

# 第3回

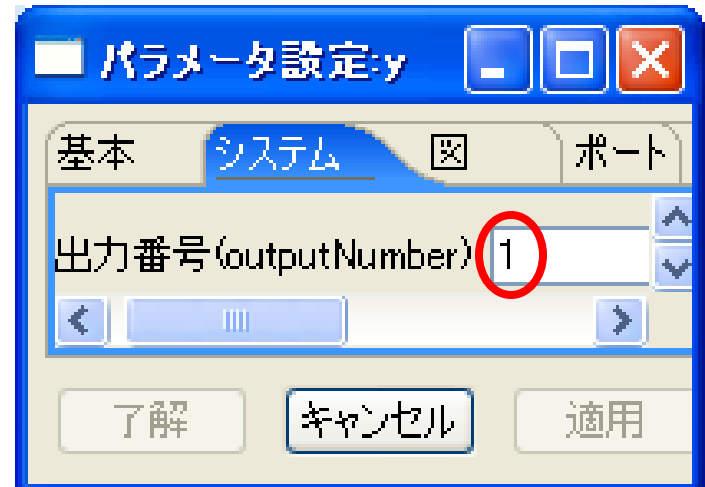
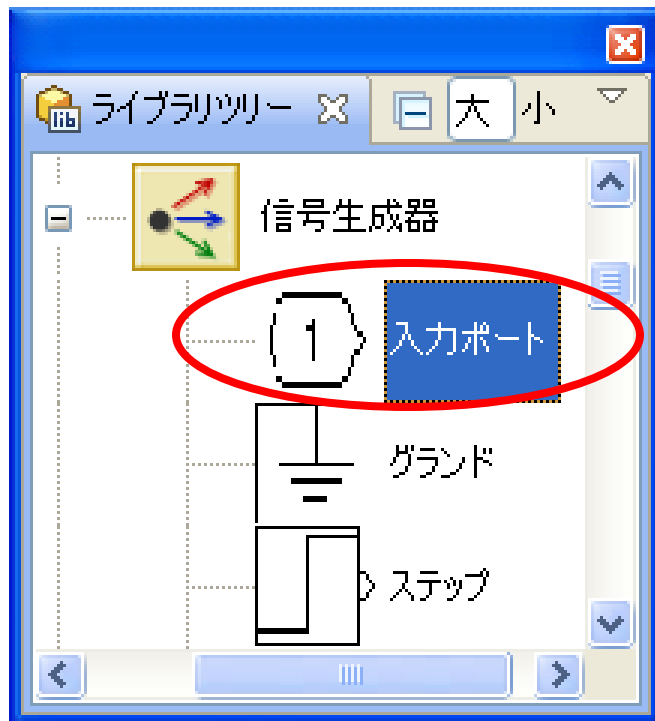
- 伝達関数の導出
- 結合と等価変換によるブロック線図の簡単化
- サブシステム化によるブロック線図の簡単化

# 入出力ポートを用いた伝達関数の導出

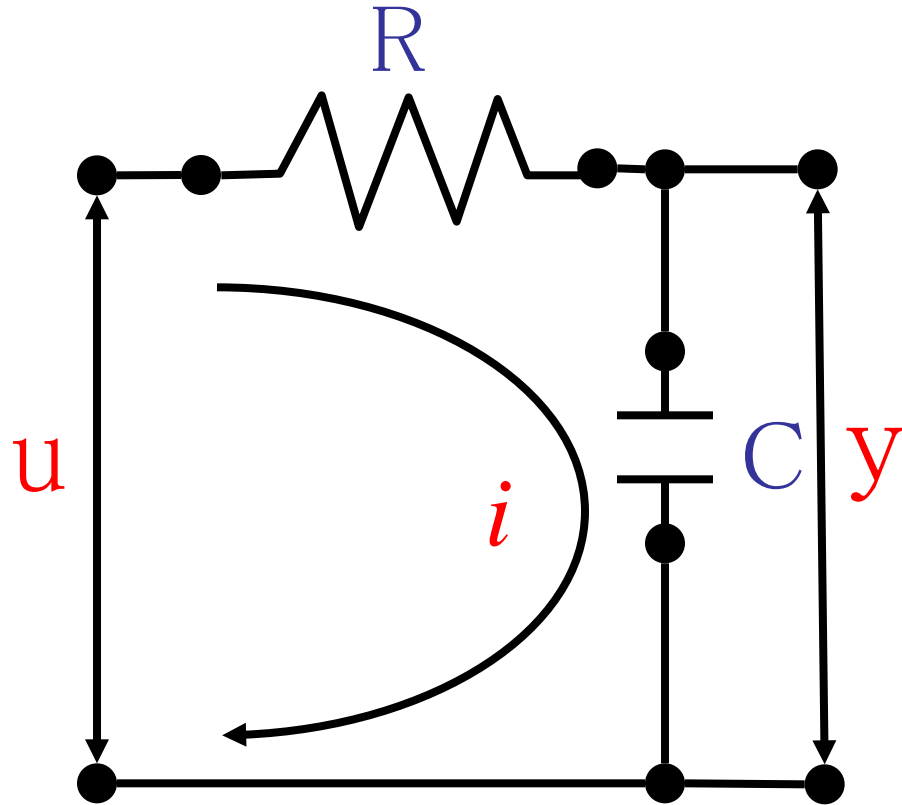
1. システムに入出力ポートを接続する。
2. 「線形解析」→「伝達関数」メニューを選択する。
3. コンソールに伝達関数が表示される。

