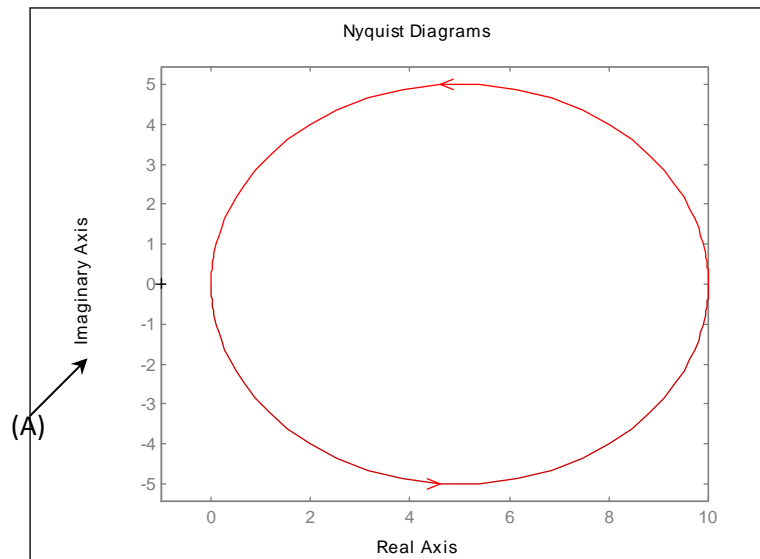
$$GH(s) = \frac{k}{1-st} \quad (\text{với } k=10, t=1)$$

» `num = 10;`

» `den = [-1 1];`

» nyquist(num,den)

Kết quả:



Nhận xét: hàm truyền vòng hở có 1 cực nằm bên phải mặt phẳng phức. Biểu đồ Nyquist không bao điểm A $(-1+j0)$.

Điểm -1 ký hiệu (+) nằm trên trục thực âm (Real Axis), điểm 0 nằm trên trục ảo (Imaginary Axis).

Kết luận: Hệ thống không ổn định.

Bài tập cuối bài:

Bài 1: Cho mô hàm truyền của hệ thống như sau:

$$H(s) = \frac{s^2 + 3s - 1}{s^3 - 5s^2 + s + 3}$$

Chuyển mô hình hàm truyền đạt này sang mô hình không gian trạng thái

Xác định đáp ứng của hệ thống với tín hiệu vào dạng bước nhảy

Bài 2: Cho mô hàm truyền của hệ thống như sau:

$$H(s) = \frac{s^2 + 3s - 1}{s^3 - 5s^2 + s + 3}$$

Chuyển mô hình hàm truyền đạt này sang mô hình không gian trạng thái

Xét ổn định cho hệ thống.

Bài 3: Cho mô hình không gian trạng thái của hệ thống với các ma trận tương ứng:

$$A = \begin{bmatrix} -0.5572 & -0.7814 \\ 0.7814 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0.5379 \\ 0 & -0.2231 \end{bmatrix}$$

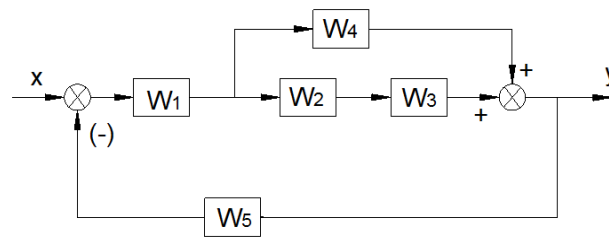
$$C = \begin{bmatrix} 1.9691 & 6.4439 \\ 1 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Chuyển mô hình không gian trạng thái này sang mô hình hàm truyền đạt

Xét tính điều khiển được của hệ thống.

Bài 4: Cho hệ thống có sơ đồ khối như sau:



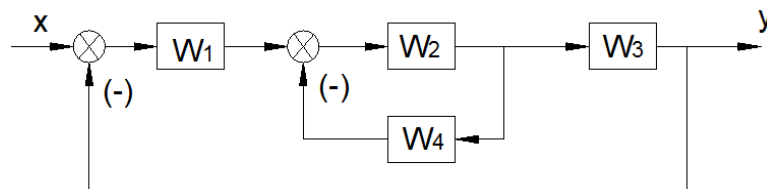
$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$

Hãy xây dựng hàm truyền hệ kín.

Chuyển mô hình hàm truyền này sang mô hình không gian trạng thái.

Bài 5: Cho hệ thống có sơ đồ khối như sau:



$$W1 = 0,5 \quad ; \quad W2 = 1/(s+2)(s+8);$$

$$W3 = 1/s \quad ; \quad W4 = 2$$

Hãy xây dựng hàm truyền hệ kín.

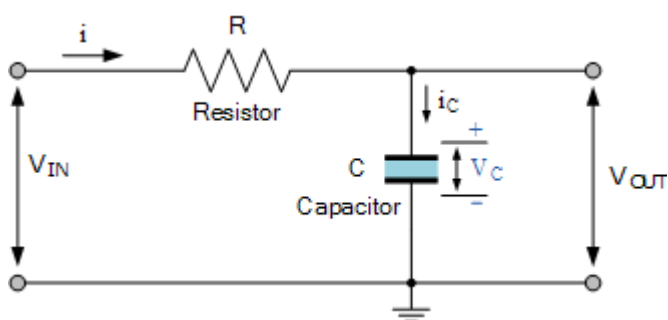
Vẽ đồ thị đáp ứng của hệ thống với tín hiệu vào dạng bước nhảy

Bài 9: Ứng dụng matlab trong mô phỏng các mạch điện – điện tử (số tiết: 3 tiết)

7.2 Ứng dụng Matlab trong mô phỏng các mạch điện – điện tử

7.2.1 Mạch tích phân RC

Mạch tích phân RC là một mạch RC được kết nối nối tiếp tạo ra tín hiệu đầu ra tương ứng với quá trình tích phân toán học. Mạch RC thụ động không gì khác hơn là một điện trở mắc nối tiếp với tụ điện, đó là một điện trở cố định mắc nối tiếp với tụ điện có điện trở phụ thuộc tần số giảm khi tần số trên các bản của nó tăng lên. Do đó, ở tần số thấp, dung kháng, X_c của tụ điện cao trong khi ở tần số cao, **dung kháng của tụ điện** thấp với công thức dung kháng tiêu chuẩn của $X_c = 1 / (2\pi fC)$.



Hình 7.1: Mạch tích phân RC

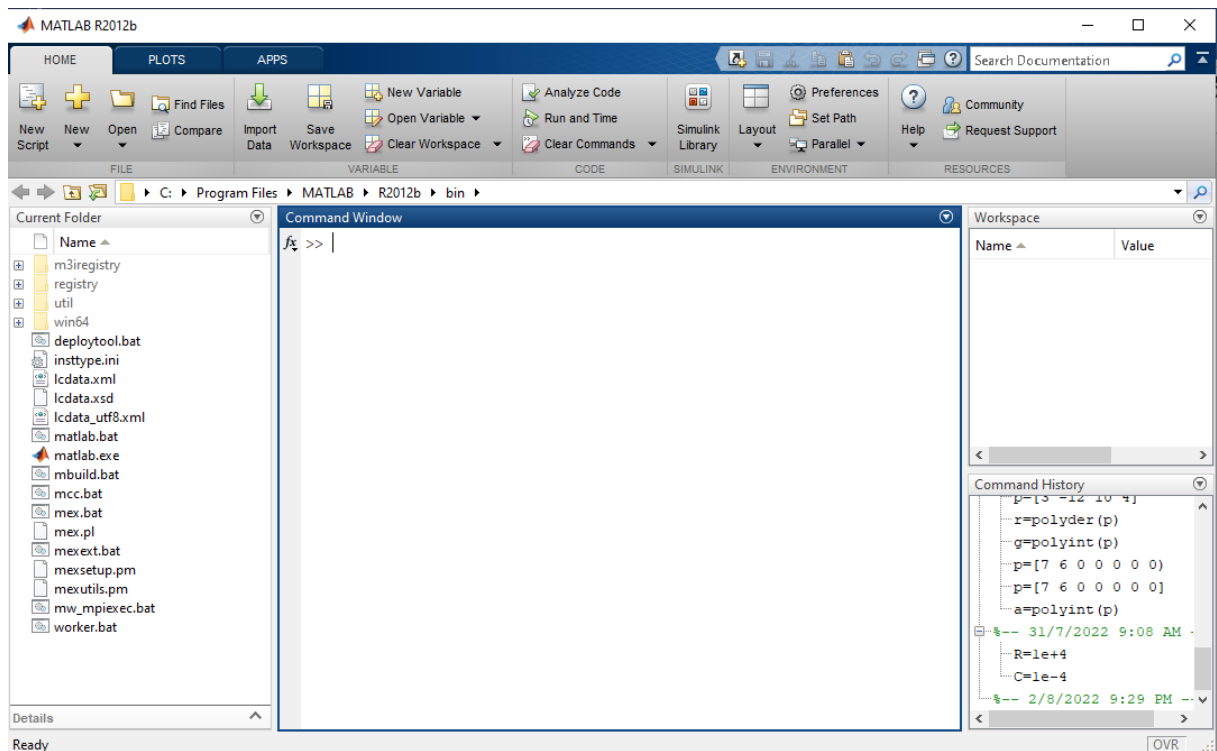
Phương trình cân bằng là:

$$e_i - iR - \frac{1}{c} \int idt = 0$$

từ đây ta có thể rút ra

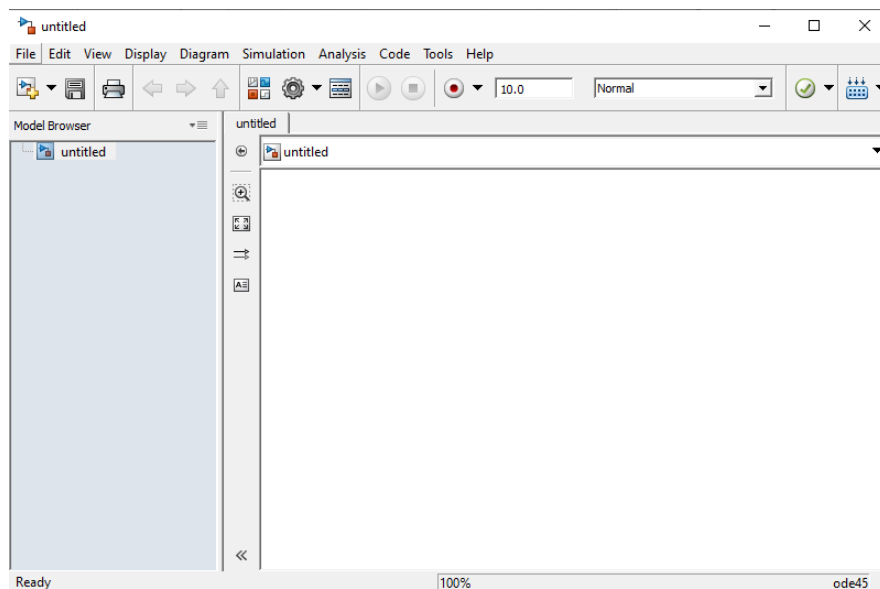
$$i = \frac{1}{R} * (e_i - \frac{1}{C} \int i dt)$$

Để mô phỏng mạch tích phân RC và quan sát tín hiệu đầu ra ta cần sử dụng SIMULINK.



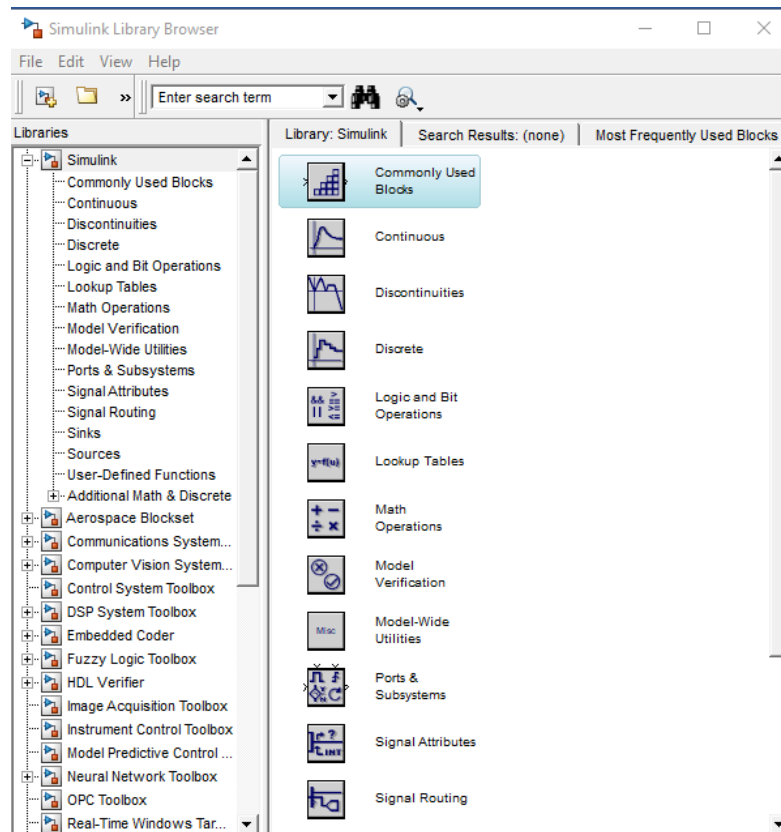
Hình 7.2: Mở Matlab 2012

Mở MATLAB bấm chọn *New* → *Simulink Mode*. Hoặc ta có thể bấm chuột phải vào khoảng trống của *Current Folder* chọn *Mode*.



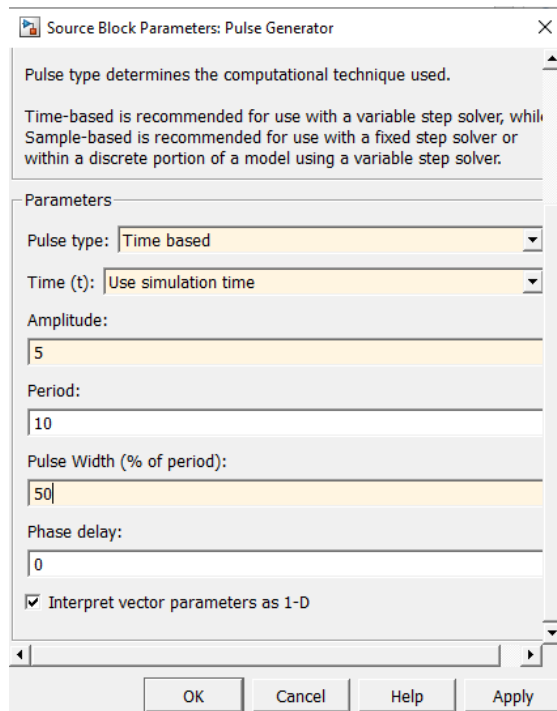
Hình 7.3: Cửa sổ Simulink Mode sử dụng để mô phỏng mạch điện tử.

Chọn View → Library Browser



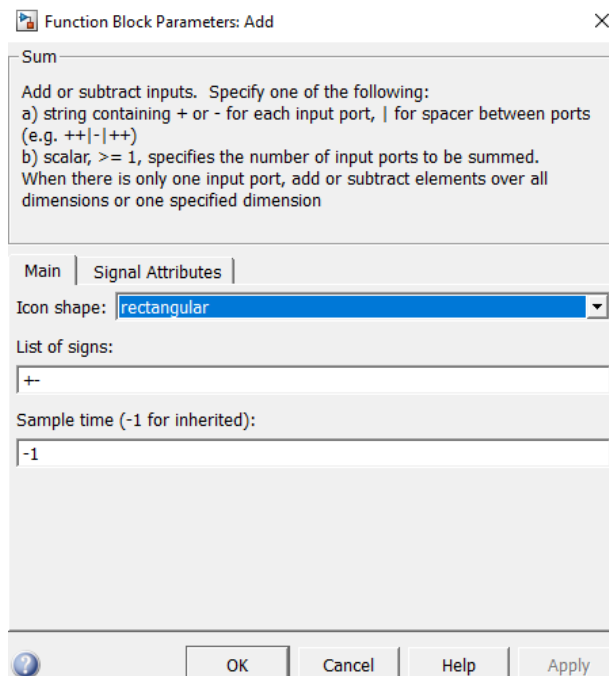
Hình 7.4: Cửa sổ Simulink Library Browser

Vào *Sources* lấy khối nguồn *Pulse Generator*. Đây là khối nguồn cấp xung vuông. Ta thay đổi thông số của nguồn như sau:



Hình 7.5: Thông số khối nguồn Pulse Generator

Tiếp tục vào *Math Operations* lựa chọn *Add*

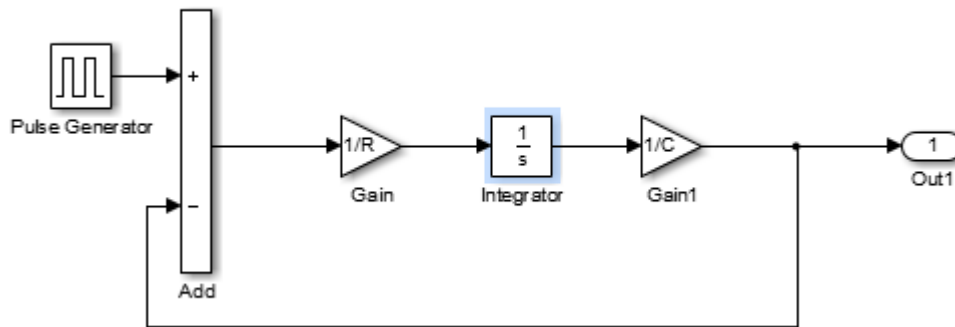


Hình 7.6: Lựa chọn trong Add

Chọn *Gain* sử dụng 2 khối *Gain* với giá trị mỗi khối tương ứng là $1/R$ và $1/C$

Trong *Continuous* lựa chọn *Integrator*.

Ghép nối các khối như hình dưới đây sau đó dùng *Scope* để quan sát tín hiệu ra.



Hình 7.7: Mô phỏng mạch tích phân RC

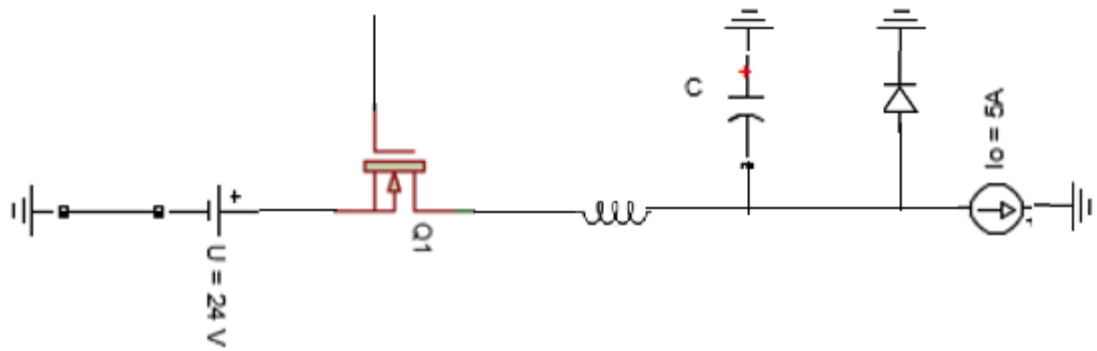
Thông qua cửa sổ Command Window cài đặt thông số cho R và C

```
Command Window
>> R=1e+4
R =
    10000
>> C=1e-4
C =
    1.0000e-04
```

Hình 7.8: Cài đặt thông số cho R và C

7.2.2. Mạch đóng mở sử dụng MOSFET

Mạch đóng mở sử dụng MOSFET là một mạch điện khá thông dụng trong mạch điện điện tử. Mô hình mạch đóng mở như sau



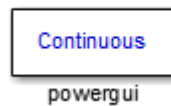
Hình 7.9: Mạch đóng ngắt dùng MOSFET

MOSFET thường được sử dụng để làm khóa đóng mở vì kênh dẫn của E-MOSFET hình thành hay không liên quan đến V_{GS} .

Khi $V_{GS} < V_{GS(th)}$ kênh dẫn giữa D và S chưa hình thành \Rightarrow Khóa hở

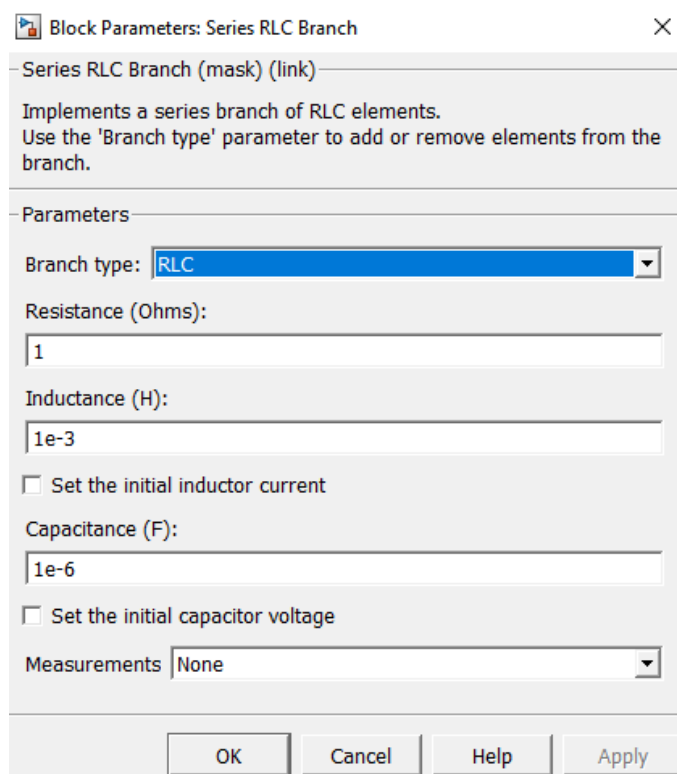
Khi $V_{GS} > V_{GS(off)}$ kênh dẫn hình thành giữa D và S \Rightarrow Khóa đóng

Trước tiên ta vào *Simscape* \rightarrow *SimPowerSystem* lấy khối *Powergui* để có thể kết nối nguồn với các thiết bị.



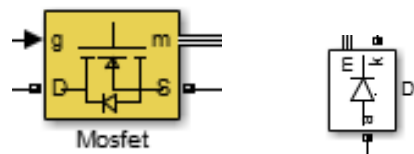
vào *Simscape* \rightarrow *SimPowerSystem* \rightarrow *Elements* để lấy các khối *Serials RLC Branch*

Trong đó ta có thể thay đổi loại tải bằng cách lựa chọn *Branch type* và thay đổi thông số của R, L, C tương ứng.



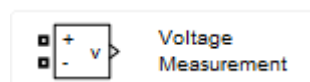
Hình 7.10: Lựa chọn loại thiết bị R, L, C tương ứng

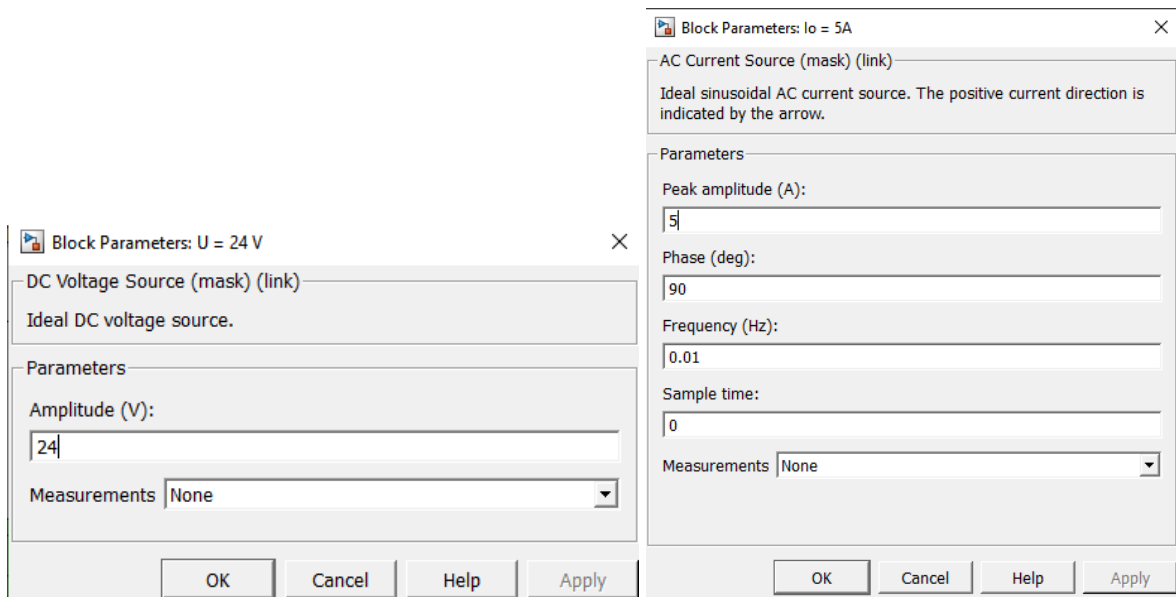
vào *Simscape* → *SimPowerSystem* → *PowerElectronics* để lấy khối Mosfet và Diot tương ứng.



vào *Simscape* → *ApplicantLibraries* → *ElectricalSources* để lấy khối nguồn dòng *AC Current Source* và *DC Voltage Source*.

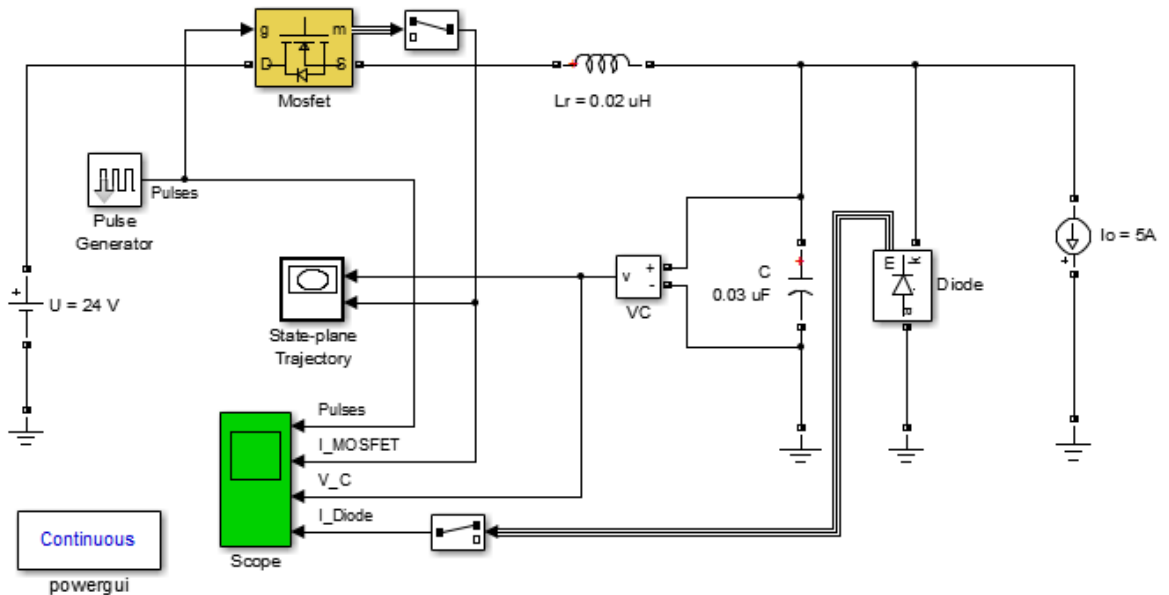
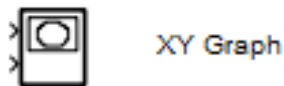
vào *Simscape* → *SimPowerSystem* → *Machines* → *Measurements*. Ở đây ta lấy khối đo áp để đo điện áp của tụ điện.





Hình 7.11: Cài đặt thông số điện áp và dòng điện.

Lấy khối vẽ đồ thị xy bằng cách vào *Simulink* → *Sink* → *XY Graph*

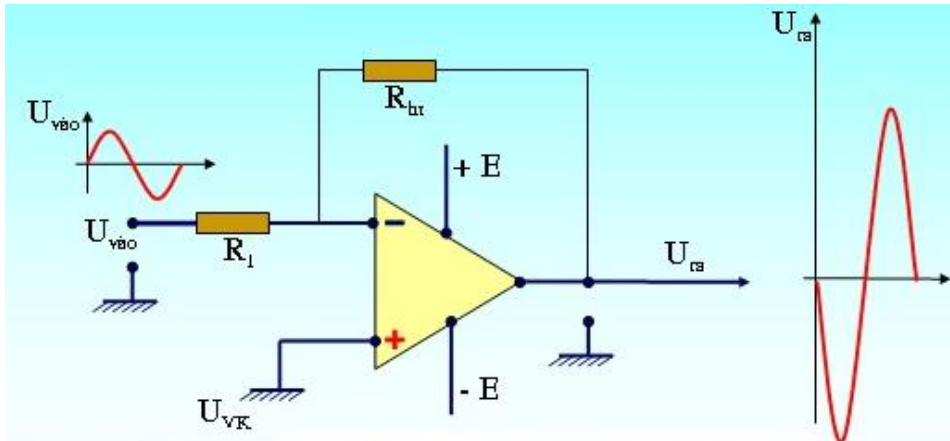


Hình 7.12: Mô phỏng trên Matlab simulink mạch đóng mở sử dụng MOSFET

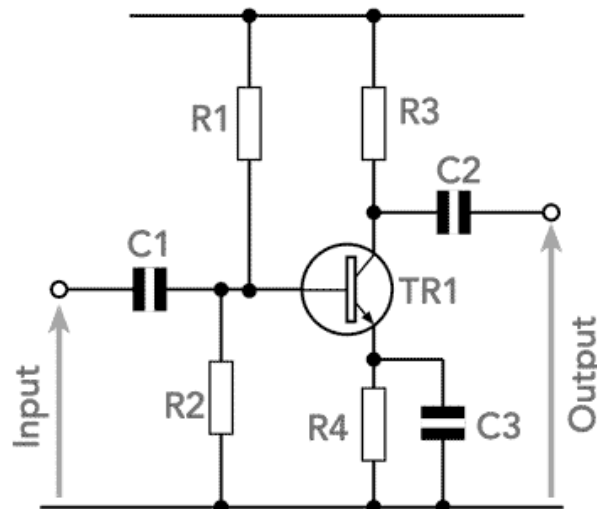
Câu hỏi cuối bài

Câu 1: Sử dụng Matlab/Simulink mô phỏng mạch vi phân RC và quan sát tín hiệu ra

Câu 2: Sử dụng Matlab/Simulink mô phỏng mạch khuếch đại sau

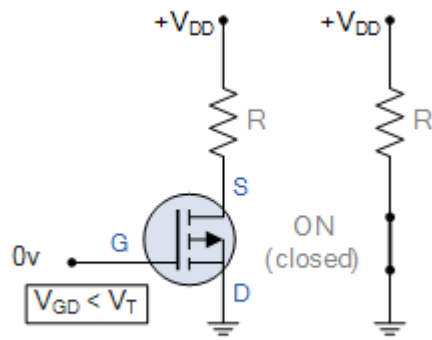


Câu 3: Sử dụng Matlab/Simulink mô phỏng mạch Khuếch đại E chung



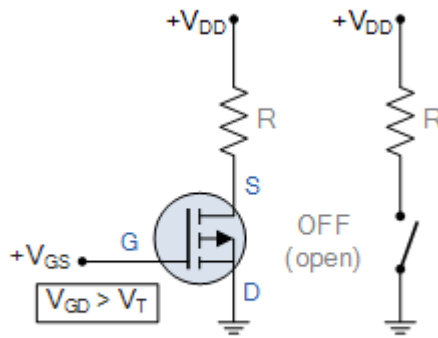
Câu 4: Sử dụng Matlab/Simulink mô phỏng mạch tích phân RC thay đổi giá trị thông số tải R, C và quan sát tín hiệu ra.

Câu 5: Sử dụng Matlab/Simulink mô phỏng mạch điện tử sau:



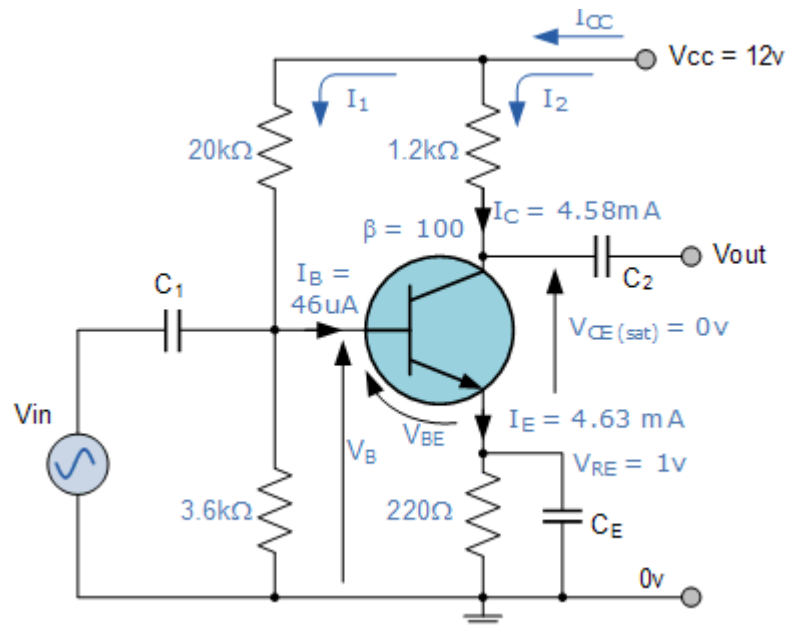
P-channel eMOSFET Biased ON

Câu 6: Sử dụng Matlab/Simulink mô phỏng mạch điện tử sau:

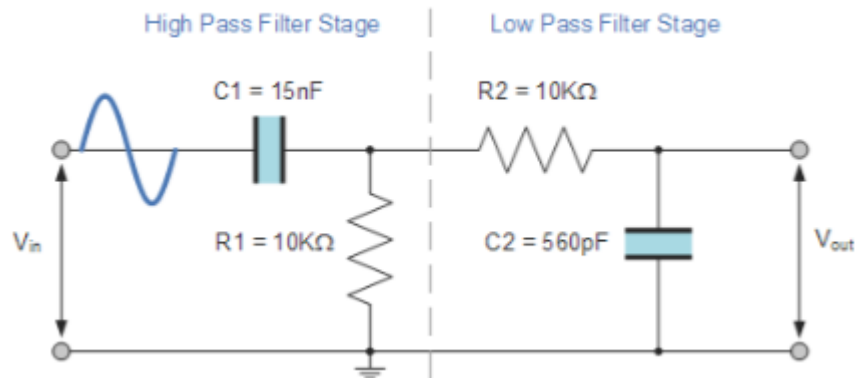


P-channel eMOSFET Biased OFF

Câu 7: Sử dụng Matlab/Simulink mô phỏng mạch điện tử sau:

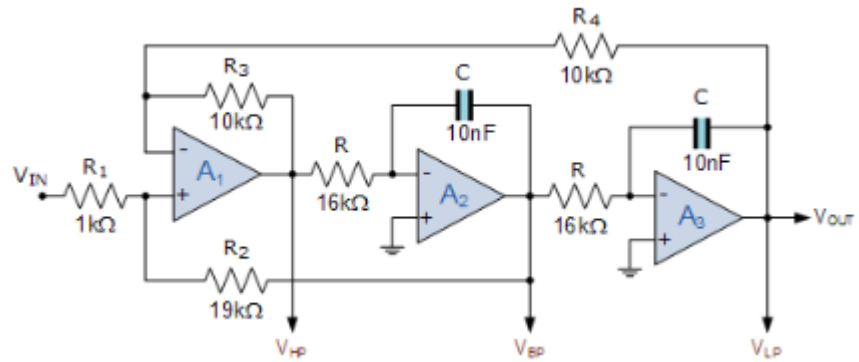


Câu 8: Sử dụng Matlab/Simulink mô phỏng mạch lọc thông dải thụ động và quan sát



Câu 9: Sử dụng Matlab/Simulin mô phỏng mạch lọc thông thấp thụ động và quan sát

Câu 10: Sử dụng Matlab/Simulin mô phỏng mạch lọc biến trạng thái và quan sát

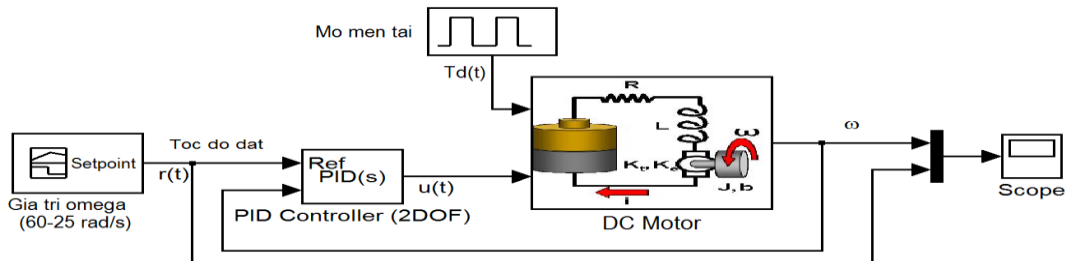


Bài 10: Ứng dụng Matlab trong mô phỏng các hệ điện cơ (số tiết: 3 tiết)

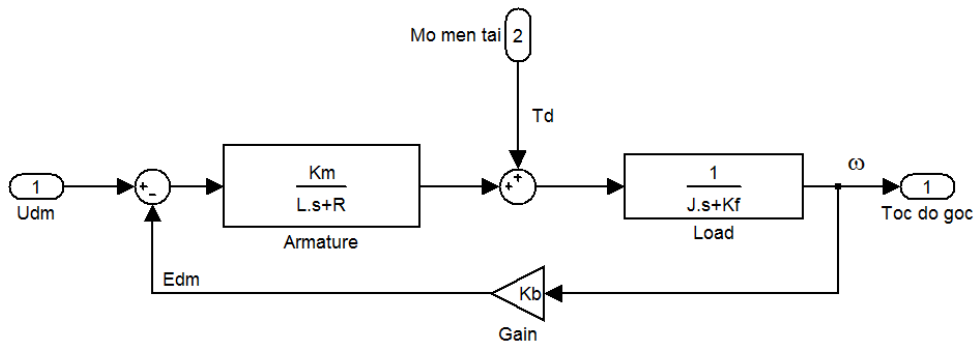
7.3 Ứng dụng Matlab trong mô phỏng các hệ điện cơ

Một số ví dụ mô phỏng hệ thống cơ điện

Ví dụ 7.1 :Sử dụng bộ điều khiển PID 2 bậc tự do điều khiển tốc độ động cơ điện một chiều kích từ độc lập (Hình 7.1)



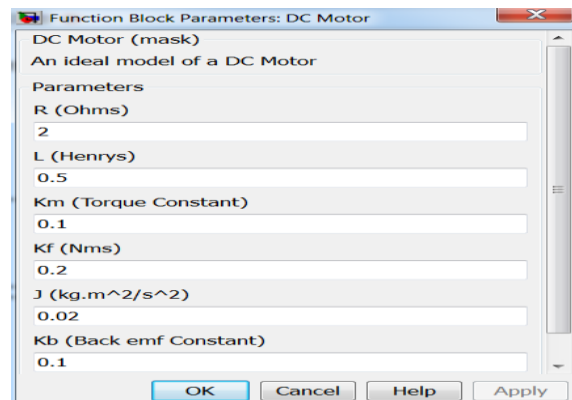
Hình 7.13. Mô hình sử dụng bộ PID 2 bậc tự do điều khiển tốc độ động cơ DC



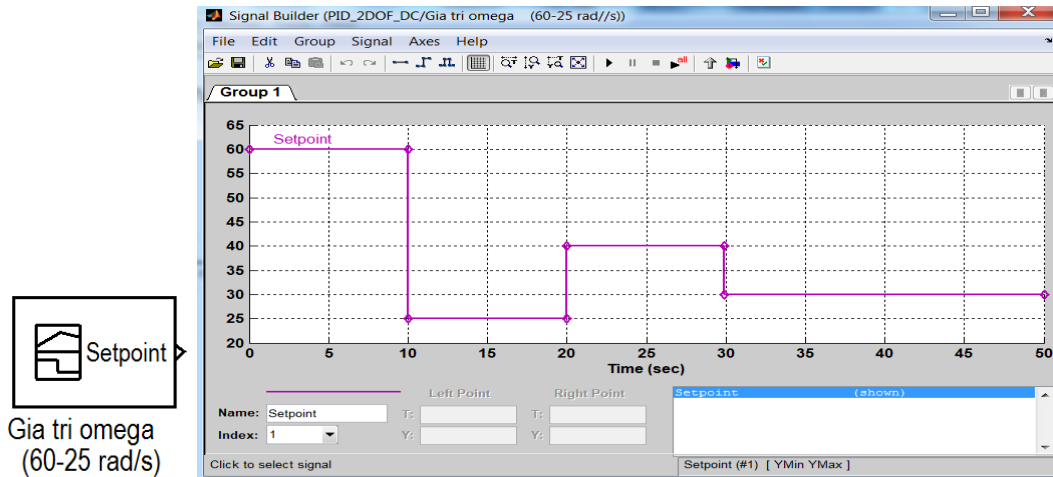
Hình 7.14. Khối động cơ DC

Các thông số của động cơ:

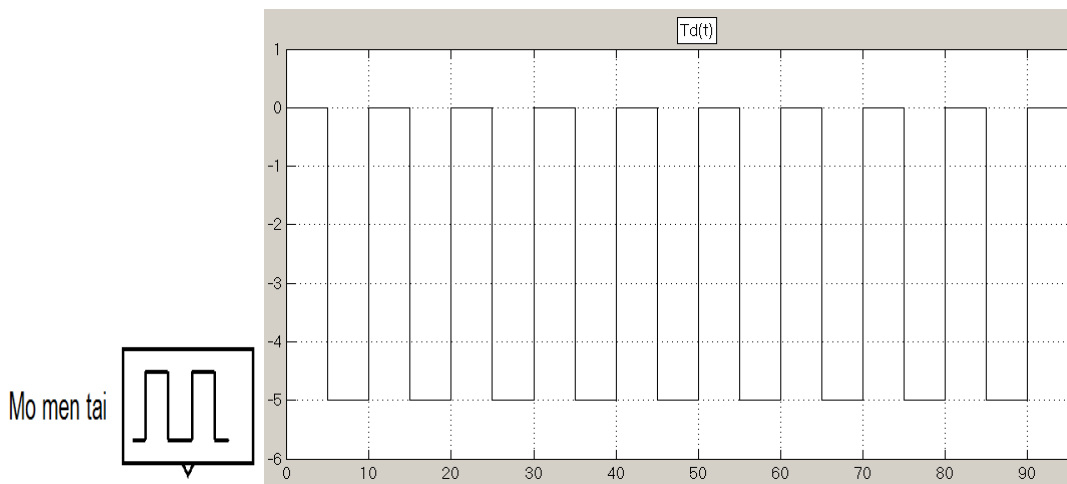
- Td : mômen tải
- J: mômen quán tính
- Km : hằng số mômen
- Kb: hằng số điện từ
- Kf: Hệ số giảm chấn
- L: điện cảm phản ứng
- R: điện trở phản ứng



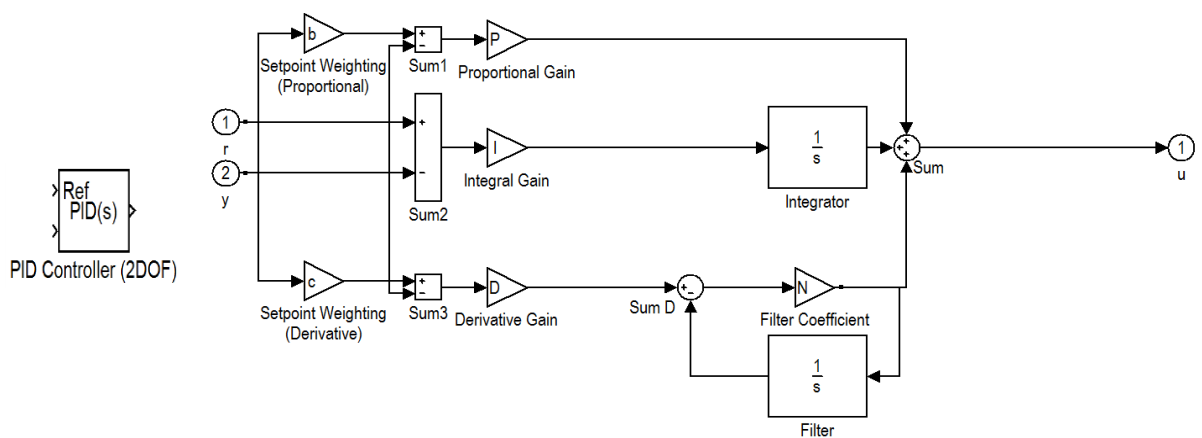
Hình 7.15: Bảng thông số động cơ DC



Hình 7.16. Khối tạo giá trị vận tốc góc ω đặt trước

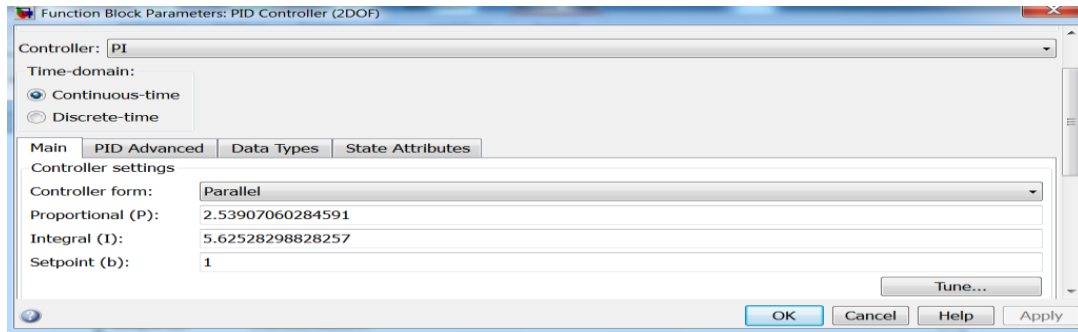


Hình 7.17. Khối mômen tải

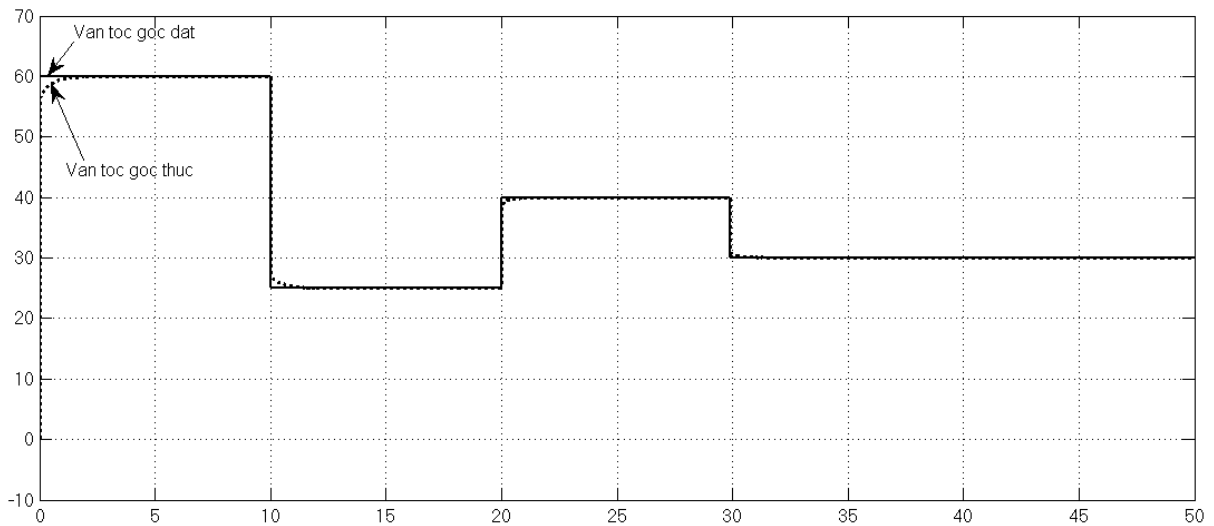


Hình 7.18. Khối PID Controller (2DOF)

Để tìm thông số của bộ PID ta sử dụng chức năng Tune của bộ PID giống như ví dụ 7.1. Ở ví dụ này ta lựa chọn bộ PI với thông số như hình 6.32



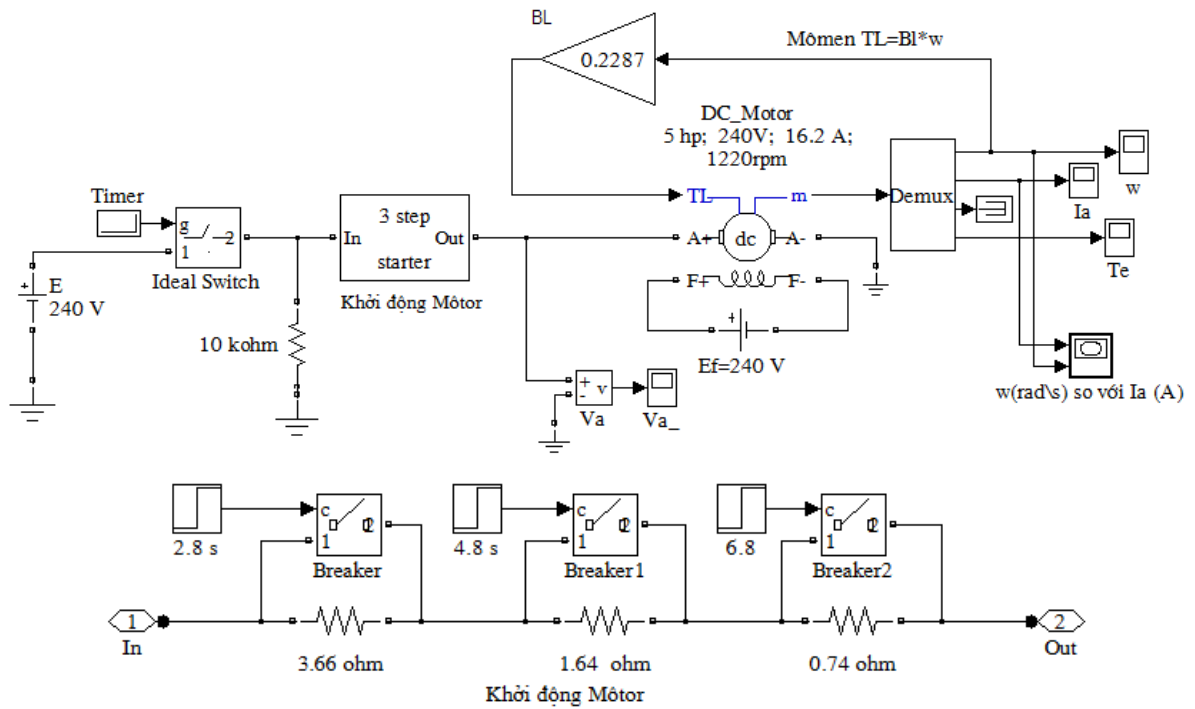
Hình 7.19. Các thông số của bộ PI



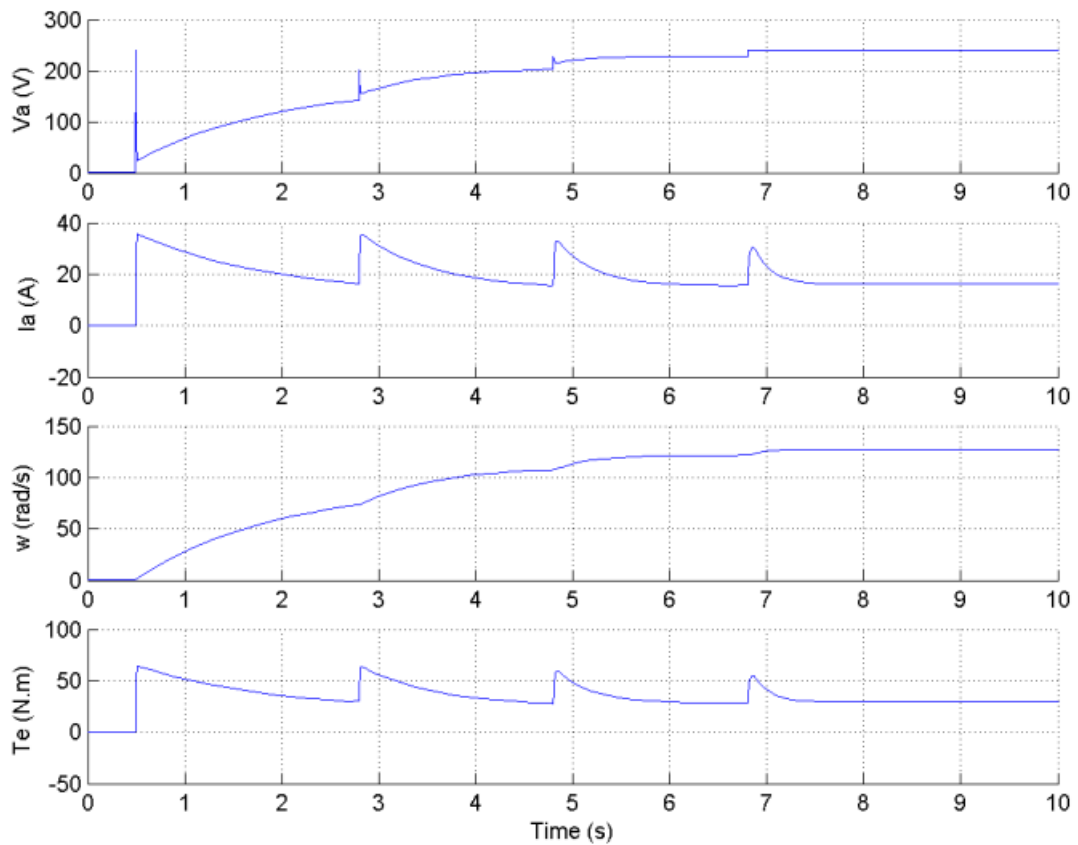
Hình 7.20. So sánh giữa vận tốc đặt và vận tốc thực

Nhận xét: Quan sát trên hình 7.9 ta thấy vận tốc thực bám sát vận tốc đặt

Ví dụ 7.2 Xây dựng mô hình Khởi động động cơ một chiều qua 3 cấp điện trở



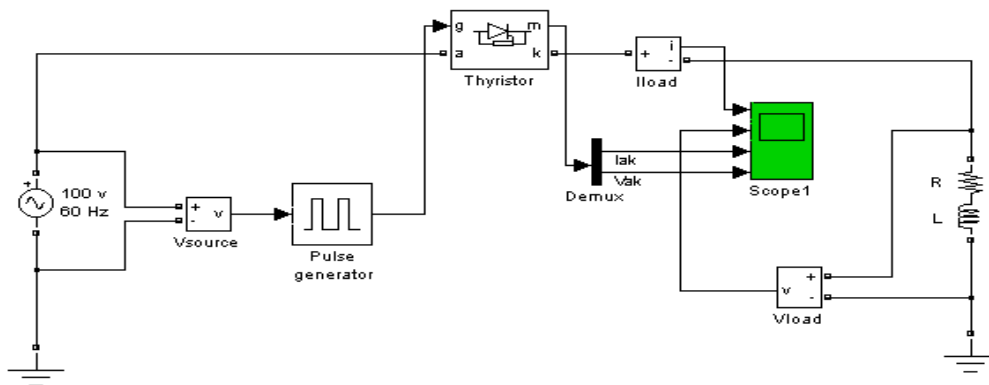
Hình 7.21. Mô hình khởi động cơ DC qua 3 cấp điện trở



Hình 7.22. Đồ thị của mô hình khởi động cơ DC qua 3 cấp điện trở

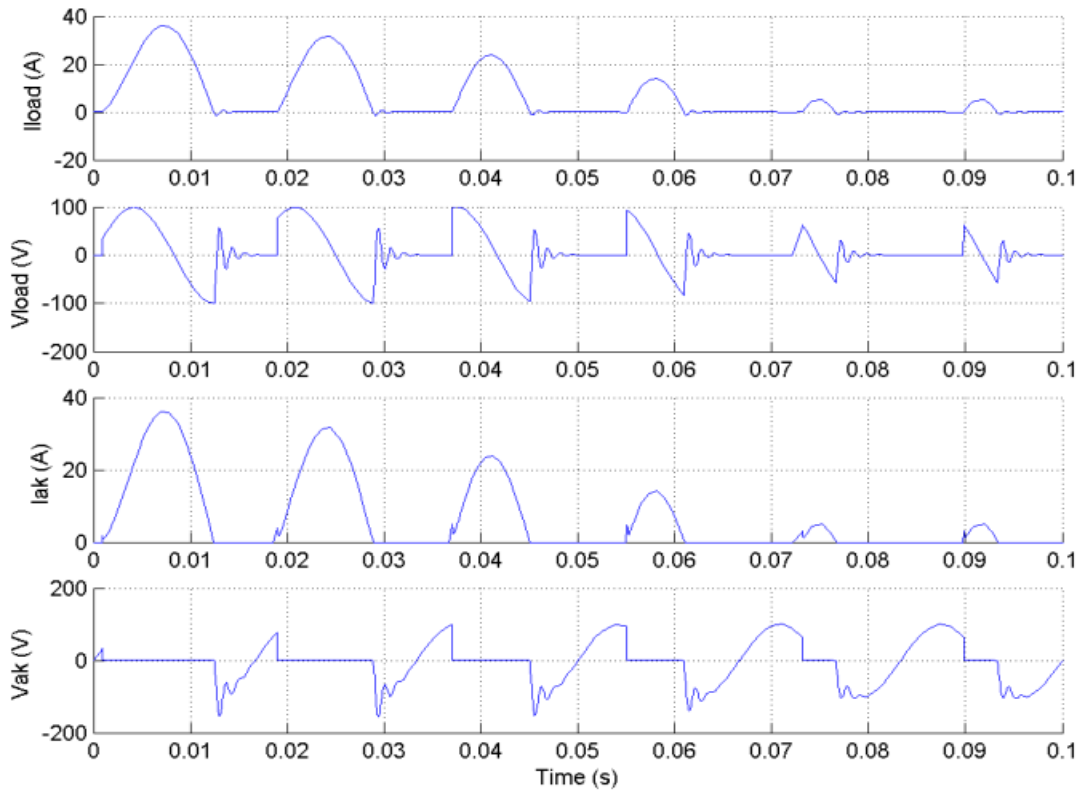
Ví dụ 7.3: Trong demo thyristor công suất với chỉnh lưu 1/2 chu kỳ được sử dụng để nuôi cho tải RL. Xung chân gate được cấp bởi bộ phát xung pulse generator đồng bộ với nguồn áp. Các tham số được sử dụng như sau:

R	1 Ω
L	10 mH
Thyristor block: Ron	0.001 W
Lon	0 H
Vf	0.8 V
Rs	20 Ω
Cs	4e-6 F



Hình 7.23. Chỉnh lưu 1/2 chu kỳ sử dụng thyristor

Góc phát xung được biến đổi bằng một bộ phát xung pulse generator đồng bộ với nguồn áp. Chạy mô phỏng và quan sát dòng/áp tải cũng như dòng/áp Tiristor.

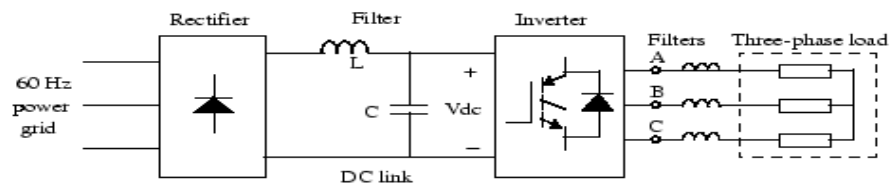


Hình 7.24. Các đồ thị chỉnh lưu $\frac{1}{2}$ chu kỳ sử dụng thyristor

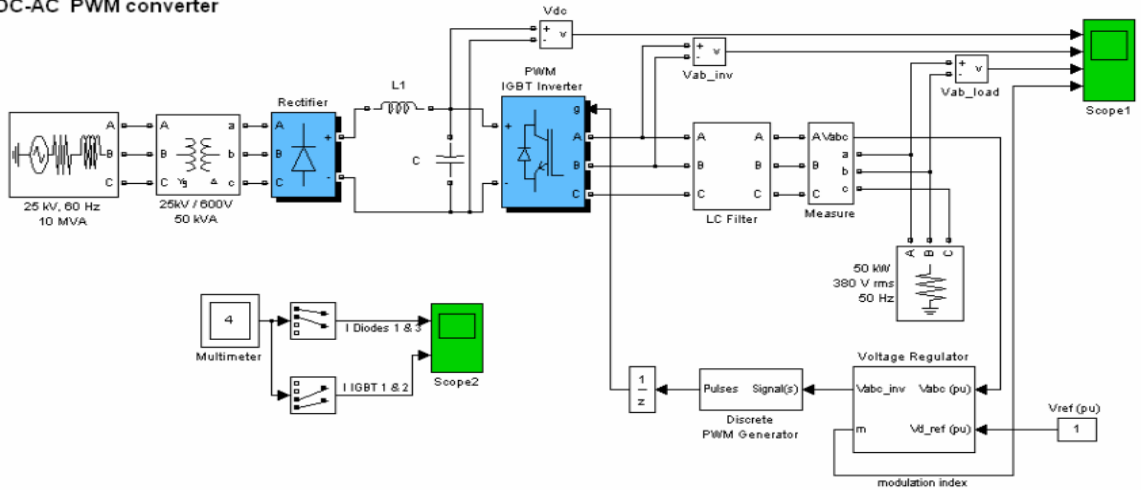
Ví dụ 7.4: Demo power_bridges minh chứng cho cách sử dụng của 2 bộ Universal Bridge blocks trong mạch biến đổi ac/dc/ac bao gồm chỉnh lưu nuôi tới biến tần IGBT thông qua ghép nối một chiều. Biến tần cùng với điều chế độ rộng xung (PWM) sinh ra điện áp sin 3 pha 50 Hz tới tải. Trong ví dụ này biến tần thay đổi nhanh với tần số 2000 Hz.

IGBT được điều khiển với bộ điều khiển PI cốt để duy trì 1 điện áp p.u. (380 V/ms, 50 Hz) tại các cực tải.

Một khối Multimeter được dùng để quan sát commutation của các dòng giữa diodes 1 và 3 trong cầu diode và giữa van IGBT/Diodes 1 và 2 trên cầu IGBT.



AC-DC-AC PWM converter

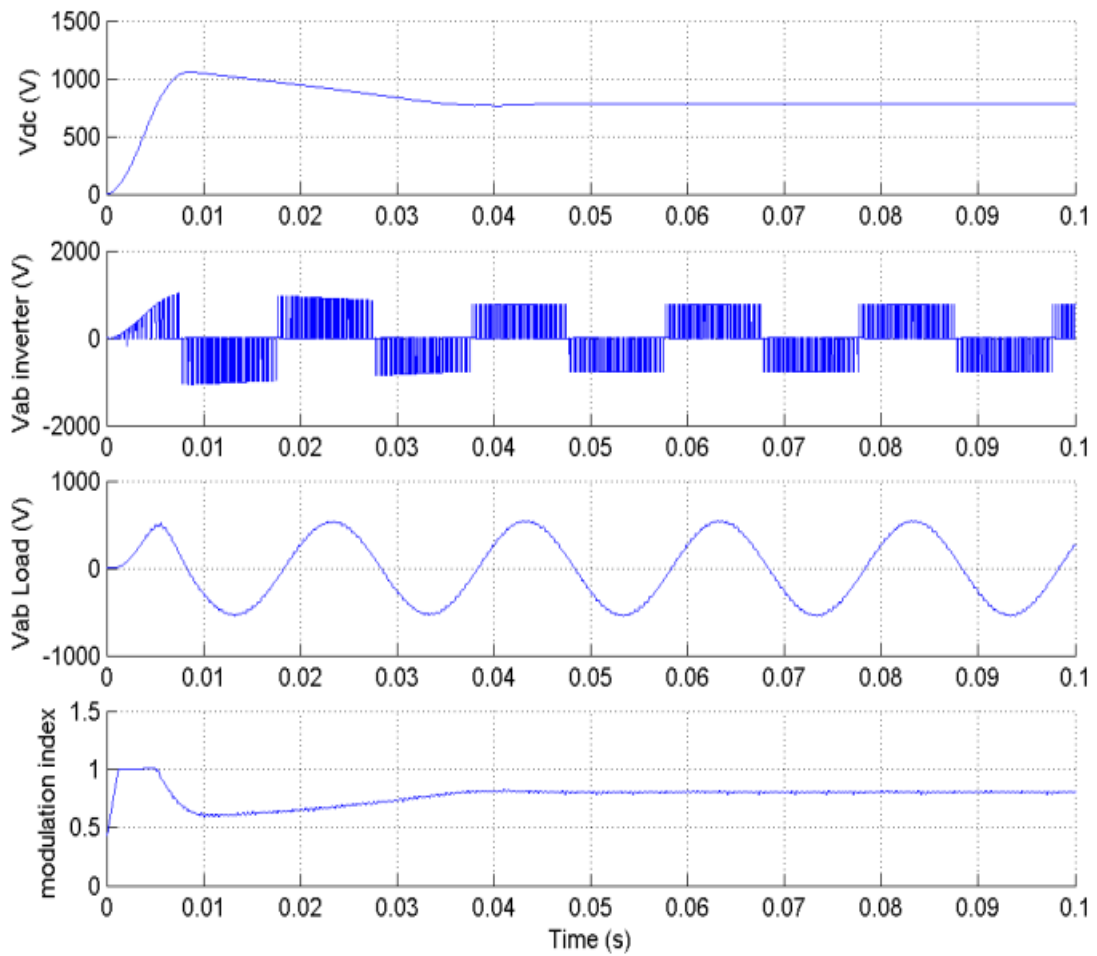


Hình 7.25. Ví dụ sử dụng hai bộ Universal Bridge blocks trong mạch biến đổi ac/dc/ac

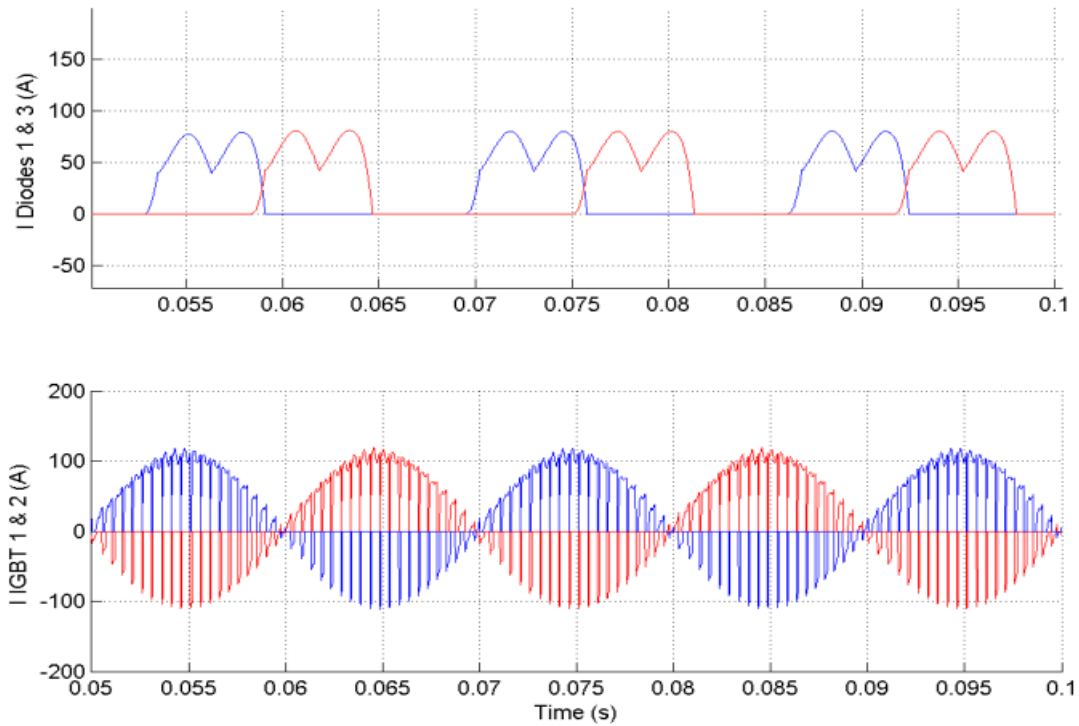
Bắt đầu mô phỏng. Sau quãng thời gian xấp xỉ 40 ms, hệ thống đặt trạng thái steady state. Quan sát dạng sóng áp tại DC bus, giá trị ra inverter, và tải trên Scope1. Harmonics được phát ra bởi biến tần quan multiples 2 kHz được lọc bởi bộ lọc LC. Khi giá trị xung tải đặt 537 V (380 V RMS).

Trong trạng thái steady state, chỉ số điều biến $m = 0.77$, và điện áp DC là 780 V. Thành phần cơ bản của điện áp 50 Hz là $V_{ab} = 780 \text{ V} * 0.612 * 0.80 = 382 \text{ V RMS}$.

Quan sát dòng Diode trên Scope2, chỉ ra từ diode 1 tới diode 3. Cũng quan sát dòng trace 2 trên van 1 và 2 của cầu IGBT/Diode (van cao hơn và thấp hơn nối tới pha A). Hai dòng này bổ sung cho nhau. Một dòng dương xác định dòng chảy trong IGBT, ngược lại dòng âm xác định một dòng chảy trong Diode chống song song.



Hình 7.26. Đồ thị của ví dụ sử dụng hai bộ Universal Bridge blocks trong mạch biến đổi ac/dc/ac



Hình 7.27. Đồ thị của ví dụ sử dụng hai bộ Universal Bridge blocks trong mạch biến đổi ac/dc/ac

Bài tập cuối chương

Bài 1: Cho mô hình không gian trạng thái có

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 5 & 2 & 1 \\ 4 & 5 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 3 \\ 1 & 2 \\ 3 & 5 \end{bmatrix}$$

$$C = [0 \quad 1 \quad 0] \quad D = [0 \quad 0]$$

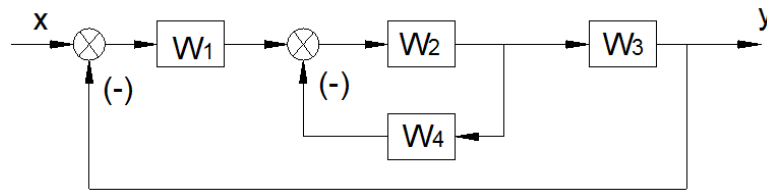
Chuyển mô hình không gian trạng thái này sang mô hình hàm truyền đạt

Bài 2: Cho mô hình hàm truyền đạt, có hàm truyền

$$H(s) = \frac{s^2 + 3s - 1}{s^3 - 5s^2 + s + 3}$$

Chuyển mô hình hàm truyền đạt này sang mô hình không gian trạng thái

Bài 3: Cho hệ thống có sơ đồ khối như sau



$$W1 = 0,5 ; W2 = 1/(s+2)(s+8);$$

$$W3 = 1/s ; W4 = 2$$

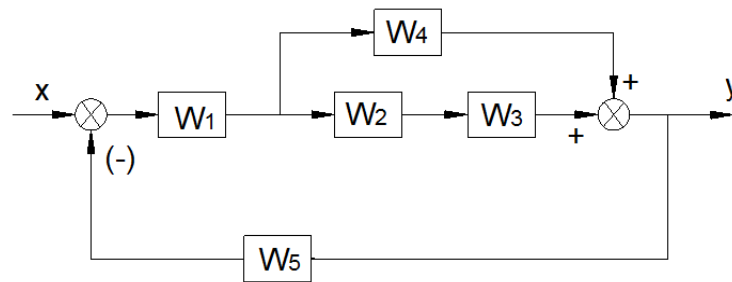
Hãy xây dựng hàm truyền hệ kín. Vẽ đồ thị đáp ứng của hệ thống với tín hiệu vào dạng bước nhảy

Chuyển mô hình hàm truyền này sang mô hình không gian trạng thái.

Bài 4: Cho hệ thống có sơ đồ khối như sau

$$W1 = 5s/(s-1) ; W2 = 1/(s-1);$$

$$W3 = 5 ; W4 = 1/(s+2); W5 = 4$$

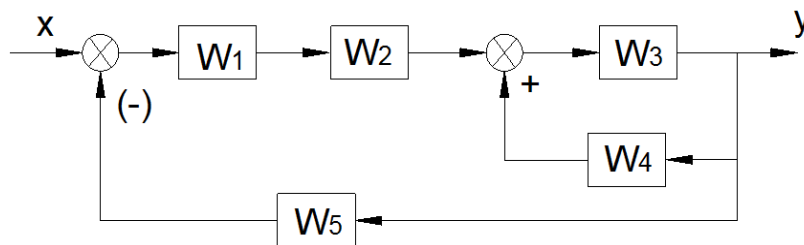


Hãy xây dựng hàm truyền hệ kín. Vẽ đồ thị đáp ứng của hệ thống

Bài 5: Cho hệ thống có sơ đồ khối như sau

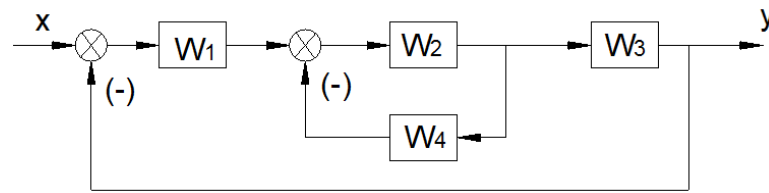
$$W1 = 5s/(s-1) ; W2 = 1/(s-1);$$

$$W3 = 5 ; W4 = 1/(s+2); W5 = 4$$



Hãy xây dựng hàm truyền hệ kín. Vẽ đồ thị đáp ứng của hệ thống

Bài 6: Cho hệ thống có sơ đồ khối như sau



$$W1 = 0,5 ; W2 = 1/(s+2)(s+8);$$

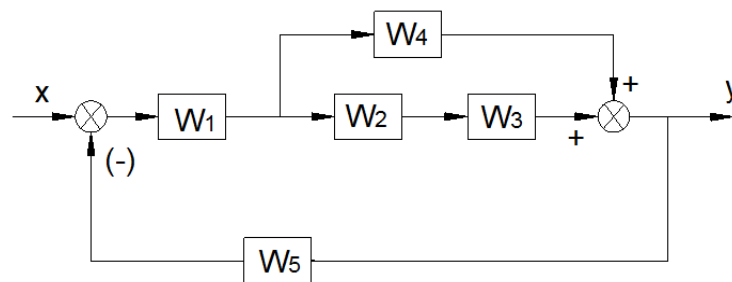
$$W3 = 1/s ; W4 = 2$$

Mô phỏng hệ thống trên bằng Matlab Simulink với tín hiệu vào là khối step

Bài 7: Cho hệ thống có sơ đồ khối như sau

$$W1 = 5s/(s-1) ; W2 = 1/(s-1);$$

$$W3 = 5 ; W4 = 1/(s+2); W5 = 4$$

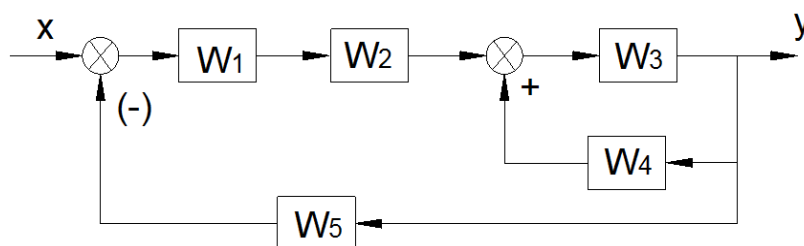


Mô phỏng hệ thống trên bằng Matlab Simulink với tín hiệu vào là khối step

Bài 8: Cho hệ thống có sơ đồ khối như sau

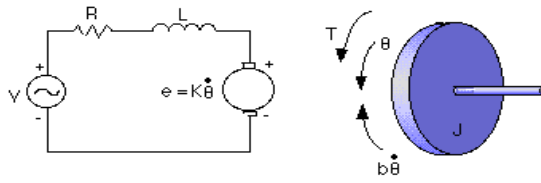
$$W1 = 5s/(s-1) ; W2 = 1/(s-1);$$

$$W3 = 5 ; W4 = 1/(s+2); W5 = 4$$



Mô phỏng hệ thống trên bằng Matlab Simulink với tín hiệu vào là khối step

Bài 9: Cho mạch điện phản ứng và sơ đồ tách của rotor động cơ DC như trong hình sau:



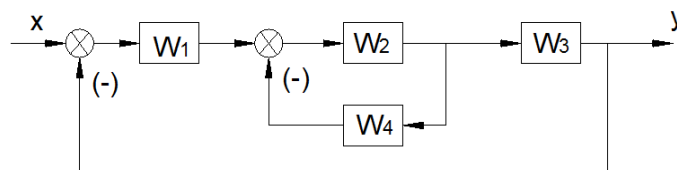
Giả thiết các thông số vật tư của hệ thống như sau:

- Moment quán tính của rotor (J) = $0.01 \text{ kg.m}^2/\text{s}^2$.
- Hệ số giảm chấn của hệ thống cơ khí (b) = 0.1 Nms .
- Hằng số điện cơ ($K=K_e=K_t$) = 0.01 Nm/Amp .
- Điện trở (R) = 1 ohm .
- Điện cảm (L) = 0.5 H .
- Input (V): điện áp nguồn.
- Output (θ): vị trí của trục.
- Giả thiết rotor và trục cứng vững tuyệt đối.

Sử dụng Matlab Simulink xây dựng mô hình điều khiển tốc độ động cơ điện một chiều với yêu cầu:

- Thời gian đạt vận tốc ổn định dưới 2 giây
- Quá điều khiển dưới 5%
- Sai số tĩnh dưới 1%

Bài 10: Cho hệ thống có sơ đồ khối như sau



$$W1 = 0,5 ; W2 = 1/(s+2)(s+8);$$

$$W3 = 1/s ; W4 = 2$$

Mô phỏng hệ thống trên bằng Matlab Simulink với tín hiệu vào là khối Constant

Câu hỏi trắc nghiệm

Câu 1: Kết nối hai hệ thống hồi tiếp sử dụng câu lệnh nào?

- A. feedback B. series C. parallel D. roots

Câu 2: Chuyển hệ thống từ hàm truyền thành dạng không gian trạng thái dùng câu lệnh nào?

- A. tf2ss.
B. ss2tf.
C. d2cm.
D. cmd2.

Câu 3: Chuyển từ mô hình không gian trạng thái sang mô hình hàm truyền đạt dùng câu lệnh nào?

- A. tf2ss.
B. ss2tf.
C. d2cm.
D. cmd2.

Câu 4: Xác định đáp ứng bước nhảy cho hệ có hàm truyền sau câu lệnh nào?

$$H(s) = \frac{s^2 + 3s - 1}{s^3 - 5s^2 + s + 3}$$

- A. step([1 3 -1],[1 -5 1 3])
B. step([1 3 -1])
C. step([[1 -5 1 3])
D. step([1 3 -1 1 -5 1 3])

Câu 5: Khi mô phỏng mạch điện tử nhất thiết phải có khối nào?

- A. Powergui
B. Scope
C. Step
D. Diode

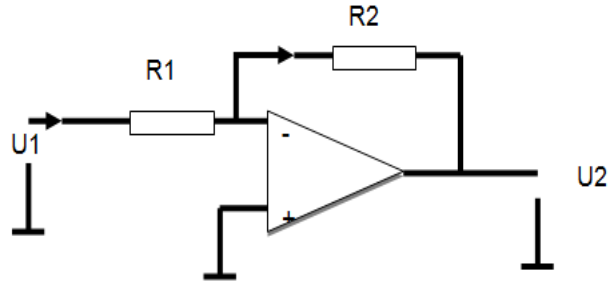
Tài liệu tham khảo

- [1] Nguyễn Phùng Quang (2004), *Matlab và Simulink dành cho kỹ sư điều khiển tự động*. NXBKHKHKT.
- [2] Nguyễn Hoàng Hải, Nguyễn Việt Anh (2006), *Matlab lập trình và ứng dụng*, NXBKHKHKT.
- [3] GS.TS. Nguyễn Công Hiền, TS. Phạm Thục Anh (2006), *Mô hình hóa hệ thống và mô phỏng*, NXBKHKHKT.
- [4] Trần Thu Hà, Hồ Đắc Lộc (2012), *Ứng dụng Matlab mô phỏng mạch điện – điện tử*, NXBKHKHKT

Các câu hỏi thường gặp

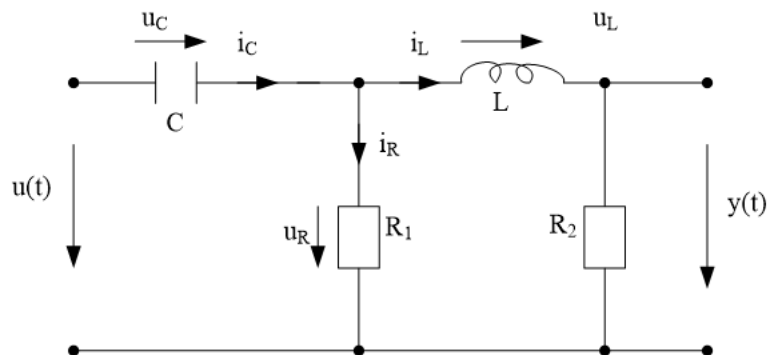
Câu 1: Thế nào là hệ thống tĩnh, thế nào là hệ thống động.

Cho mạch khuếch đại như sau:



Hãy tìm hàm truyền của hệ thống và cho biết hệ thống trên thuộc loại hệ thống tĩnh hay động.

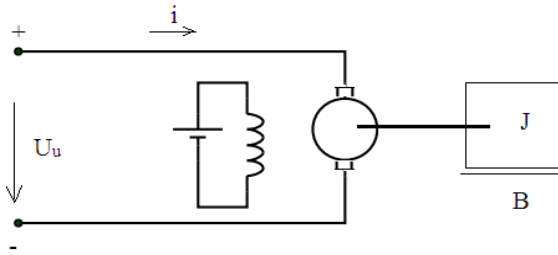
Câu 2: Cho một mạch điện biết trước các giá trị của C của tụ điện, L của cuộn dây, R_1 và R_2 của điện trở là các phần tử trong mạch điện. Hãy xác định hàm truyền đạt mô tả quan hệ giữa tín hiệu vào là điện áp $u(t)$ và tín hiệu ra $y(t)$ là điện áp trên R_2 của hệ thống mạch điện trên.



Câu 3: Mô hình vật lý là gì? Trình bày ưu nhược điểm và cho ví dụ về mô hình vật lý tương tự và mô hình vật lý thu nhỏ.

Câu 4: Hệ thống là gì phân biệt hệ thống liên tục và hệ thống gián đoạn.

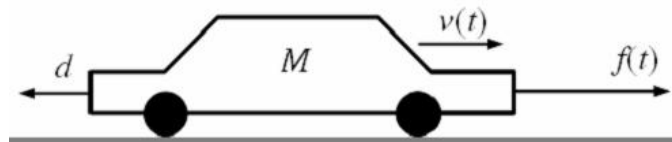
Câu 5: Cho hệ thống động cơ điện một chiều như hình vẽ



Với J là mô men quá tính quy đổi về trục động cơ; B là hệ số ma sát ở trục; giả sử từ thông không đổi.

Xác định hàm truyền đạt của động cơ với điện áp phần ứng U_u là tín hiệu vào và tín hiệu ra là tốc độ động cơ ω .

Câu 6: Tìm hàm truyền của ô tô sau



Với: M - khối lượng xe, B - hệ số ma sát; $f(t)$ - lực kéo; $v(t)$ - tốc độ xe

Câu 7: Sử dụng câu lệnh trong Matlab vẽ hai đồ thị trên hai hệ tọa độ khác nhau nhưng cùng một màn hình.

Với:

$$y_1 = \sin(x_1) \text{ và } y_2 = \sin(x_2) \cdot 2^{-x_2}$$

$$\text{Với } x_1 \in [0, 2\pi]; x_2 \in [-30, 30].$$

Câu 8: Sử dụng Matlab tính cách tích phân sau

$$a) \int \frac{dx}{x(1+\sqrt{x})^2} \quad b) \int \frac{x dx}{(x+1)^{1/2} - (x+1)^{1/3}}$$

$$c) \int_1^3 \frac{dx}{x\sqrt{x^2+5x+1}} \quad d) \int_0^\pi \frac{x \sin(x)}{2 \cos^2 x + 1} dx$$

Câu 9: Sử dụng Matlab tìm ảnh Laplace của các hàm gốc sau:

$$a) \frac{1}{2}(\sin t - \cos t) \quad b) \frac{\cos(bt) - \cos(at)}{t}$$

$$c) \frac{e^t - 1 - t}{t} \quad d) t - \frac{1}{2}e^{-t}$$

Câu 10: Cho các hàm ảnh Laplace sử dụng Matlab tìm các hàm gốc tương ứng

$$F1(s) = \frac{5s + 3}{(s-1)(s^2 + 2s + 5)}$$

$$F2(s) = \frac{s^2}{(s^2 + 4)(s^2 + 9)}$$

Câu 11: Sử dụng câu lệnh trong Matlab vẽ hai đồ thị sau trên cùng một cửa sổ đồ họa nhưng trên hai tọa độ khác nhau.

$$y1 = 10\sin(x) + \cos(x)$$

$$y2 = 4x^2 + 6x - 7$$

Với x thuộc khoảng [-5,5]

Câu 12: Sử dụng câu lệnh trong Matlab vẽ hai đồ thị sau trên cùng một cửa sổ đồ họa nhưng trên hai tọa độ khác nhau

$$y1 = 2\sin(2x) + \sin(x)\cos(x)$$

$$y2 = \frac{2x+3}{x^2 - 3x + 5}$$

Với x nằm trong khoảng [-20,20]

Câu 13: Sử dụng câu lệnh trong Matlab vẽ các đồ thị sau:

$$y1 = \sin(x) ; y2 = e^{-x} ; y3 = x^2 + x + 1 ; y4 = (x^2 + 1)^{1/2}$$

Với y1, y2 trên cùng một trục tọa độ và y3, y4 trên cùng một trục tọa độ khác nhưng cùng trên một cửa sổ đồ họa. Với x thuộc khoảng [-10, 10].

Câu 14: Cho mô hình không gian trạng thái của hệ thống với các ma trận tương ứng:

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 5 & 2 & 1 \\ 4 & 5 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 3 \\ 1 & 2 \\ 3 & 5 \end{bmatrix}$$

$$C = [0 \quad 1 \quad 0] \quad D = [0 \quad 0]$$

Chuyển mô hình không gian trạng thái này sang mô hình hàm truyền đạt

Vẽ đồ thị đáp ứng của hệ thống với tín hiệu vào dạng bước nhảy

Câu 15: Cho mô hình truyền của hệ thống như sau:

$$H(s) = \frac{s^2 + 3s - 1}{s^3 - 5s^2 + s + 3}$$

Chuyển mô hình hàm truyền đạt này sang mô hình không gian trạng thái

Xác định đáp ứng của hệ thống với tín hiệu vào dạng bước nhảy

Câu 16: Cho mô hình truyền của hệ thống như sau:

$$H(s) = \frac{s^2 + 3s - 1}{s^3 - 5s^2 + s + 3}$$

Chuyển mô hình hàm truyền đạt này sang mô hình không gian trạng thái

Xét ổn định cho hệ thống.

Câu 17: Cho hệ thống có hàm truyền như sau:

$$G(s) = \frac{2s + 2}{s^3 + 4s^2 + 3s}$$

Xác định đáp ứng của hệ thống với tín hiệu vào dạng bước nhảy

Dùng giản đồ Bode để khảo sát ổn định cho hệ thống dựa vào độ dự trữ biên, độ dự trữ pha.

Câu 18: Cho mô hình không gian trạng thái của hệ thống với các ma trận tương ứng:

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 5 & 2 & 1 \\ 4 & 5 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 3 \\ 1 & 2 \\ 3 & 5 \end{bmatrix}$$

$$C = [0 \quad 1 \quad 0] \quad D = [0 \quad 0]$$

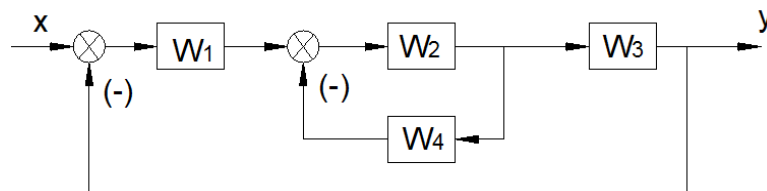
Chuyển đổi mô hình từ liên tục sang gián đoạn với $T_d=0.35$, thời gian lấy mẫu $T_s=0,1$

Câu 19: Cho mô hàm truyền của hệ thống như sau:

$$H(s) = \frac{s^2 + 3s - 1}{s^3 - 5s^2 + s + 3}$$

Chuyển đổi mô hình từ liên tục sang gián đoạn với $T_d=0.35$, thời gian lấy mẫu $T_s=0,1$

Câu 30: Cho hệ thống có sơ đồ khối như sau:



$$W1 = 0,5 \quad ; \quad W2 = 1/(s+2)(s+8);$$

$$W3 = 1/s \quad ; \quad W4 = 2$$

Hãy xây dựng hàm truyền hệ kín.

Vẽ đồ thị đáp ứng của hệ thống với tín hiệu vào dạng bước nhảy

Bài thực hành số 1 (Số tiết: 3 tiết) Khai báo biến và các thao tác với biến, ma trận và phép toán với biến và ma trận

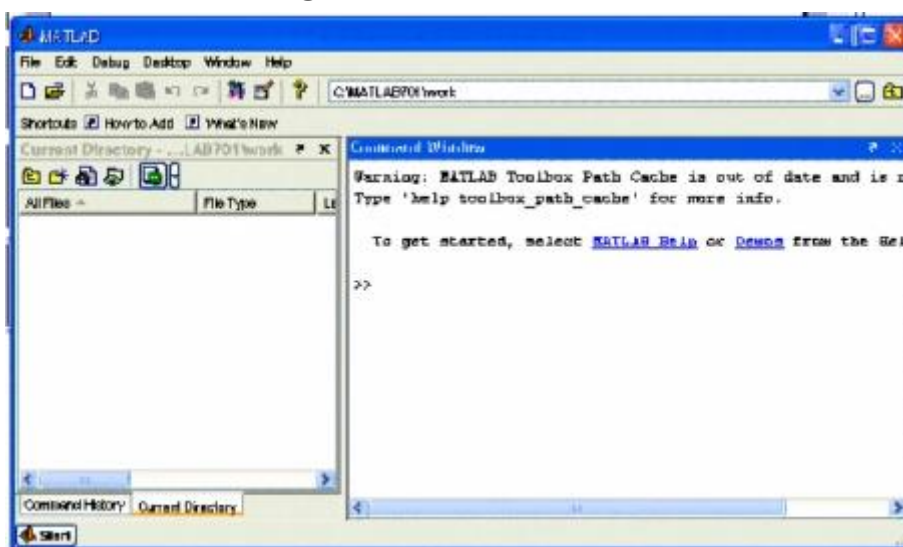
+ **Mục đích của bài thực hành:** Giúp sinh viên thực hiện thành thạo về khai báo biến và các thao tác với biến, ma trận và phép toán với biến và ma trận trên Matlab.

+ **Yêu cầu cần đạt được của bài thực hành:** Sinh viên sử dụng Matlab để khai báo biến và ma trận, sau đó thực hiện các phép toán với biến và ma trận. Tính toán kiểm tra lại kết quả.

+ **Bài giải mẫu**

1.1 Làm quen với môi trường làm việc của Matlab

1. Một số thao tác cơ bản trong Matlab



Hình 1.1: MATLAB 2012

Trong Matlab, thanh trình đơn thay đổi tùy theo cửa sổ lựa chọn.

Trình đơn *File*:

- *New*: Tạo đối tượng mới
- *Open*: Mở một file theo định dạng Matlab
- *Import data*: Nhập dữ liệu từ file khác vào Matlab.
- *Print*: in

Trình đơn *Window* để kích hoạt cửa sổ

- *Command window*: Cửa sổ lệnh
- *Current directory*: cửa sổ thư mục hiện hành
- *Command history*: Cửa sổ lịch sử
- *Work space*: Không gian làm việc

1. Các chức năng trên cửa sổ lệnh

Clc: Làm sạch màn hình

Clear: Làm sạch cửa sổ workspace

Help: Lấy thông tin trợ giúp

Edit: Soạn thảo chương trình

Type: Xem nội dung file

1.2 Biến và khai báo biến

Với MATLAB, bạn không cần khai báo biến trước khi sử dụng. Mỗi khi bạn gán một biến mới, MATLAB sẽ giúp bạn định nghĩa dựa vào dữ liệu biến ấy chứa. MATLAB hỗ trợ nhiều loại biến, thông thường bạn dùng MATLAB để tính toán với ma trận hoặc vector cả số học hay kí tự. Để tạo một biến, chỉ cần gán một giá trị vào cho một tên biến

```
>> var1=3.14
```

```
>> myString='hello world'
```

Quy tắc đặt tên biến:

- kí tự đầu tiên phải là **chữ cái**
- các kí tự sau có thể là chữ cái hoặc số, hay `_`
- MATLAB phân biệt chữ hoa với chữ thường (var1 khác với Var1)

Một số biến đã được MATLAB định nghĩa sẵn, không được dùng những tên biến này.

- i và j được dùng cho đơn vị phức
- pi : số pi, có giá trị 3.1415926...
- ans : lưu kết quả của phép toán vừa thực hiện
- Inf và -Inf là dương và âm vô cực
- NaN : thể hiện Not a Number, không là một số.

Biến vô hướng:

- Một biến vô hướng có thể được định nghĩa bằng cách gán giá trị trực tiếp:

```
>> a = 10
```

(biến này sẽ lập tức xuất hiện trên Workspace!)

```
>> c = 1.3*45-2*a
```

Nếu không muốn MATLAB đưa ra kết quả trên Command Window, bạn thêm dấu `;` vào cuối dòng lệnh.

```
>> d = 13/3;
```

1.3 Ma trận và các ma trận đặc biệt

Mảng là một thành phần rất quan trọng của MATLAB. Có 2 loại mảng:

- Mảng số: ma trận số thực hoặc số phức.

- Mảng cell: một cấu trúc dữ liệu trừu tượng, là mảng của các đối tượng.

Vector hàng: dùng dấu cách hoặc dấu phẩy để ngăn cách các phần tử trong dấu ngoặc vuông []

```
>> row = [1 2 5.4 -6.6] % hoặc sử dụng cách sau
```

```
>> row = [1,2,5.4,-6.6]
```

Cả hai cách đều cho cùng một kết quả.

Vector cột: dùng dấu chấm phẩy ngăn cách các phần tử trong dấu ngoặc vuông

```
>> col = [4;2;7;4]
```

Để xem kích thước các biến có thể dùng 3 cách

- Xem biến trong Workspace
- Gọi biến trong Command Window
- Dùng hàm size để xác định kích thước biến

Ma trận: Khởi tạo ma trận như với vector hàng và vector cột:

Có thể khởi tạo bằng cách liệt kê các phần tử của mảng

```
>> a = [1 2; 3 4]
```

Để lưu các biến đang có vào 1 file ta dùng lệnh save:

```
>> save myFile a b %câu lệnh này sẽ lưu 2 biến a và b vào file myFile.m, file này được tạo trong thư mục hiện thời.
```

Để xóa các biến hiện có, dùng câu lệnh clear. Ví dụ muốn xóa hai biến a, b:

```
>> clear a b
```

Để tải các biến đã lưu, dùng lệnh load

```
>> load a b
```

Có thể dùng các lệnh trên cho toàn bộ các biến như sau:

```
>> save myenv; clear all; load myenv;
```

Các phép toán

Các phép toán cơ bản (+ , - , * , /).

Ví dụ:

```
>> 7/45
```

```
>> (1+i)*(2-1)
```

Phép lũy thừa (^). Ví dụ:

```
>> 4^2
```

```
>>(3+e)^0.1
```

Với các biểu thức phức tạp, cần sử dụng các dấu ngoặc đơn.

```
>> ((2+3)*3)^(e-1)
```

- Chú ý để không bị thiếu dấu nhân, ví dụ:

>> 3(1+0.7) % MATLAB sẽ báo lỗi

- Các hàm toán học được định nghĩa sẵn (Built-in Functions)

>> sqrt(2) % Căn bậc hai

>> log(2); log10(1.2) % Hàm loga

>> cos(1.2), atan(-.8) % Các hàm lượng giác

>> round(1.2), floor(3.3), ceil(4.23) % Làm tròn số

>> exp(2+4*i) % Hàm lũy thừa cơ số tự nhiên

>> angle(i); abs(1+i); % Các phép toán với số phức

+ Các bài tập thực hành mức độ cơ bản

1. Tạo ma trận trong Matlab xác nhận kích thước của ma trận, trích xuất phần tử của ma trận

Bước 1: Tạo ma trận bằng cách nhập vào Matlab, mỗi phần tử trên một dòng được phân biệt bởi dấu khoảng trắng hoặc dấu phẩy, và một dòng mới được bắt đầu với dấu chấm phẩy.

```
>> A=[1 2 3 0;5 -1 0 0;3 -2 5 0]
A =
     1     2     3     0
     5    -1     0     0
     3    -2     5     0
```

Bước 2: Chúng ta có thể sử dụng lệnh size để xác định kích thước của ma trận A(3x4):

```
>> size(A)
ans =
     3     4
```

Bước 3: Chỉ số của ma trận trong Matlab giống như chỉ số của Vector. Điểm khác biệt ở đây là có hai chiều. Để truy xuất tới thành phần của dòng 2, cột 3 của ma trận A, sử dụng lệnh sau:

```
>> A(2,3)
ans =
     0
```

Có thể truy xuất một dòng của ma trận với dấu hai chấm trong Matlab. Lệnh A(2,:) sẽ lấy mọi cột của dòng số 2.

```
>> A(2,:)
ans =
     5    -1     0     0
```

Tương tự như vậy, có thể truy xuất đến bất kỳ cột nào của ma trận A với lệnh $A(:,x)$ với x là cột của ma trận.

```
>> A(:,2)
ans =
     2
    -1
    -2
```

Cần lấy ma trận con sử dụng dòng 1 và 3 và cột 2 và 4, ta sử dụng lệnh sau:

```
>> A([1,3],[2,4])
ans =
     2     0
    -2     0
```

Bước 4: Gán các phần tử của ma trận

```
>> A(3,4)=12
A =
     1     2     3     0
     5    -1     0     0
     3    -2     5    12
```

Khi muốn gán một dòng, một cột hay một ma trận con của A, ta phải thay thế nội dung với một dòng, cột hoặc ma trận con với số phần tử tương ứng. Ví dụ sau sẽ gán nội dung mới cho dòng đầu tiên của ma trận A:

```
>> A(1,:)=20:23
A =
    20    21    22    23
     5    -1     0     0
     3    -2     5    12
```

```
>> A(:,2)=11
A =
    20    11    22    23
     5    11     0     0
     3    11     5    12
```

2. Cho ma trận $A = [2 \ 4 \ 1 ; 6 \ 7 \ 2 ; 3 \ 5 \ 9]$ viết lệnh Matlab để:

- Gán cho vector x là dòng thứ nhất của A.
- Gán cho ma trận y là hai dòng còn lại (cuối) của A.
- Gán cho ma trận A dòng đầu tiên giá trị tăng dần từ 24-26.

+ Các bài thực hành mức độ nâng cao.

Hoán vị ma trận Có thể dễ dàng hoán vị ma trận giống như hoán đổi dòng thành cột trong vector. Ví dụ, thực hiện lệnh magic như sau:

```
>> A=magic(4)
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

Có thể dễ dàng tính A^T với lệnh như sau:

```
>> A.'
ans =
    16     5     9     4
     2    11     7    14
     3    10     6    15
    13     8    12     1
```

Phép nhân vô hướng

Matlab cho phép nhân ma trận với một số (gọi là phép nhân vô hướng):

Nếu A là ma trận:

$$A = 3 \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},$$

Ta thực hiện phép nhân với 3 như sau:

```
>> 3*A
ans =
     3     6     9
    12    15    18
    21    24    27
```

Phép cộng ma trận

Nếu hai ma trận có cùng chiều, ta có thể cộng ma trận theo từng thành phần như sau: Ví dụ: Cho hai ma trận A và B như sau:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

```
>> A=[1 1 1;2 2 2;3 3 3]; B=[1 1 1;1 1 1;1 1 1];
>> A+B
ans =
     2     2     2
     3     3     3
     4     4     4
```

Phép nhân ma trận - vector:

Giả sử ta có hệ phương trình 3 ẩn như sau:

$$\begin{cases} 2x + 3y + 4z = 6 \\ 3x + 2y = 4z = 8 \\ 5x - 3y + 8z = 1 \end{cases}$$

ta có thể viết lại thành dạng sau:

$$x \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} + y \begin{bmatrix} 3 \\ 2 \\ -3 \end{bmatrix} + z \begin{bmatrix} 4 \\ 4 \\ 8 \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix}$$

Ta đưa về dạng $Ax=b$

$$A = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 2 & 4 \\ 5 & -3 & 8 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 6 \\ 8 \\ 1 \end{bmatrix}$$

Đưa vào Matlab:

```
>> A=[1 2 -3;3 0 4;0 -2 2]; x=[1; -2; 3];
```

```
>> A*x  
ans =  
-12  
15  
10
```

Phép nhân: ma trận – ma trận

```
>> A=[1 2;3 4]; B=[1 -2;2 1];
```

```
>> A*B  
ans =  
5 0  
11 -2
```

1. Có các ma trận sau:

$$A = \begin{bmatrix} 3 & 3 \\ 2 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix}, \quad C = \begin{bmatrix} 3 & 1 \\ 5 & 8 \end{bmatrix}$$

Sử dụng các lệnh Matlab, thực hiện các thuộc tính sau (0 là ma trận Zero):

- $A + B = B + A$
- $(A + B) + C = A + (B+C)$
- $A + 0 = A$
- $A + (-A) = 0$
- $A(B+C) = AB + AC$
- $(A + B)C = AC + BC$

4. Có ma trận sau

$$A = \begin{bmatrix} 2 & 2 \\ 4 & 7 \end{bmatrix}, B = \begin{bmatrix} 3 & 1 \\ 8 & 9 \end{bmatrix}$$

Và giá trị vô hướng sau: $\alpha = 2$ và $\beta = -3$. Sử dụng Matlab tính các phép tính sau:

a) $(\alpha + \beta)A = \alpha A + \beta A$

b) $\alpha(A + B) = \alpha A + \alpha B$

c) $\alpha(\beta A) = (\alpha\beta)A$

d) $(\alpha A)B = \alpha(AB) = A(\alpha B)$

5. Cho $A = [2 \ 7 \ 9 \ 7 ; 3 \ 1 \ 5 \ 6 ; 8 \ 1 \ 2 \ 5; 1 \ 2 \ 3 \ 5]$, viết lệnh Matlab để

- Gán cho ma trận B là các cột ở vị trí chẵn
- Gán cho ma trận C là các dòng ở vị trí lẻ
- Gán lại A thành chuyển vị của nó
- Tính nghịch đảo mọi phần tử của A
- Lấy căn bậc hai mọi phần tử của A

Bài thực hành số 2 (Số tiết: 3 tiết) Thao tác với các hàm toán học

+ **Mục đích của bài thực hành:** Giúp sinh viên thực hiện thành thạo về đa thức, tính giá trị đa thức, tìm nghiệm của đa thức hay các phép tính với đa thức. Giúp sinh viên thực hành thành thạo với các biến logic, các hàm toán học giúp tính toán nhanh chóng và đơn giản với Matlab.

+ **Yêu cầu cần đạt được của bài thực hành:** Sinh viên sử dụng Matlab để khai báo đa thức và tính toán đa thức một cách thành thạo.

+ Bài giải mẫu

2.1 Đa thức và các hàm xử lý đa thức

2.1.1 Tìm nghiệm của đa thức

Ta có một đa thức: $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$

Khai báo đa thức trong Matlab:

```
>> p=[an an-1 ..... a0]
```

Để tính toán giá trị của đa thức tại một giá trị của x ta sử dụng hàm:

Polyval (p, x)

Ví dụ: $A(x) = 6x^3 - 3x^2 + 4x + 12$ tính giá trị A tại $x=4$

```
>> A=[6 -3 4 12]

A =

     6     -3     4    12

>> x=4

x =

     4

>> y=polyval(p,x)

y =

    28
```

Matlab có thể tìm các nghiệm của một đa thức bằng lệnh:

```
>> r = roots(p)
```

ví dụ: cho $p(x) = x^3 - 2x^2 + 4x + 12$ xác định nghiệm của đa thức khi $p(x) = 0$


```

>> p=[1 -2 4 12]

p =

     1     -2      4     12

>> a=roots(p)

a =

    1.6914 + 2.4118i
    1.6914 - 2.4118i
   -1.3828

```

Trong Matlab nếu biết trước nghiệm của đa thức có thể tìm được các hệ số của đa thức đó bằng lệnh:

```

>> p=poly(r)
>> poly(a)

ans =

    1.0000   -2.0000    4.0000   12.0000

```

2.1.2 Cộng đa thức

Matlab không cung cấp các hàm trực tiếp thực hiện phép cộng hai đa thức. Sử dụng phép cộng ma trận chỉ có tác dụng khi hai đa thức là hai vector có cùng kích thước.

Ví dụ: ta có hàm $f_1(x) = 3x^6 + 15x^5 - 10x^3 - 3x^2 + 15x - 40$
 $f_2(x) = 3x^3 - 2x - 6$

```

>> f1=[3 15 0 -10 -3 15 -40]

f1 =

     3     15     0    -10     -3     15    -40

>> f2=[0 0 0 3 0 -2 -6]

f2 =

     0     0     0     3     0     -2     -6

>> f1+f2

ans =

     3     15     0     -7     -3     13    -46

```

2.1.3 Nhân đa thức

Để nhân hai đa thức ta thực hiện cú pháp:

```
>> c=conv(a, b)
```

Trong đó: a và b là các vector hệ số của đa thức
c là vector hệ số của tích.

```

>> a=[1 12 -2]

a =

     1     12     -2

>> b=[1 2]

b =

     1     2

>> c=conv(a,b)

c =

     1     14     22     -4

```

2.1.4 Chia đa thức

Thực hiện chia đa thức theo cú pháp: [q, r]=deconv(u,v)

Trong đó: u là vector hệ số của các đa thức bị chia

v là vector hệ số của đa thức chia

q là vector hệ số của thương

r là vector hệ số của phần dư

```
>> [q,r]=deconv(a,b)
```

```
q =
```

```
1 10
```

```
r =
```

```
0 0 -22
```

2.1.5 Đạo hàm của đa thức

Để đạo hàm của đa thức ta thực hiện lệnh: `k=polyder(p)`

Trong đó: p là vector hệ số của đa thức

K là vector hệ số của đạo hàm

```
>> p=[3 -12 10 4]
```

```
p =
```

```
3 -12 10 4
```

```
>> r=polyder(p)
```

```
r =
```

```
9 -24 10
```

2.1.6 Nguyên hàm của đa thức

Matlab tính nguyên hàm của đa thức bởi lệnh: `g=polyint(h, k)`

Trong đó: h là vector hệ số của đa thức

g là vector hệ số của nguyên hàm

k là hằng số tích phân, mặc định là 0

Ví dụ:

2.1.7 Phân thức hữu tỉ

Đôi khi ta gặp những bài toán liên quan đến tỉ số giữa 2 đa thức người ta gọi đó là phân thức hữu tỉ, ví dụ như các hàm truyền hay hàm xấp xỉ pade có dạng như sau:

$$\frac{n(x)}{d(x)} = \frac{N_1x^m + N_2x^{m-1} + \dots + N_{m-1}x^0}{D_1x^n + D_2x^{n-1} + \dots + D_{m-1}x^0}$$

Trong Matlab phân thức cũng được mô phỏng bằng hai đa thức riêng.

2.2 Các phần tử cơ bản và hàm toán học

2.2.1 Các phần tử cơ bản

Chuỗi ký tự được đặt giữa 2 dấu nháy đơn

Chuỗi ký tự là một mảng nhiều ký tự. Ký tự được lưu dưới dạng mã ASCII. >>

```
name= 'Trường Đại học DL Công Nghệ Sài Gòn'
```

Có thể truy xuất đến từng phần tử chuỗi

```
>> fprintf('Trường tôi là %s\n', name(8:35));
```

Kết hợp các string tạo string mới

```
>> text1='Tôi học tại'; text=[text1 ' ' name];
```

Nhập string từ bàn phím:

```
>> str= input('Nhập vào một chuỗi','s');
```

2.2.2 Hàm toán học

Bảng 2.1: Các hàm toán học cơ bản

Hàm	Ý nghĩa
Round	Làm tròn về số nguyên gần nhất
fix	Làm tròn về 0
Floor	Làm tròn nhỏ hơn
ceil	Làm tròn lớn hơn
log(x)	Ln(x)
log10(x)	Log thập phân
pow2(x)	Luỹ thừa cơ số 2
nextpow2(N)	Tìm p: $2^p=N$
Exp(x)	Hàm e^x
Sqrt(x)	Căn bậc hai của x
Abs(x)	Modun của số phức x
Imag(x)	Phần ảo của x
Real(x)	Phần thực của x
Sign(x)	Dấu của x

+ Các bài tập thực hành mức độ cơ bản

1. Giải các phương trình và hệ phương trình sau đây:

a. $x^3 - 5x^2 + 32x + 10 = 22$

b. Giải hệ phương trình sau theo hai cách dùng ma trận hoặc dùng hàm solve

2. Tính định thức của ma trận sau:

$$\begin{bmatrix} 8 & 9 & 3 \\ 9 & 6 & 5 \\ 1 & 1 & 10 \end{bmatrix}$$

+ **Các bài thực hành mức độ nâng cao.**

Cho ma trận A như sau:

$$\begin{bmatrix} 3 & -2 & 7 \\ -5 & 9 & 21 \\ 12 & 4 & -3 \end{bmatrix}$$

Tính đa thức đặc trưng của ma trận A.

Bài thực hành số 3 (Số tiết: 3 tiết) Đồ họa trong Matlab

+ **Mục đích của bài thực hành:** Giúp sinh viên thực hiện vẽ đồ họa 2D, 3D trên nền MATLAB. Ứng dụng trong các bài toán cơ bản.

+ **Yêu cầu cần đạt được của bài thực hành:** Sinh viên sử dụng thành thạo MATLAB vẽ đồ họa 2D, 3D và quan sát nêu nhận xét.

+ **Bài giải mẫu**

3.1 Vẽ đồ thị trong 2D

Lệnh cơ bản: $plot(x, f(x))$

Với x : vec-tơ chứa miền giá trị của hàm f .

$f(x)$: các giá trị của f ứng với x .

Ví dụ 1.5.1. Vẽ đồ thị $y = \sin(x)$ từ $[0, 2\pi]$

```
>> x = 0 : pi/100 : 2*pi;
```

```
>> y = sin(x);
```

```
>> plot(x, y);
```

Chú thích trên đồ thị:

$text(x, y, '...')$: đặt một chú thích (trong dấu ' ') lên đồ thị tại tọa độ (x, y) .

$gtext('...')$: đặt chú thích lên đồ thị, vị trí được xác định bởi click chuột.

$title('...')$: tựa đề của đồ thị.

$xlabel('...')$: ghi nhãn cho trục Ox.

$ylabel('...')$: ghi nhãn cho trục Oy.

$hold on/off$: bật / tắt chế độ cho phép vẽ nhiều đồ thị trong cùng một hệ trục tọa độ.

Các tùy chỉnh về nét vẽ, dấu và màu sắc:

Lệnh: $plot(x, y, 'Nét vẽ_Dấu_Màu sắc')$

Ví dụ:

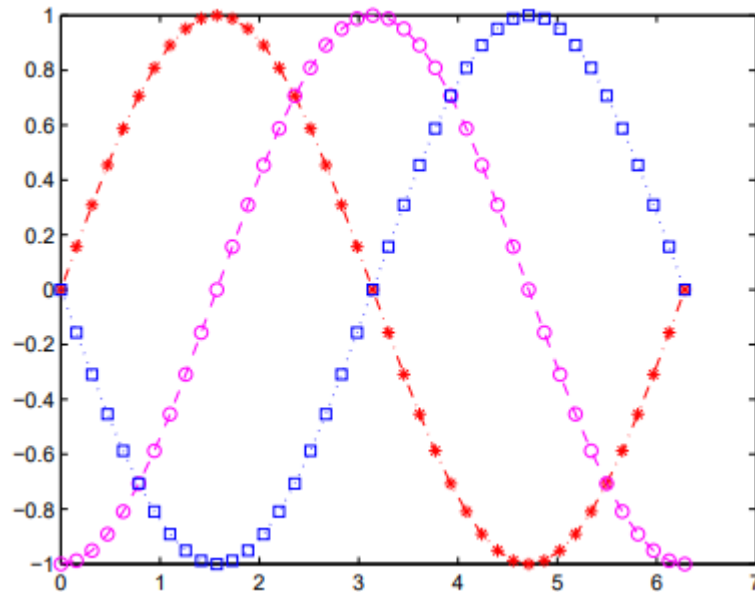
```
>> x = 0:pi/20:2*pi;
```

```
>> plot(x, sin(x), '-.*r');
```

```
>> hold on plot(x, sin(x - pi/2), '-om');
```

```
>> plot(x, sin(x - pi), ':bs');
```

```
>> hold off
```



Hình 3.1: Đồ thị hàm sin x

Xác định tọa độ:

`axis([xmin xmax ymin ymax])`

`xlim([xmin xmax])`

`ylim([ymin ymax])`

Tùy chỉnh các kiểu trục tọa độ:

`axis on/off/auto`

`axis normal/square/equal/tight`

`axis ij/xy`

`grid on/off`

`subplot` - Vẽ nhiều đồ thị trong cùng một cửa sổ:

`subplot(m, n, p)`: tạo ra một ma trận m hàng, n cột chứa $m \times n$ đồ thị, p là vị trí của từng đồ thị, thứ tự từ trên xuống dưới theo hàng.

3.2. Vẽ đồ thị trong 3D

Lệnh cơ bản: *plot3(x, y, z)*

Trong *plot3*, ta cần xác định các vec-tơ (x, y, z).

Để vẽ mặt (x, y, z = f(x, y)), sử dụng lệnh *meshgrid(x,y)*.

Một số lệnh vẽ đồ thị trong 3-D khác:

- *contour / contourf / contour3*

- *mesh / meshc / meshz*

- *surf / surfc*

- *waterfall*

- *bar3 / bar3h*

- *pie3 / fill3*

- *comet3 / scatter3 / stem3*

+ **Các bài tập thực hành mức độ cơ bản**

1. Vẽ 4 đồ thị trong cùng 1 cửa sổ

```
>> t = 0:pi/20:2*pi; [x,y] = meshgrid(t);
```

```
>> subplot(2,2,1)
```

```
>> plot(sin(t),cos(t))
```

```
>> axis equal
```

```
>> subplot(2,2,2)
```

```
>> z = sin(x)+cos(y);
```

```
>> plot(t,z)
```



```
>> axis([0 2*pi -2 2])
```

```
>> subplot(2,2,3)
```

```
>> z = sin(x).*cos(y);
```

```
>> plot(t,z)
```

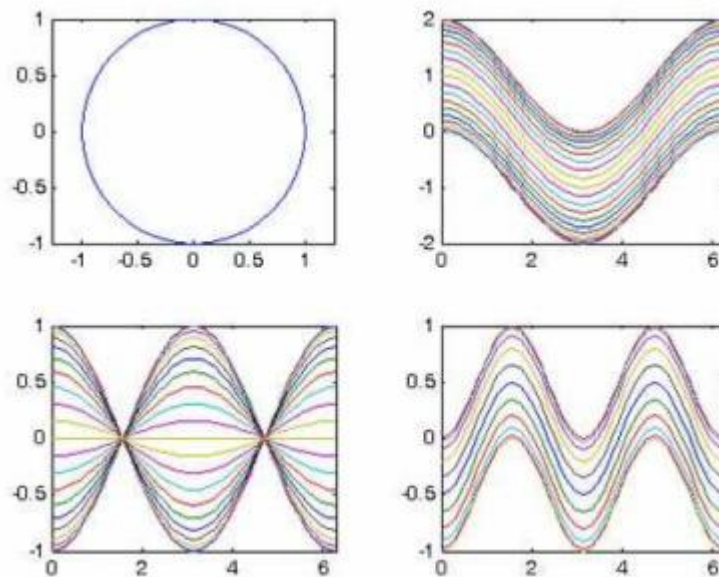
```
>> axis([0 2*pi -1 1])
```

```
>> subplot(2,2,4)
```

```
>> z = (sin(x).^2)-(cos(y).^2);
```

```
>> plot(t,z)
```

```
>> axis([0 2*pi -1 1])
```



Hình 3.2: Vẽ 4 đồ thị trong cùng 1 cửa sổ

+ Các bài thực hành mức độ nâng cao.

2. Vẽ đồ thị 3D biết hàm $x = \sin(t)$; $y = \cos(t)$; $z = t$;

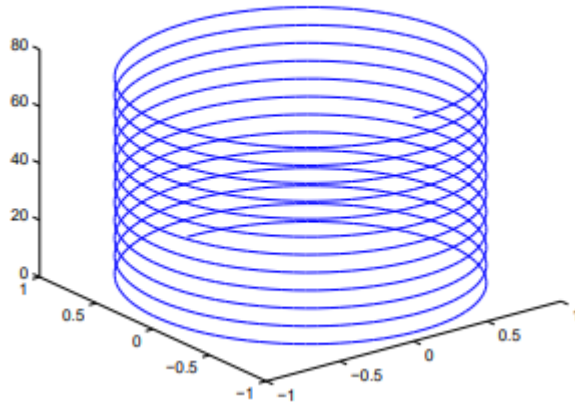
```
>> t = 0:0.02*pi:25*pi;
```

```
>> x = sin(t);
```

```
>> y = cos(t);
```

```
>> z = t;
```

```
>> plot3(x,y,z);
```



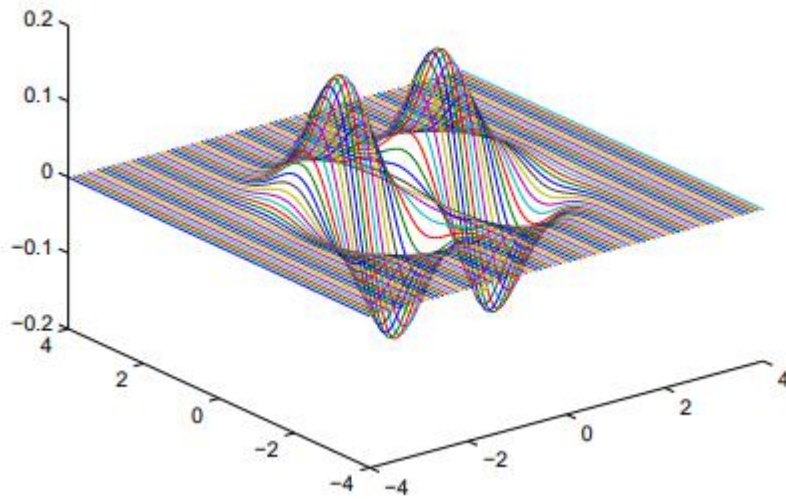
Hình 3.3: Đồ họa 3D

3. Vẽ mặt $z(x, y) = x^2ye^{-x^2-y^2}$ với $-4 \leq x \leq 4$ và $-4 \leq y \leq 4$.

```
>> [x,y]=meshgrid([-4:0.1:4]);
```

```
>> z=x.*x.*y.*exp(-x.^2-y.^2);
```

```
>> plot3(x,y,z)
```



Hình 3.4: Đồ họa 3D $z(x,y)$

Bài thực hành số 4 (Số tiết: 3 tiết) Mô phỏng với Matlab/Simulink

+ **Mục đích của bài thực hành:** Giúp sinh viên thực hiện thành thạo việc xây dựng mô hình mô phỏng trên Matlab/Simulink.

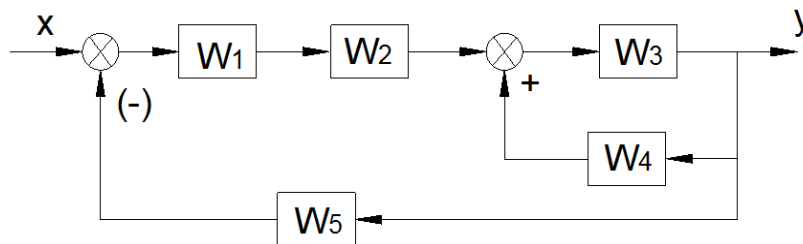
+ **Yêu cầu cần đạt được của bài thực hành:** Sinh viên biết cách xây dựng mô hình mô phỏng bằng việc sử dụng các thư viện trong Simulink.

+ Bài giải mẫu

Bài 1: Cho hệ thống có sơ đồ khối như sau:

$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$



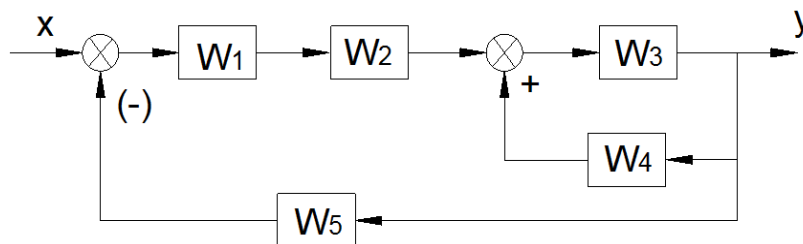
Sử dụng bộ điều khiển PID trong Matlab/Simulink để điều khiển hệ thống trên.

+ Các bài tập thực hành mức độ cơ bản

Bài 1: Cho hệ thống có sơ đồ khối như sau:

$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$



Xác định đáp ứng của hệ thống trên với tín hiệu vào có dạng Constant.

Bài 2: Cho lò nhiệt điện trở có hàm truyền đạt như sau:

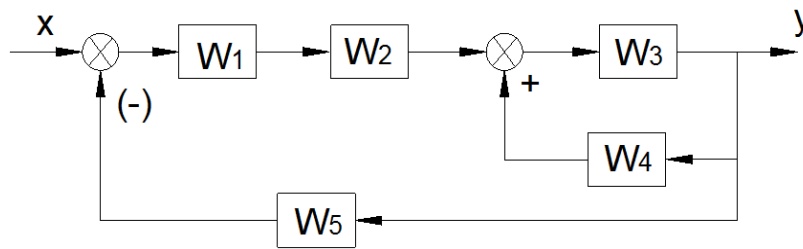
$$W(s) = \frac{K.e^{-T_2s}}{T_1s + 1}$$

Với $K=100, T_1=250, T_2=50$. Sử dụng bộ điều khiển PID trong Matlab/Simulink để điều khiển đối tượng trên.

Bài 3: Cho hệ thống có sơ đồ khối như sau:

$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$

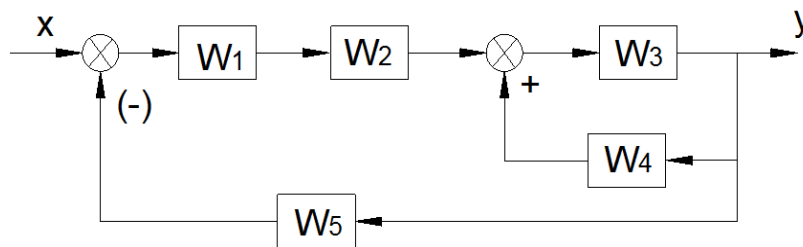


Sử dụng bộ điều khiển PID trong Matlab/Simulink để điều khiển hệ thống trên.

Bài 4: Cho hệ thống có sơ đồ khối như sau:

$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$



Xác định đáp ứng của hệ thống trên với tín hiệu vào có dạng Step.

+ Các bài thực hành mức độ nâng cao.

Bài 1: Cho đối tượng có mô hình không gian trạng thái, với các ma trận A, B, C, D được cho như sau :

$$A = \begin{bmatrix} 0 & 0,3 \\ -0,5 & -0,2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$C = [0 \quad 1] \quad D = [0]$$

Xây dựng mô hình mô phỏng đối tượng trên simulink với tín hiệu vào có dạng Step

Bài 2: Cho đối tượng có mô hình không gian trạng thái, với các ma trận A, B, C, D được cho như sau :

$$A = \begin{bmatrix} 0 & 0,3 \\ -0,5 & -0,2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$C = [0 \quad 1] \quad D = [0]$$

Xây dựng mô hình mô phỏng đối tượng trên simulink với tín hiệu vào có dạng Constant.

Bài 3: Cho đối tượng có mô hình không gian trạng thái, với các ma trận A, B, C, D được cho như sau :

$$A = \begin{bmatrix} 0 & 0,3 \\ -0,5 & -0,2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$C = [0 \quad 1] \quad D = [0]$$

Xây dựng mô hình mô phỏng hệ thống điều khiển đối tượng trên sử dụng bộ điều khiển PID trong simulink với tín hiệu vào có dạng Step

Bài 4: Cho đối tượng có mô hình không gian trạng thái, với các ma trận A, B, C, D được cho như sau :

$$A = \begin{bmatrix} 0 & 0,3 \\ -0,5 & -0,2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$C = [0 \quad 1] \quad D = [0]$$

Xây dựng mô hình mô phỏng hệ thống điều khiển đối tượng trên sử dụng bộ điều khiển PID 2 bậc trong simulink với tín hiệu vào có dạng Step

Bài thực hành số 5 (Số tiết: 3 tiết) Mô phỏng với Matlab/Simulink (Tiếp)

+ **Mục đích của bài thực hành:** Giúp sinh viên thực hiện thành thạo việc xây dựng một mô hình mô phỏng trên Matlab/Simulink.

+ **Yêu cầu cần đạt được của bài thực hành:** Sinh viên biết cách xây dựng mô hình mô phỏng bằng việc kết nối sử dụng các thư viện trong Simulink.

+ Bài giải mẫu

Bài tập 1: Xây dựng mô hình mô phỏng Mạch chỉnh lưu 1 pha, $\frac{1}{2}$ chu kỳ, không điều khiển, tải R-L-E:

Sinh viên điều chỉnh các trường hợp sau:

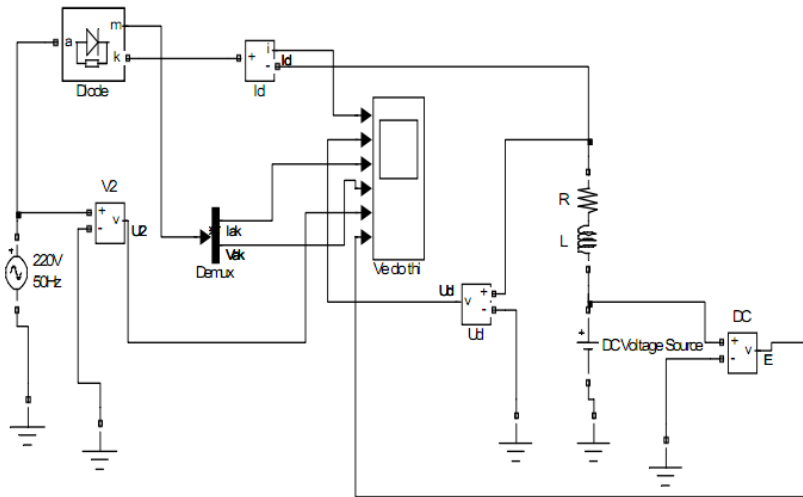
- + Tải R.
- + Tải R-L.
- + Tải R-E.
- + Tải L-E.
- + Tải R-L-E.

Kết hợp với điều kiện:

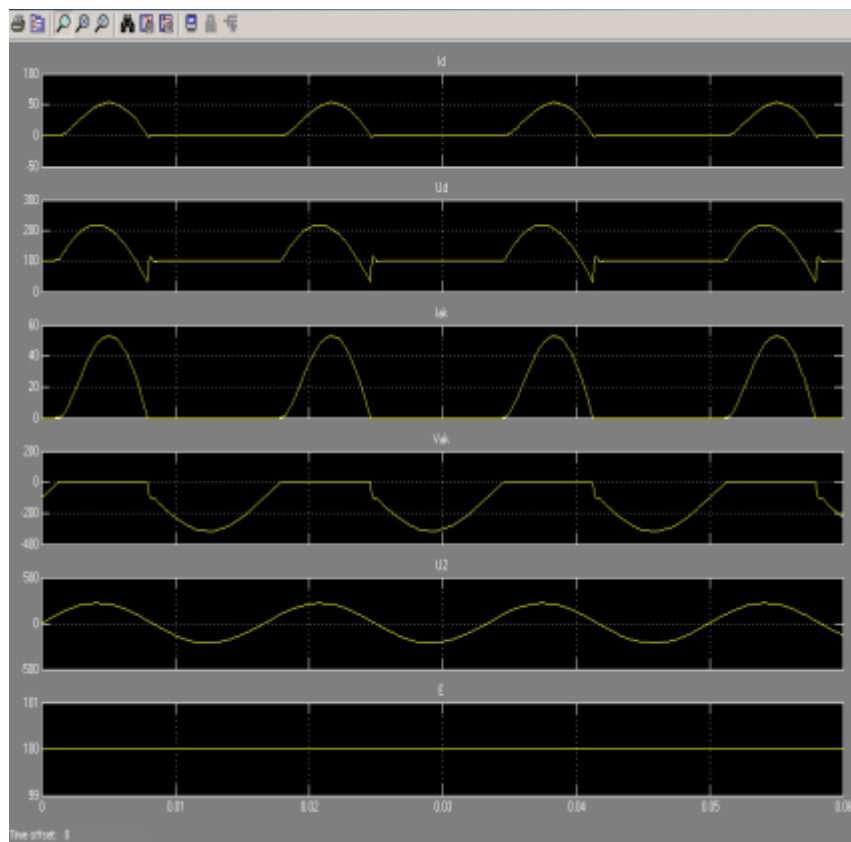
- + Có diode đệm D0.
- + Không có diode đệm D0.

Thực hiện mô phỏng - Vẽ đồ thị mỗi trường hợp.

Xác định kết quả, đối chiếu lý thuyết và đưa ra kết luận.



Kết quả:



Bài tập 2: Xây dựng mô hình mô phỏng Mạch chỉnh lưu 1 pha, $\frac{1}{2}$ chu kỳ, có điều khiển, tải R-L-E:

Sinh viên điều chỉnh các trường hợp sau:

- + Tải R.
- + Tải R-L.
- + Tải R-E.
- + Tải L-E.
- + Tải R-L-E.

Điều khiển, tải R-L-E:

Sinh viên điều chỉnh các trường hợp sau:

- + Tải R.
- + Tải R-L.
- + Tải R-E.
- + Tải L-E.
- + Tải R-L-E.

Kết hợp với điều kiện:

- + Có diode đệm D0.
- + Không có diode đệm D0.

Bài tập 2: Xây dựng mô hình mô phỏng Mạch chỉnh chỉnh lưu 1 pha, hình cầu, có điều khiển, tải R-L-E:

Sinh viên điều chỉnh các trường hợp sau:

- + Tải R.
- + Tải R-L.
- + Tải R-E.
- + Tải L-E.
- + Tải R-L-E.

Kết hợp với điều kiện:

- + Có diode đệm D0.
- + Không có diode đệm D0.

Bài tập 3: Xây dựng mô hình mô phỏng Mạch chỉnh chỉnh lưu 3 pha, hình tia, không điều khiển, tải R-L-E:

Sinh viên điều chỉnh các trường hợp sau:

- + Tải R.
- + Tải R-L.
- + Tải R-E.
- + Tải L-E.
- + Tải R-L-E.

Kết hợp với điều kiện:

- + Có diode đệm D0.
- + Không có diode đệm D0.

+ Các bài thực hành mức độ nâng cao.

Bài tập 1: Xây dựng mô hình mô phỏng Mạch chỉnh chỉnh lưu 3 pha, hình tia, có điều khiển, tải R-L-E:

Sinh viên điều chỉnh các trường hợp sau:

- + Tải R.
- + Tải R-L.
- + Tải R-E.
- + Tải L-E.
- + Tải R-L-E.

Kết hợp với điều kiện:

- + Có diode đệm D0.
- + Không có diode đệm D0.

Bài tập 2: Cho mạch điện như hình vẽ với

$$e_1 = 200 \text{ (V)}$$

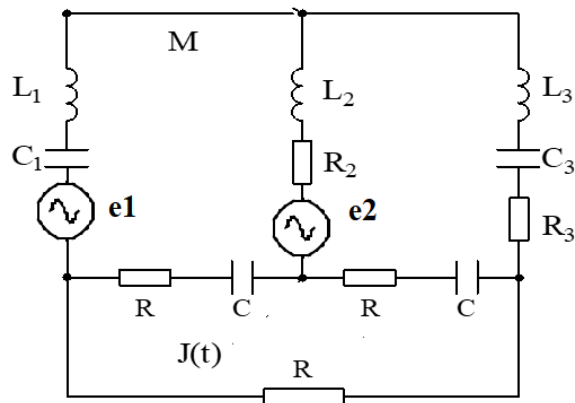
$$e_2 = 220 \text{ (V)}$$

$$j = 2 \sin \omega t \text{ (V)}$$

$$R_2 = R_3 = 80 \Omega, R = 50 \Omega.$$

$$L_1 = L_2 = 1.5 \text{ mH}; L_3 = 1.2 \text{ mH}$$

$$C_1 = C_3 = 200 \mu\text{F}; C = 100 \mu\text{F} \quad f = 50\text{Hz}$$



Hãy sử dụng các khối trong Simulink để vẽ mạch điện trên sử dụng khối đo dòng để xác định dòng điện chạy trong các nhánh

Bài thực hành số 6 (Số tiết: 3 tiết) Ứng dụng Matlab trong điều khiển

+ **Mục đích của bài thực hành:** Giúp sinh viên có công cụ để tiến hành phỏng các đối tượng, hệ thống nhằm kiểm chứng lại nội dung lý thuyết đã học. Làm cơ sở để nghiên cứu sâu hơn về các đối tượng và hệ thống.

+ **Yêu cầu cần đạt được của bài thực hành:** Sinh viên sử dụng Cách sử dụng phần mềm Matlab để mô tả đặc điểm của hệ thống và xây dựng mô hình hệ thống, chuyển đổi mô hình, khảo sát ổn định và các đáp ứng của hệ thống.

+ Bài giải mẫu

Bài tập 1: Thực hành đánh giá khả năng điều khiển được, quan sát được của hệ thống

Xác định khả năng điều khiển của hệ không gian trạng thái ở ví dụ về lệnh dcgrain

$$a = [-0.5572 \quad -0.7814; 0.7814 \quad 0];$$

$$b = [1 \quad 0.5379; 0 \quad -0.2231];$$

$$c = [1.9691 \quad 6.4439; 1 \quad 0];$$

$$d = [0 \quad 0; 0 \quad 0];$$

$$Gc = \text{gram}(a,b)$$

Ta nhận được ma trận:

$$Gc =$$

$$1.2016 \quad -0.0318$$

$$-0.0318 \quad 1.0708$$

Tìm hạng ma trận bằng lệnh:

$$r = \text{rank}(Gc)$$

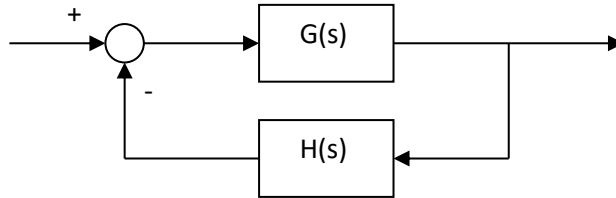
Ta được $r = 2$ và bằng kích thước của ma trận đánh giá. Vậy hệ thống này có thể điều khiển được.

Bài tập 2: Thực hành các lệnh để kết nối các khâu trong hệ thống.

Ví dụ 1:

Kết nối khâu có hàm truyền $G(s) = \frac{2s^2 + 5s + 1}{s^2 + s + 3}$ với khâu hồi tiếp có hàm truyền

$H(s) = \frac{5(s + 2)}{s + 10}$ theo dạng hồi tiếp âm như sau:



numg = [2 5 1];

deng = [1 2 3];

numh = [5 10];

denh = [1 10];

[num,den] = feedback(numg, deng, numh, denh);

Kết quả:

num =

2 25 51 10

den =

11 57 78 40

Ví dụ 2:

Kết nối 2 khâu có hàm truyền $G(s)$ và $H(s)$ để tạo thành hệ thống nối tiếp.

$$G(s) = \frac{3}{s + 4}, \quad H(s) = \frac{2s + 4}{s^2 + 2s + 3}$$

Ta thực hiện như sau:

num1 = 3;

den1 = [1 4];

num2 = [2 4];

den2 = [1 2 3];

[num,den] = series(num1,den1, num2,den2)

ta được kết quả:

num = [0 0 6 12]

den = [1 6 11 12]

Ví dụ 3:

Nối 2 khâu có hàm truyền $G(s)$ và $H(s)$ thành hệ thống song song:

$$G(s) = \frac{3}{s + 4}$$

$$H(s) = \frac{2s + 4}{s^2 + 2s + 4}$$

$$\text{numg} = 3;$$

$$\text{deng} = [1 \quad 4];$$

$$\text{numh} = [2 \quad 4];$$

$$\text{denh} = [1 \quad 2 \quad 3];$$

$$[\text{num}, \text{den}] = \text{parallel}(\text{numg}, \text{deng}, \text{numh}, \text{denh});$$

Ta được hệ thống song song có hàm truyền

$G'(s) = \text{num}(s)/\text{den}(s)$ với các hệ số:

$$\text{num} = [0 \quad 5 \quad 18 \quad 25]$$

$$\text{den} = [1 \quad 6 \quad 11 \quad 12]$$

+ Các bài tập thực hành mức độ cơ bản

Bài tập 1: Thực hành chuyển đổi mô hình từ liên tục sang gián đoạn.

Cho hệ thống: $H(s) = (s - 1)/(s^2 + 4s + 5)$

Với $T_d=0.35$, thời gian lấy mẫu $T_s=0,1$

$$\gg \text{num}=[1 \ -1];$$

$$\gg \text{den}=[1 \ 4 \ 5];$$

$$\gg H=\text{tf}(\text{num}, \text{den}, \text{'inputdelay'}, 0.35)$$

Kết quả:

Transfer function:

$$\frac{s - 1}{s^2 + 4s + 5} \exp(-0.35s)$$

$$\gg \text{Hd}=\text{c2d}(H, 0.1, \text{'foh'})$$

Transfer function:

$$\frac{0.0115 z^3 + 0.0456 z^2 - 0.0562 z - 0.009104}{z^3 - 1}$$

$$z^3 - 1.629 z^2 + 0.6703 z$$

Sampling time: 0.1

+ Các bài thực hành mức độ nâng cao.

Bài tập 1: Thực hành với câu lệnh chuyển đổi mô hình từ gián đoạn sang liên tục.

1:Lệnh C2DM

a) Công dụng:

Chuyển đổi hệ liên tục sang gián đoạn.

b) Cú pháp:

[ad,bd,cd,dd] = c2dm(a,b,c,d,Ts,'method')

[numd,dend] = c2dm(num,den,Ts,'method').

c) Ví dụ:

Cho hệ thống có mô hình không gian trạng thái như sau :

Chuyển hệ không gian trạng thái liên tục thành hệ gián đoạn dùng phương pháp 'Tustin', vẽ 2 đồ thị đáp ứng so sánh.

a = [1 1; 2 -1];

b = [1; 0];

c = [2 4];

d = 1;

Ts = 1;

[ad,bd,cd,dd] = c2dm(a,b,c,d,Ts,'tustin')

c2dm(a,b,c,d,Ts,'tustin') %vẽ đồ thị so sánh

title ('Do thi so sanh 2 dap ung lien tục va gian doan')

grid on

Kết quả: Ta được đồ thị và các giá trị như sau:

ad =

11 4

8 3

bd =

6

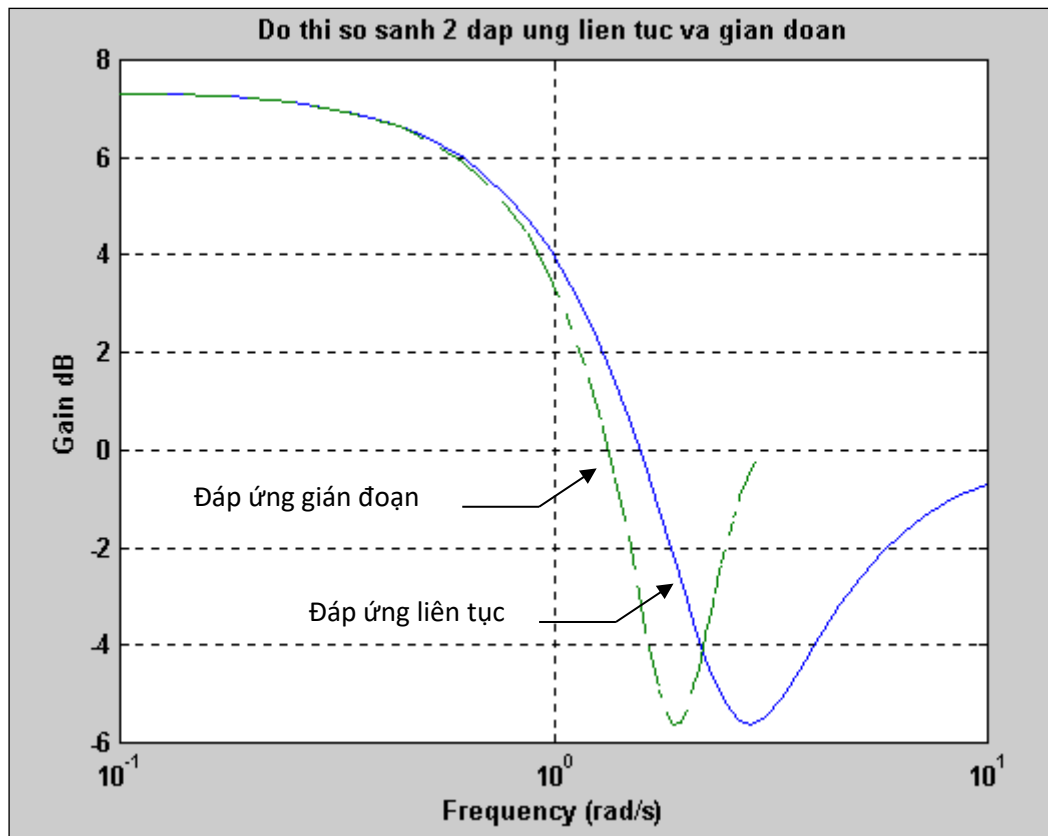
4

cd =

28 12

dd =

15



Bài tập 2: Thực hành với các lệnh chuyển đổi mô hình không gian trạng thái từ gián đoạn sang liên tục.

1. Lệnh D2CM

a) Công dụng:

Chuyển đổi mô hình không gian trạng thái từ gián đoạn sang liên tục.

b) Cú pháp:

$$[ac, bc, cc, dc] = d2cm(a, b, c, d, Ts, 'method')$$

$$[numc, denc] = d2cm(num, den, Ts, 'method')$$

c) Ví dụ:

Cho hệ thống gián đoạn có mô hình không gian trạng thái như sau :

$$x[n+1] = Ax[n] + Bu[n]$$

$$y[n] = Cx[n] + Du[n]$$

với:

$$A = \begin{bmatrix} 11 & 4 \\ 8 & 3 \end{bmatrix}; \quad B = \begin{bmatrix} 6 \\ 4 \end{bmatrix}; \quad C = [28 \quad 12] \quad D = 15;$$

Chuyển hệ không gian trạng thái gián đoạn:

A = [11 4; 8 3];

B = [6; 4];

C = [28 12];

D = 15;

Ts = 1;

[ac, bc, cc, dc] = d2cm(a, b, c, d, Ts, 'tustin')

d2cm(a, b, c, d, Ts, 'tustin') % vẽ đồ thị so sánh

title ('Do thi so sanh 2 dap ung lien tục va gian đoạn')

ta được đồ thị và các tham số như sau:

ac =

1 1

2 -1

bc =

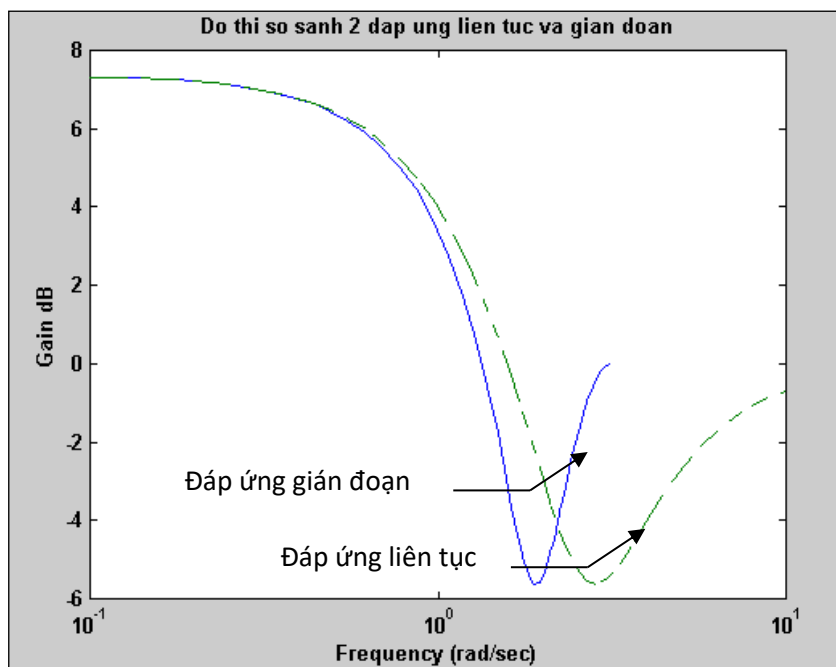
1

0

cc =

2 4

dc = 1



Bài tập 3: Thực hành với các lệnh chuyển hệ thống từ dạng không gian hàm truyền thành dạng trạng thái.

1. Lệnh TF2SS

a) Công dụng:

Chuyển hệ thống từ dạng không gian hàm truyền thành dạng trạng thái.

b) Cú pháp:

$$[a,b,c,d] = \text{tf2ss}(\text{num},\text{den})$$

c) Ví dụ:

Xét hệ thống có hàm truyền:

$$H(s) = \frac{2s+3}{s^2+2s+1} \cdot \frac{1}{s^2+0.4s+1}$$

Chuyển hệ thống thành dạng không gian trạng thái :

$$\text{Num} = [0 \ 2 \ 3; \ 1 \ 2 \ 3];$$

$$\text{den} = [1 \ 0.4 \ 1];$$

$$[a,b,c,d] = \text{tf2ss}(\text{num},\text{den});$$

ta được kết quả:

$$a =$$

$$\begin{matrix} -0.4000 & -1.0000 \\ 1.0000 & 0 \end{matrix}$$

$$b =$$

$$1$$

$$0$$

$$c =$$

$$\begin{matrix} 2.0000 & 3.0000 \\ 1.0000 & 2.0000 \end{matrix}$$

$$d =$$

$$0$$

$$1$$

Ví dụ 2: Cho hàm truyền: $(s^2+7s+2) / (s^3+9s^2+26s+24)$

Chuyển hệ thống thành dạng không gian trạng thái :

$$\gg \text{num}=[1 \ 7 \ 2];$$

$$\gg \text{den}=[1 \ 9 \ 26 \ 24];$$

$$\gg [A,B,C,D]=\text{tf2ss}(\text{num},\text{den})$$

Bài thực hành số 7 (Số tiết: 3 tiết) Ứng dụng Matlab trong điều khiển (Tiếp)

+ **Mục đích của bài thực hành:** Giúp sinh viên có công cụ để tiến hành mô phỏng các đối tượng, hệ thống nhằm kiểm chứng lại nội dung lý thuyết đã học. Làm cơ sở để nghiên cứu sâu hơn về các đối tượng và hệ thống.

+ **Yêu cầu cần đạt được của bài thực hành:** Sinh viên sử dụng Cách sử dụng phần mềm Matlab để mô tả đặc điểm của hệ thống và xây dựng mô hình hệ thống, chuyển đổi mô hình, khảo sát ổn định và các đáp ứng của hệ thống.

+ Bài giải mẫu

Bài tập 1: Vẽ đáp ứng tần số Bode trong Matlab

Ví dụ:

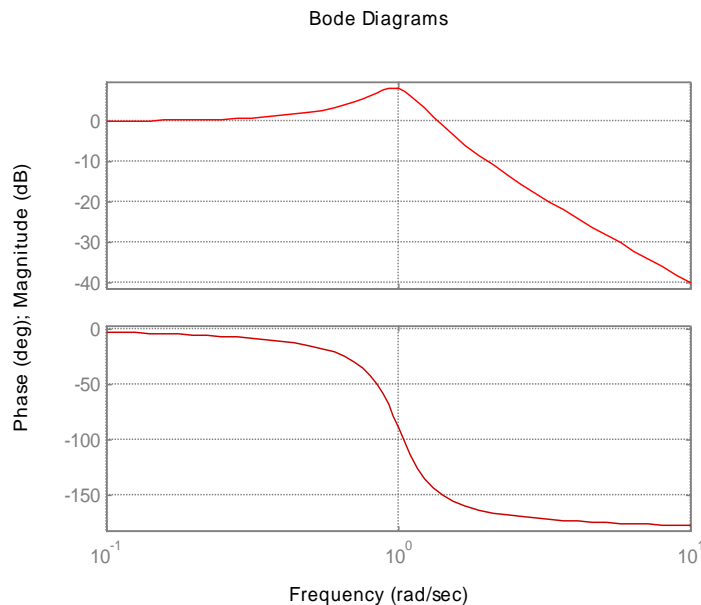
Vẽ đáp ứng biên độ và pha của hệ bậc 2 với tần số tự nhiên $\omega_n = 1$ và hệ số tắt dần $\zeta = 0.2$

```
[a,b,c,d] = ord2(1,0.2);
```

```
bode(a,b,c,d)
```

```
grid on
```

và ta được giản đồ Bode đáp ứng tần số của hệ thống như sau:



Bài tập 2: Thực hành vẽ đáp ứng của phép biến đổi Laplace.

1. Lệnh FREQS

a) Công dụng:

Tìm đáp ứng tần số của phép biến đổi Laplace.

b) Cú pháp:

```
h = freqs(b,a,w)
```

```
[h,w] = freqs(b,a)
```

[h,w] = freqs(b,a,n)

freqs(b,a)

c) Ví dụ:

Tìm và vẽ đáp ứng tần số của hệ thống có hàm truyền:

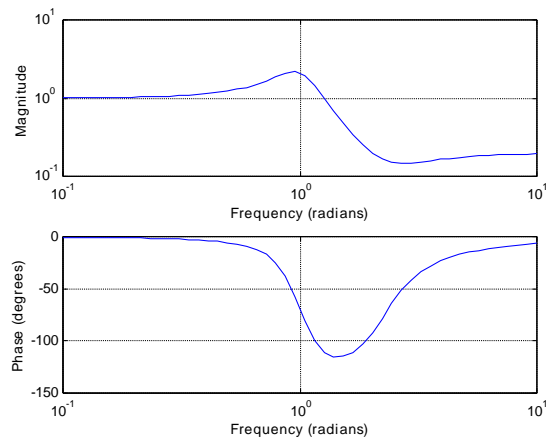
$$H(s) = \frac{0.2s^2 + 0.3s + 1}{s^2 + 0.4s + 1}$$

% Khai báo hàm truyền:

a = [1 0.4 1]; b = [0.2 0.3 1];

% Xác định trục tần số: w = logspace(-1,1);

% Thực hiện vẽ đồ thị: freqs(b,a,w)



Bài tập 3: Thực hành tìm đáp ứng bước nhảy

Lệnh STEP

a) Công dụng:

Tìm đáp ứng nấc đơn vị.

b) Cú pháp:

[y,x,t] = step(a,b,c,d)

[y,x,t] = step(a,b,c,d,iu)

[y,x,t] = step(a,b,c,d,iu,t)

[y,x,t] = step(num,den)

[y,x,t] = step(num,den,t)

c) Ví dụ:

Vẽ đồ thị đáp ứng nấc của hệ không gian trạng thái bậc 2 sau:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -0.5572 & -0.7814 \\ 0.7814 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

$$y = [1.9691 \quad 6.4493] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0] u$$

```
a = [-0.5572 -0.7814 ; 0.7814 0];
```

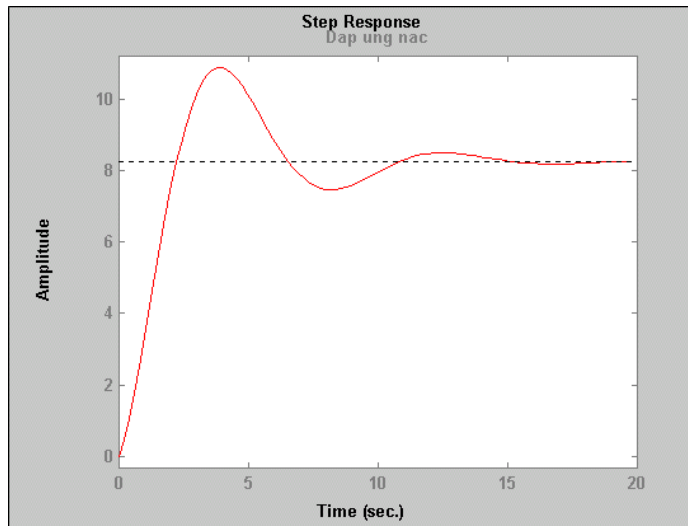
```
b = [1 ; 0];
```

```
c = [1.9691 6.4493];
```

```
d = [0];
```

```
step(a,b,c,d); title('Đáp ứng nấc')
```

và ta được đồ thị đáp ứng nấc của hệ thống như sau:



+ Các bài tập thực hành mức độ cơ bản

Bài tập 1: Thực hành vẽ biểu đồ đáp ứng tần số Nyquist.

Vẽ biểu đồ Nyquist của hệ thống có hàm truyền:

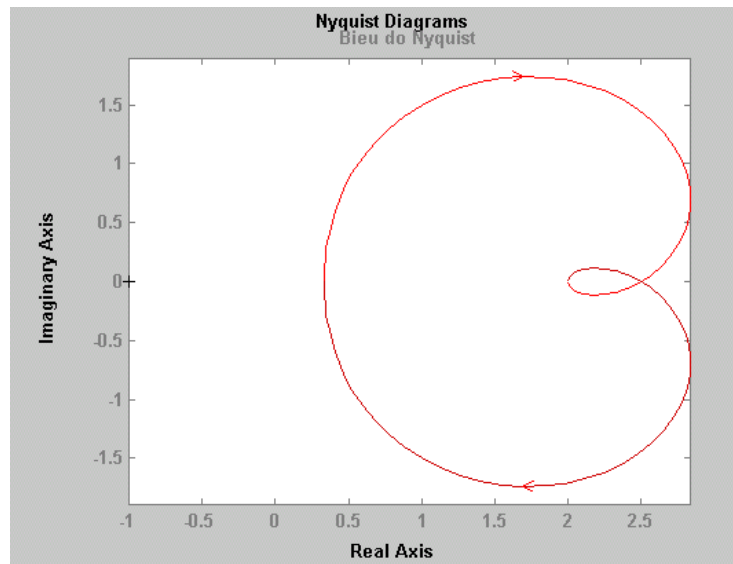
$$H(s) = \frac{2s^2 + 5s + 1}{s^2 + 2s + 3}$$

```
num = [2 5 1];
```

```
den = [1 2 3];
```

```
nyquist(num,den); title('Bieu do Nyquist')
```

và ta được biểu đồ Nyquist như hình vẽ:



Bài tập 2: Thực hành vẽ biểu đồ đáp ứng tần số Nichols.

Lệnh NICHOLS

a) Công dụng:

Vẽ biểu đồ đáp ứng tần số Nichols.

b) Cú pháp:

`[mag,phase,w] = nichols(a,b,c,d)`

`[mag,phase,w] = nichols(a,b,c,d,iu)`

`[mag,phase,w] = nichols(a,b,c,d,iu,w)`

`[mag,phase,w] = nichols(num,den)`

`[mag,phase,w] = nichols(num,den,w)`

c) Ví dụ:

Vẽ đáp ứng Nichols của hệ thống có hàm truyền:

$$H(s) = \frac{-4s^4 + 48s^3 - 18s^2 + 250s + 600}{s^4 + 30s^3 + 282s^2 + 525s + 60}$$

`num = [-4 48 -18 250 600];`

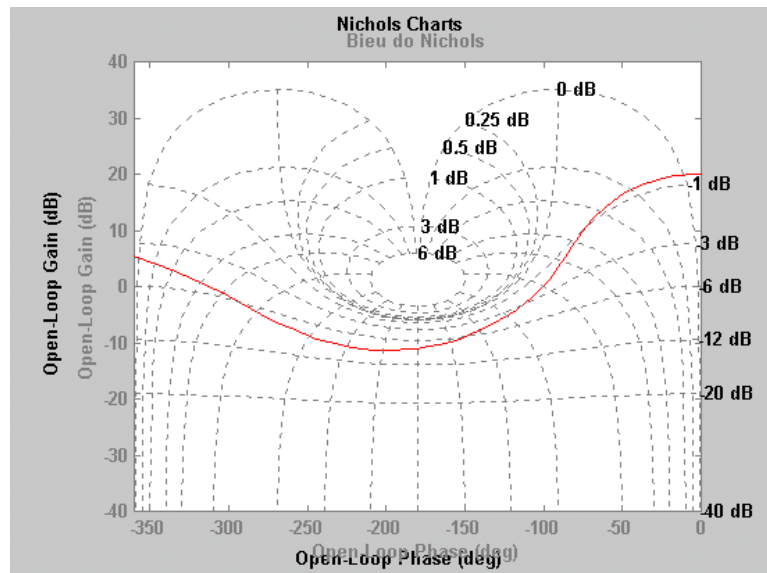
`den = [1 30 282 525 60];`

`nichols(num,den)`

`title('Bieu do Nichols')`

`ngrid('new')`

và ta được biểu đồ Nichols như hình vẽ:



Bài tập 3: Thực hành tính biên dự trữ và pha dự trữ.

Lệnh MARGIN

a) Công dụng:

Tính biên dự trữ và pha dự trữ.

b) Cú pháp:

`[Gm,Pm,Wcp,Wcg] = margin(mag,phase,w)`

`[Gm,Pm,Wcp,Wcg] = margin(num,den)`

`[Gm,Pm,Wcp,Wcg] = margin(a,b,c,d)`

c) Ví dụ:

Tìm biên dự trữ, pha dự trữ và vẽ giản đồ Bode của hệ bậc 2 có $\omega_n = 1$ và $\zeta = 0.2$

`[a,b,c,d] = ord(1,0.2);`

`bode(a,b,c,d)`

`margin(a,b,c,d)`

`[Gm,Pm,Wcp,Wcg] = margin(a,b,c,d)`

và ta được kết quả:

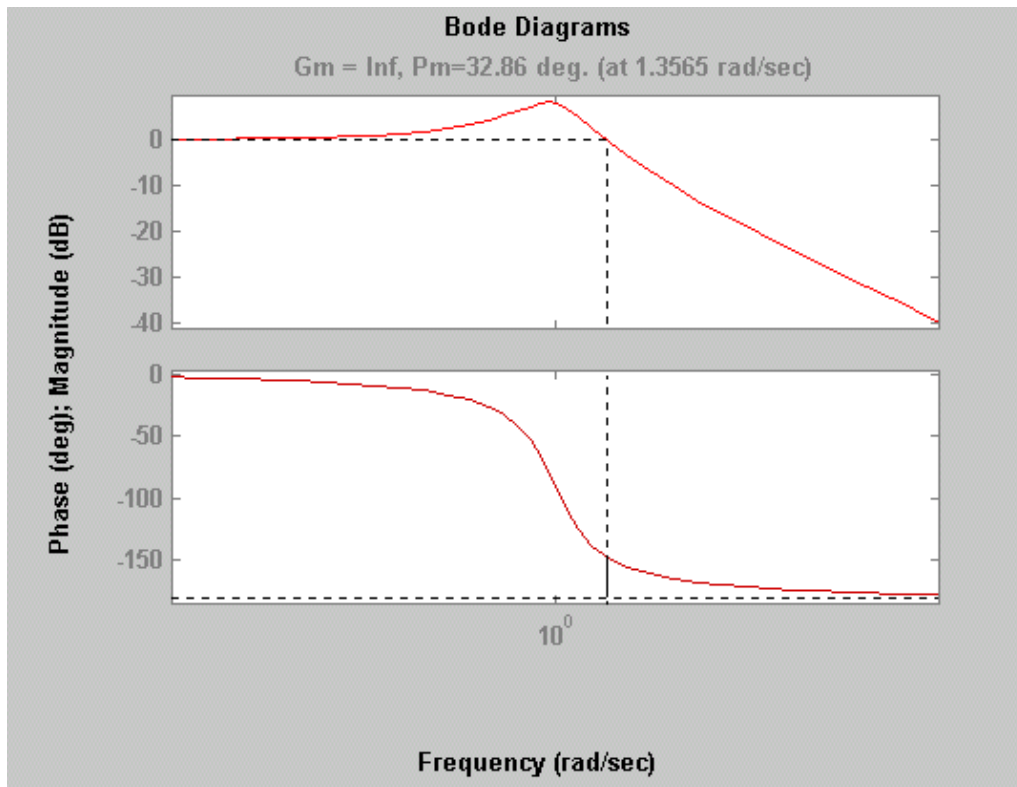
$Gm = \text{Inf}(\infty)$

$Pm = 32.8599$ độ

$Wcg = \text{NaN}$ (không xác định)

$Wcp = 1.3565$

Giản đồ Bode của hệ:



+ Các bài thực hành mức độ nâng cao.

Bài tập 1: Thực hành khảo sát hệ thống theo tiêu chuẩn tần số

- Hệ thống ổn định ở trạng thái hở, sẽ ổn định ở trạng thái kín nếu biểu đồ Nyquist không bao điểm $(-1+i0)$ trên mặt phẳng phức.
- Hệ thống không ổn định ở trạng thái hở, sẽ ổn định ở trạng thái kín nếu biểu đồ Nyquist bao điểm $(-1+i0)$ p lần ngược chiều kim đồng hồ (p là số cực GH nằm ở phải mặt phẳng phức).

» num = [nhập các hệ số của tử số theo chiều giảm dần của số mũ].

» den = [nhập các hệ số của mẫu số theo chiều giảm dần của số mũ].

» nyquist(num,den)

Ví dụ 1:

$$GH(s) = \frac{k}{1-st} \quad (\text{với } k=10, t=1)$$

Ví dụ 2:

$$GH(s) = \frac{k}{s(1-st)} \quad (k=10, t=1)$$

Ví dụ 3:

$$GH(s) = \frac{k}{(t_1s+1)(t_2s+1)} \quad (k=10, t_1=1, t_2=2)$$

Ví dụ 4 :

$$G(s) = \frac{k}{s(t_1s+1)(t_2s+1)} \quad (k=10, t_1=1, t_2=2)$$

Ví dụ 5:

$$G(s) = \frac{k}{s(t_1s+1)(t_2s+1)(t_3s+1)} \quad (t_1=1, t_2=2, t_3=3, k=10)$$

Bài thực hành số 8 (Số tiết: 3 tiết) Mô phỏng các mạch điện, điện tử

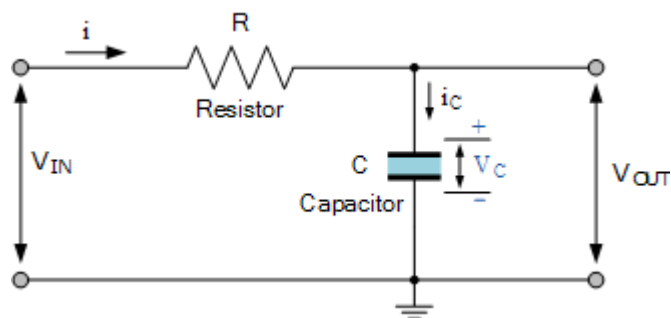
+ **Mục đích của bài thực hành:** Giúp sinh viên có công cụ để tiến hành phỏng các đối tượng, hệ thống nhằm kiểm chứng lại nội dung lý thuyết đã học. Làm cơ sở để nghiên cứu sâu hơn về các đối tượng và hệ thống.

+ **Yêu cầu cần đạt được của bài thực hành:** Sinh viên thực hiện thành thạo phần mềm Matlab/Simulink để tiến hành mô phỏng cho các đối tượng trong hệ thống điện, điện tử

+ Bài giải mẫu

Mạch tích phân RC

Mạch tích phân RC là một mạch RC được kết nối nối tiếp tạo ra tín hiệu đầu ra tương ứng với quá trình tích phân toán học. Mạch RC thụ động không gì khác hơn là một điện trở mắc nối tiếp với tụ điện, đó là một điện trở cố định mắc nối tiếp với tụ điện có điện trở phụ thuộc tần số giảm khi tần số trên các bản của nó tăng lên. Do đó, ở tần số thấp, dung kháng, X_c của tụ điện cao trong khi ở tần số cao, **dung kháng của tụ điện** thấp với công thức dung kháng tiêu chuẩn của $X_c = 1 / (2\pi fC)$.



Hình 8.1: Mạch tích phân RC

Phương trình cân bằng là:

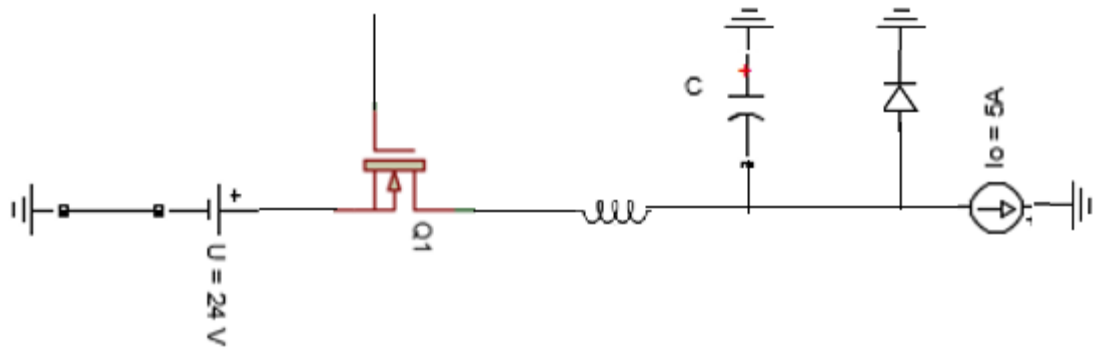
$$e_i - iR - \frac{1}{C} \int i dt = 0$$

từ đây ta có thể rút ra

$$i = \frac{1}{R} * (e_i - \frac{1}{C} \int i dt)$$

Mạch đóng mở sử dụng MOSFET

Mạch đóng mở sử dụng MOSFET là một mạch điện khá thông dụng trong mạch điện điện tử. Mô hình mạch đóng mở như sau



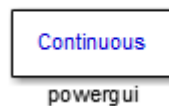
Hình 8.2: Mạch đóng ngắt dùng MOSFET

MOSFET thường được sử dụng để làm khóa đóng mở vì kênh dẫn của E-MOSFET hình thành hay không liên quan đến V_{GS} .

Khi $V_{GS} < V_{GS(th)}$ kênh dẫn giữa D và S chưa hình thành => Khóa hở

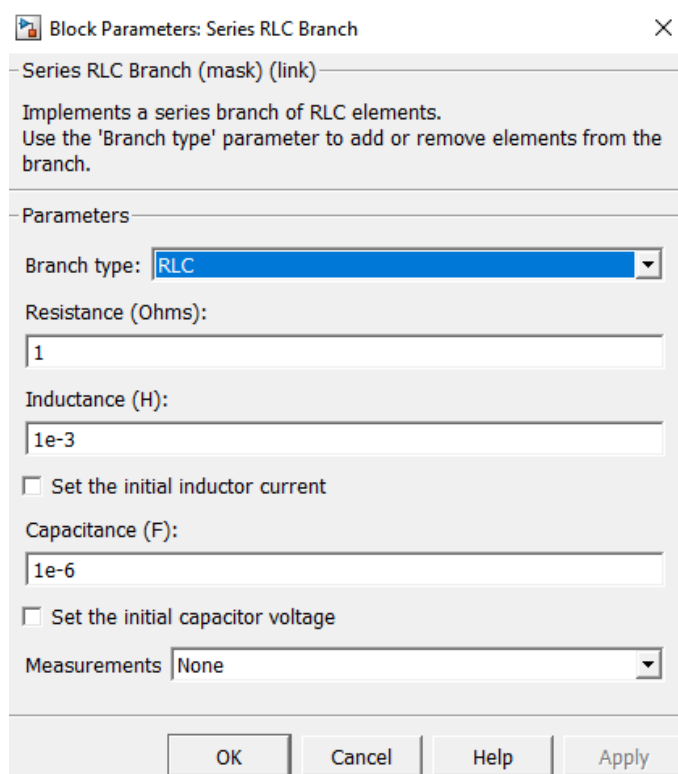
Khi $V_{GS} > V_{GS(off)}$ kênh dẫn hình thành giữa D và S => Khóa đóng

Trước tiên ta vào *Simscape* → *SimPowerSystem* lấy khối *Powergui* để có thể kết nối nguồn với các thiết bị.



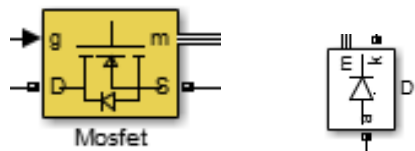
vào *Simscape* → *SimPowerSystem* → *Elements* để lấy các khối *Serials RLC Branch*

Trong đó ta có thể thay đổi loại tải bằng cách lựa chọn *Branch type* và thay đổi thông số của R, L, C tương ứng.



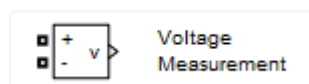
Hình 8.10: Lựa chọn loại thiết bị R, L, C tương ứng

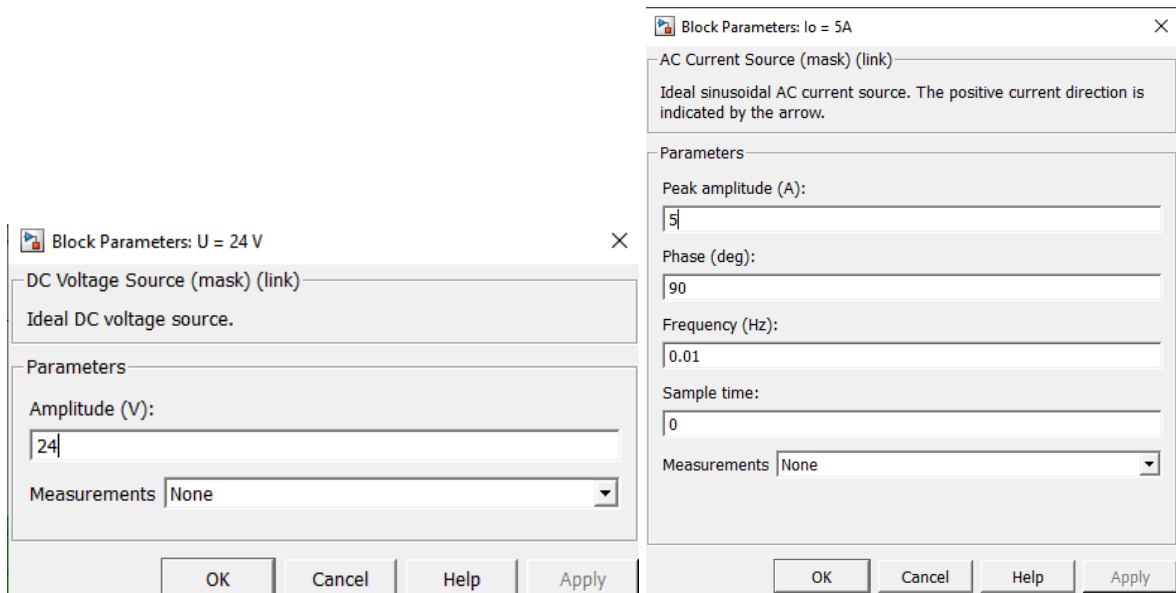
vào *Simscape* → *SimPowerSystem* → *PowerElectronics* để lấy khối Mosfet và Diot tương ứng.



vào *Simscape* → *ApplicantLibraries* → *ElectronicalSources* để lấy khối nguồn dòng *AC Current Source* và *DC Voltage Source*.

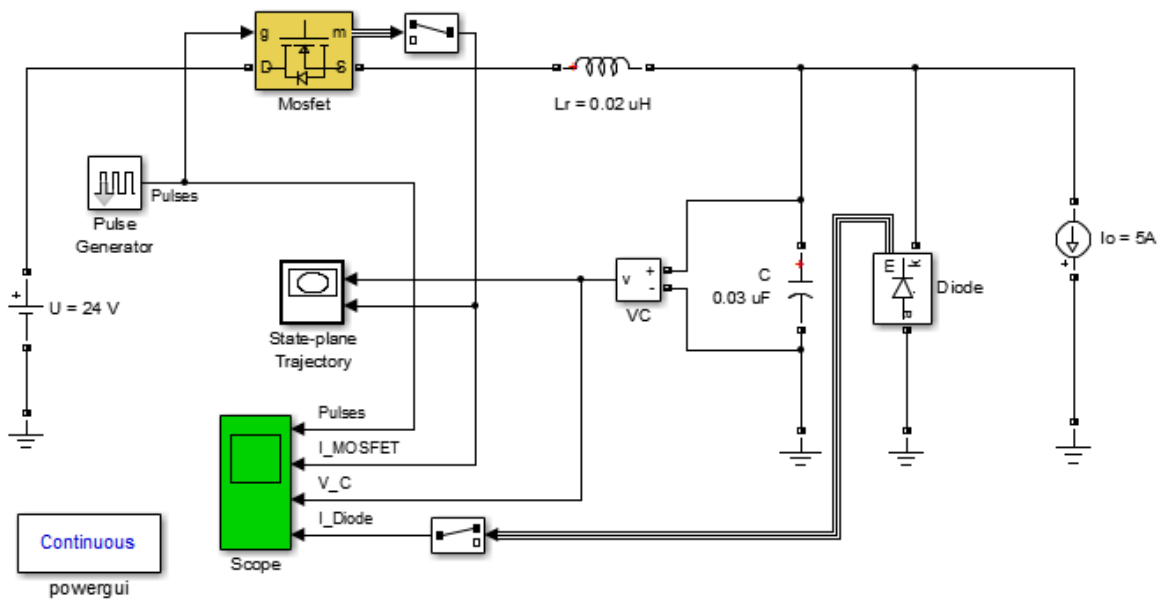
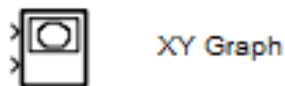
vào *Simscape* → *SimPowerSystem* → *Machines* → *Measurements*. Ở đây ta lấy khối đo áp để đo điện áp của tụ điện.





Hình 8.11: Cài đặt thông số điện áp và dòng điện.

Lấy khối vẽ đồ thị xy bằng cách vào Simulink → Sink → XY Graph



Hình 8.12: Mô phỏng trên Matlab simulink mạch đóng mở sử dụng MOSFET

+ Các bài thực hành mức độ cơ bản.

Bài 1: Mô phỏng mạch tích phân RC với các thông số $R=10\text{kohm}$, $C=10^{-6}\text{F}$

+ Các bài thực hành mức độ nâng cao.

Bài 2: Mô phỏng mạch đóng ngắt dùng Mosfet.

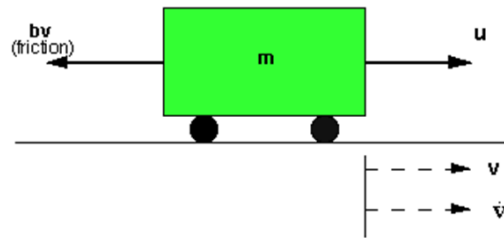
Bài thực hành số 9 (Số tiết: 3 tiết) Mô phỏng hệ điện cơ

+ **Mục đích của bài thực hành:** Giúp sinh viên có công cụ để tiến hành phỏng các đối tượng, hệ thống nhằm kiểm chứng lại nội dung lý thuyết đã học. Làm cơ sở để nghiên cứu sâu hơn về các đối tượng và hệ thống.

+ **Yêu cầu cần đạt được của bài thực hành:** Sinh viên mô tả toán học và xây dựng mô hình mô phỏng và sử dụng Simulink để mô phỏng hệ thống

+ Bài giải mẫu

Bài tập 1: Xây dựng mô hình mô phỏng hệ thống xe tải bằng Matlab Simulink



cho bởi phương trình:

$$m \frac{dv}{dt} = u - bv$$

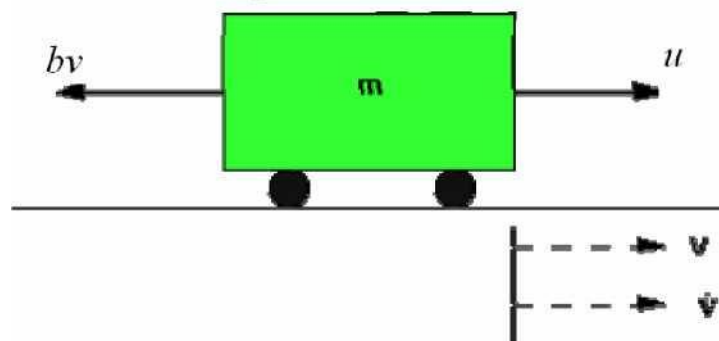
$u = 500 \text{ N}$ lực phát động của động cơ (ngõ vào của mô hình)

$m = 1000 \text{ kg}$ khối lượng xe

$b = 50 \text{ N sec/m}$ hệ số ma sát

v vận tốc (ngõ ra của mô hình)

Bước 1: Xây dựng mô hình hệ thống xe tải:



Mô hình xe tải cho bởi phương trình:

$$m \frac{dv}{dt} = u - bv \text{ hay } \frac{dv}{dt} = \frac{1}{m}(u - bv)$$

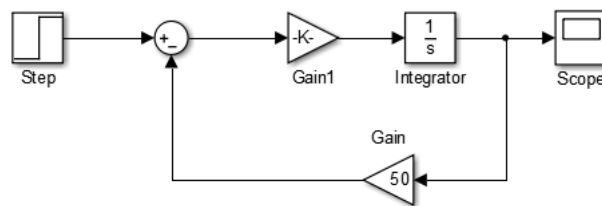
Trong đó m là khối lượng xe, u là lực tác động của động cơ (ngõ vào của mô

hình), b là hệ số ma sát và v là vận tốc đạt được (ngõ ra của mô hình).

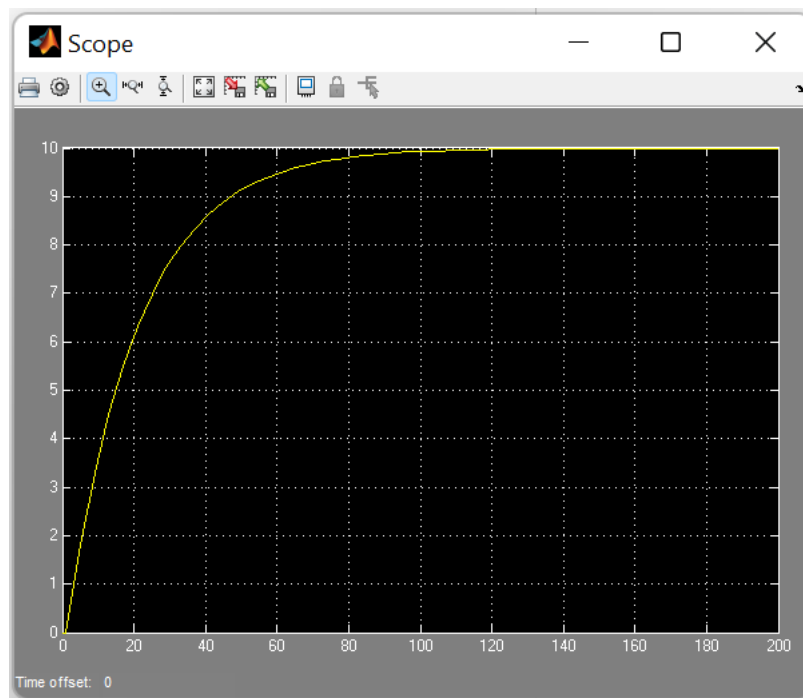
Xuất phát của việc xây dựng các mô hình hệ thống từ các phương trình vi phân tương ứng là các khối tích phân (Integrator). Nếu trong phương trình mô tả hệ thống có vi phân bậc n thì ta sẽ đặt vào mô hình n khối tích phân, do quan hệ $\int \frac{dv}{dt} = v$

Bước 2: Xây dựng mô hình mô phỏng trên Simulink

Mô hình mô phỏng với với tham số đã cho



Kết quả mô phỏng:



+ Các bài thực hành mức độ cơ bản.

Bài 1: Cho đối tượng có mô hình hàm truyền như sau :

$$W(s) = \frac{10s + 3}{s^2 + 3s + 1}$$

Xây dựng mô hình mô phỏng hệ thống điều khiển đối tượng trên sử dụng bộ điều khiển PID trong simulink với tín hiệu vào có dạng Step

Bài 2: Cho đối tượng có mô hình hàm truyền như sau :

$$W(s) = \frac{10s + 3}{s^2 + 3s + 1}$$

Xây dựng mô hình mô phỏng hệ thống điều khiển đối tượng trên sử dụng bộ điều khiển PID 2 bậc trong simulink với tín hiệu vào có dạng Step

+ Các bài thực hành mức độ nâng cao.

Bài 1: Cho lò nhiệt điện trở có hàm truyền đạt như sau:

$$W(s) = \frac{K.e^{-T_2s}}{T_1s + 1}$$

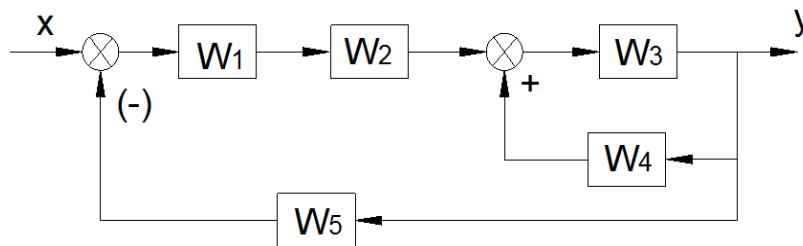
Với $K=150, T_1=3000, T_2=50$. Sử dụng bộ điều khiển PID trong Matlab/Simulink để điều khiển đối tượng trên.

Bài 2: Cho hệ thống có sơ đồ khối như sau:

$W1 = s/(s-1)$; $W2 = 1/(s-1)$;

$$W3 = \frac{10s + 3}{s^2 + 3s + 1}$$

$W4 = 1/(s+2)$, $W5 = 4$



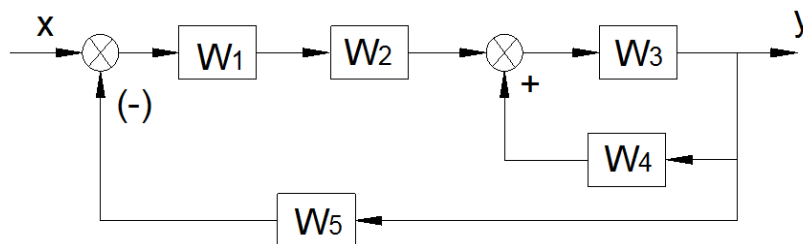
Xác định đáp ứng của hệ thống trên với tín hiệu vào có dạng Constant.

Bài 3: Cho hệ thống có sơ đồ khối như sau:

$W1 = s/(s-1)$; $W2 = 1/(s-1)$;

$$W3 = \frac{10s + 3}{s^2 + 3s + 1}$$

$W4 = 1/(s+2)$, $W5 = 4$



Sử dụng bộ điều khiển PID trong Matlab/Simulink để điều khiển hệ thống trên.

Bài 4 : Cho lò nhiệt điện trở có hàm truyền đạt như sau:

$$W(s) = \frac{K.e^{-T_2s}}{T_1s + 1}$$

Với $K=150, T_1=3000, T_2=50$. Sử dụng bộ điều khiển PID 2 bậc trong Matlab/Simulink để điều khiển đối tượng trên.

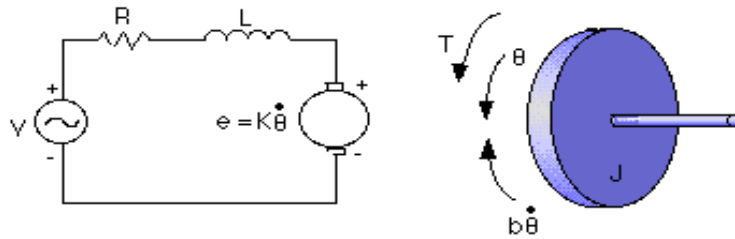
Bài thực hành số 10 (Số tiết: 3 tiết) Mô phỏng hệ điện cơ (Tiếp)

+ **Mục đích của bài thực hành:** Giúp sinh viên có công cụ để tiến hành phỏng các đối tượng, hệ thống nhằm kiểm chứng lại nội dung lý thuyết đã học. Làm cơ sở để nghiên cứu sâu hơn về các đối tượng và hệ thống.

+ **Yêu cầu cần đạt được của bài thực hành:** Sinh viên mô tả toán học và xây dựng mô hình mô phỏng và sử dụng Simulink để mô phỏng hệ thống

+ Bài giải mẫu

Bài 1: Cho mạch điện phần ứng và sơ đồ tách của rotor động cơ DC như trong hình sau:



Giả thiết các thông số vật tư của hệ thống như sau:

- Moment quán tính của rotor (J) = $0.01 \text{ kg.m}^2/\text{s}^2$.
- Hệ số giảm chấn của hệ thống cơ khí (b) = 0.1 Nms .
- Hằng số điện cơ ($K=K_e=K_t$) = 0.01 Nm/Amp .
- Điện trở (R) = 10 ohm .
- Điện cảm (L) = 0.5 H .
- Input (V): điện áp nguồn.
- Output (θ): vị trí của trục.
- Giả thiết rotor và trục cứng vững tuyệt đối.

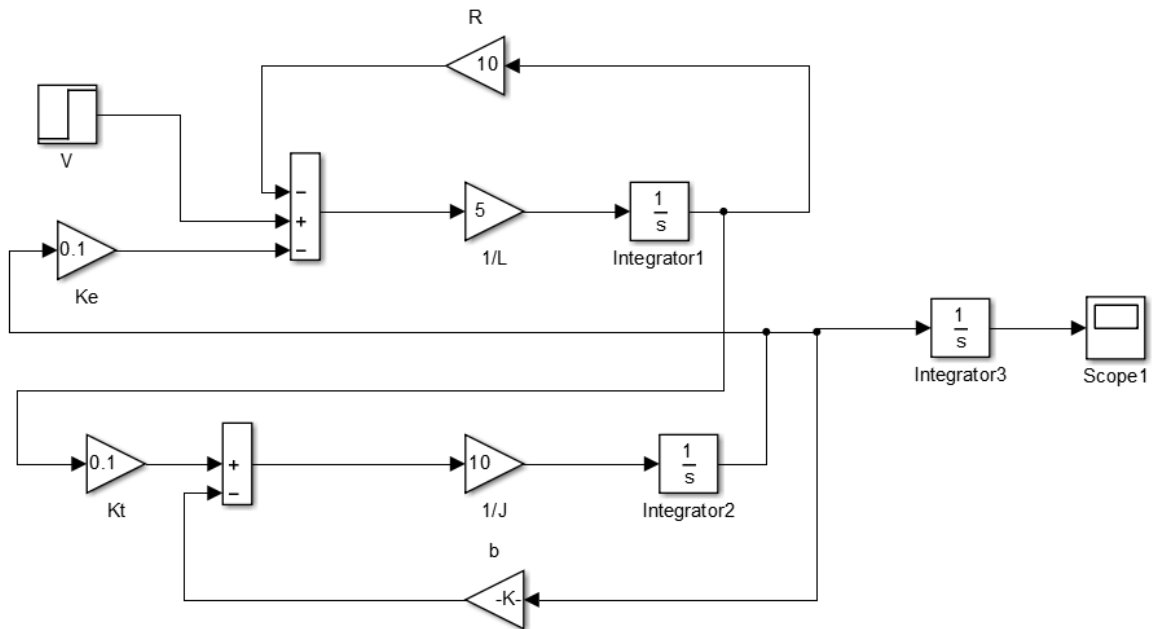
Sử dụng Matlab Simulink xây dựng mô hình điều khiển tốc độ động cơ điện một chiều với yêu cầu:

Ta có hệ thống điều khiển vị trí motor DC cho bởi phương trình vi phân sau:

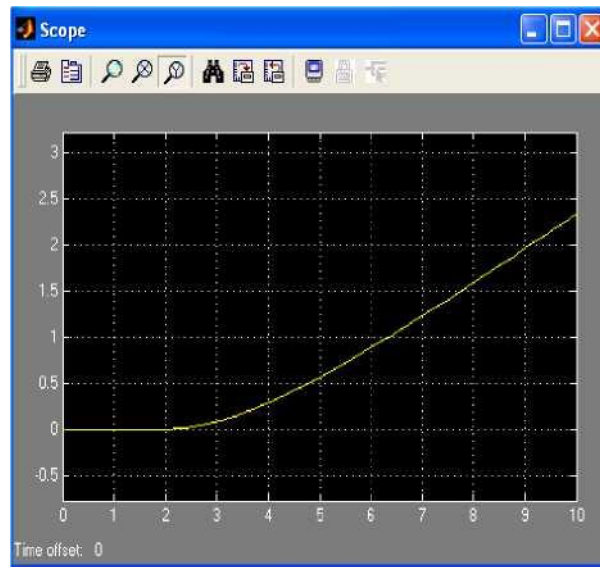
$$\frac{d^2\theta}{dt^2} = \frac{1}{J} (K_t i - b \frac{d\theta}{dt})$$

$$\frac{di}{dt} = \frac{1}{L} (-Ri + V - K_e \frac{d\theta}{dt})$$

Từ các phương trình trên ta xây dựng mô hình hệ thống điều khiển vị trí motor DC trên Simulink như sau:



Với các tham số đã cho ta có kết quả mô phỏng:



+ Các bài thực hành mức độ cơ bản.

Bài 1: Cho mô hình không gian trạng thái có

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 5 & 2 & 1 \\ 4 & 5 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 3 \\ 1 & 2 \\ 3 & 5 \end{bmatrix}$$

$$C = [0 \quad 1 \quad 0] \quad D = [0 \quad 0]$$

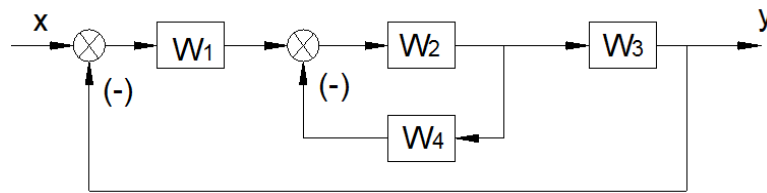
Chuyển mô hình không gian trạng thái này sang mô hình hàm truyền đạt

Bài 2: Cho mô hình hàm truyền đạt, có hàm truyền

$$H(s) = \frac{s^2 + 3s - 1}{s^3 - 5s^2 + s + 3}$$

Chuyển mô hình hàm truyền đạt này sang mô hình không gian trạng thái

Bài 3: Cho hệ thống có sơ đồ khối như sau



$$W1 = 0,5 \quad ; \quad W2 = 1/(s+2)(s+8);$$

$$W3 = 1/s \quad ; \quad W4 = 2$$

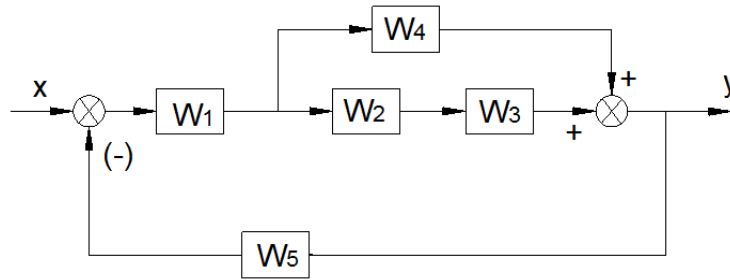
Hãy xây dựng hàm truyền hệ kín. Vẽ đồ thị đáp ứng của hệ thống với tín hiệu vào dạng bước nhảy

Chuyển mô hình hàm truyền này sang mô hình không gian trạng thái.

Bài 4: Cho hệ thống có sơ đồ khối như sau

$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$

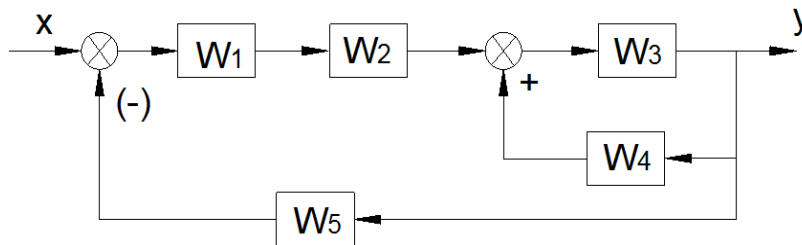


Hãy xây dựng hàm truyền hệ kín. Vẽ đồ thị đáp ứng của hệ thống

Bài 5: Cho hệ thống có sơ đồ khối như sau

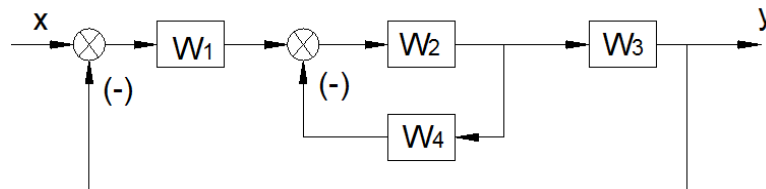
$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$



Hãy xây dựng hàm truyền hệ kín. Vẽ đồ thị đáp ứng của hệ thống

Bài 6: Cho hệ thống có sơ đồ khối như sau



$$W1 = 0,5 \quad ; \quad W2 = 1/(s+2)(s+8);$$

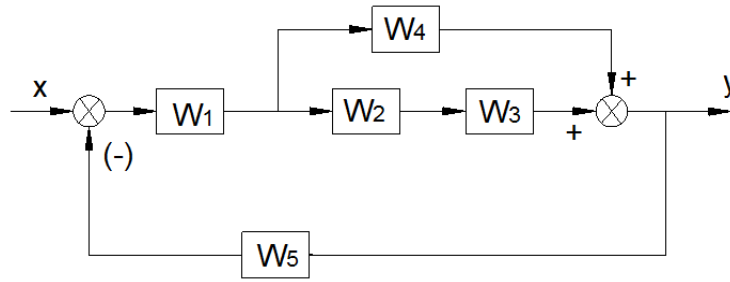
$$W3 = 1/s \quad ; \quad W4 = 2$$

Mô phỏng hệ thống trên bằng Matlab Simulink với tín hiệu vào là khối step

Bài 7: Cho hệ thống có sơ đồ khối như sau

$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$

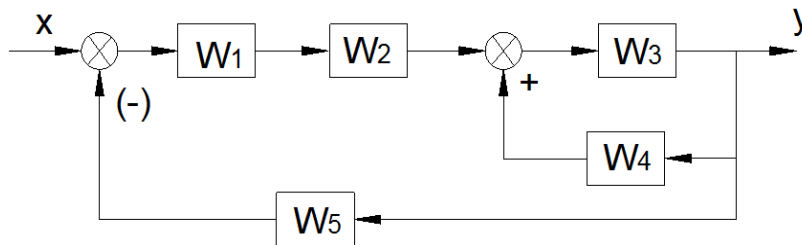


Mô phỏng hệ thống trên bằng Matlab Simulink với tín hiệu vào là khối step

Bài 8: Cho hệ thống có sơ đồ khối như sau

$$W1 = 5s/(s-1) \quad ; \quad W2 = 1/(s-1);$$

$$W3 = 5 \quad ; \quad W4 = 1/(s+2); \quad W5 = 4$$



Mô phỏng hệ thống trên bằng Matlab Simulink với tín hiệu vào là khối step

+ Các bài thực hành mức độ nâng cao.

Bài 1: Xây dựng mô hình Khởi động động cơ một chiều qua 3 cấp điện trở tương ứng 3.5Ω , 2.5Ω , 1.5Ω . với tham số động cơ 50HP, 500V, 1750RPM

Bài 2: Xây dựng mô hình Khởi động động cơ một chiều qua 2 cấp điện trở tương ứng 3.2Ω , 2.0Ω , 1.5Ω . với tham số động cơ 100HP, 500V, 1750RPM