複数の入出力ポートが存在するとき、  
全ての入力ポートから出力ポートまでの  
伝達関数(伝達関数行列)が表示される。

# システムの入出力ポート



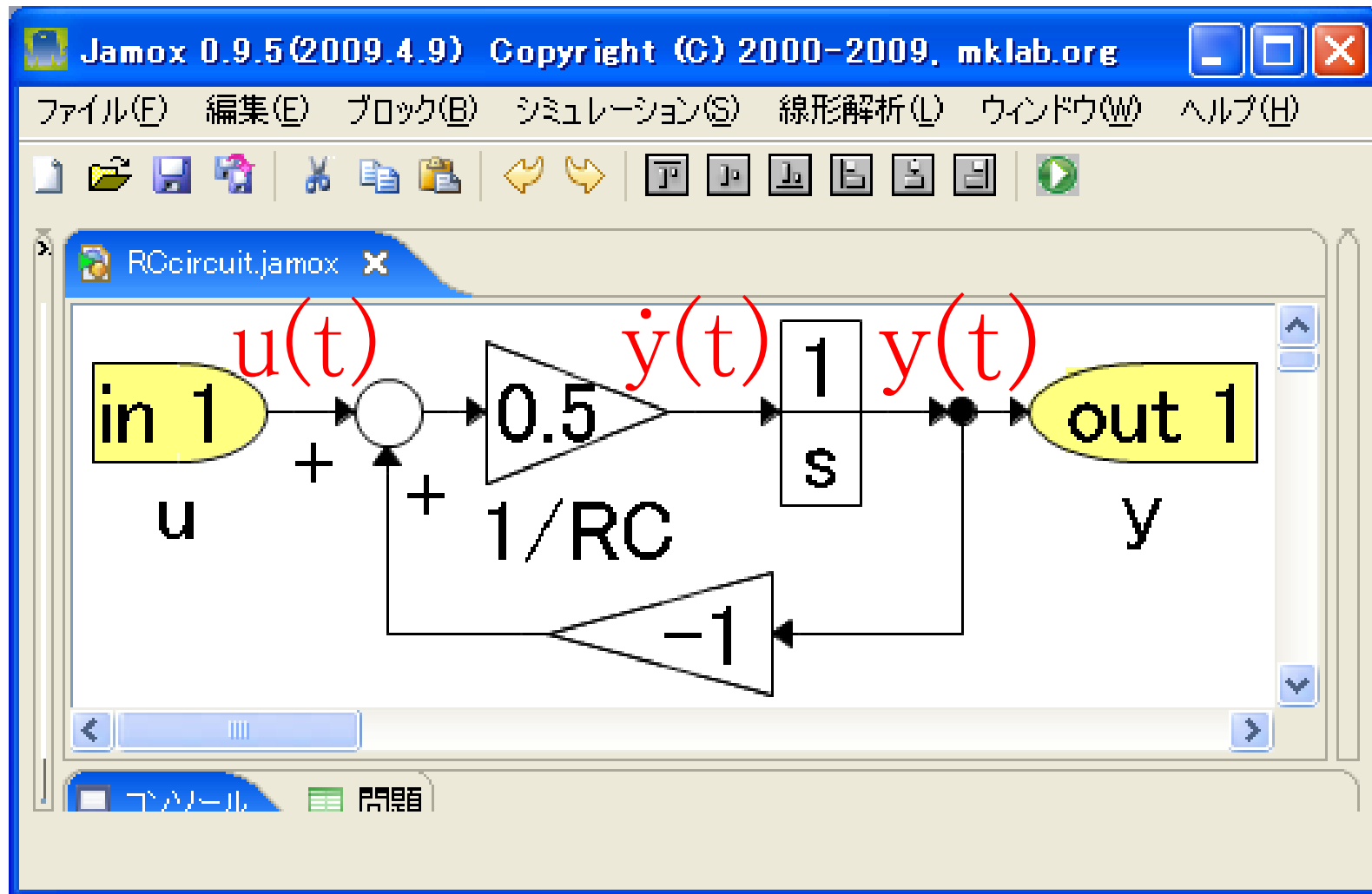
# RC回路



$$\dot{y}(t) = (-y(t) + u(t)) / RC$$

# RC回路のブロック線図

$$\dot{y}(t) = (-y(t) + u(t)) / RC, RC = 2$$



# RC回路の伝達関数

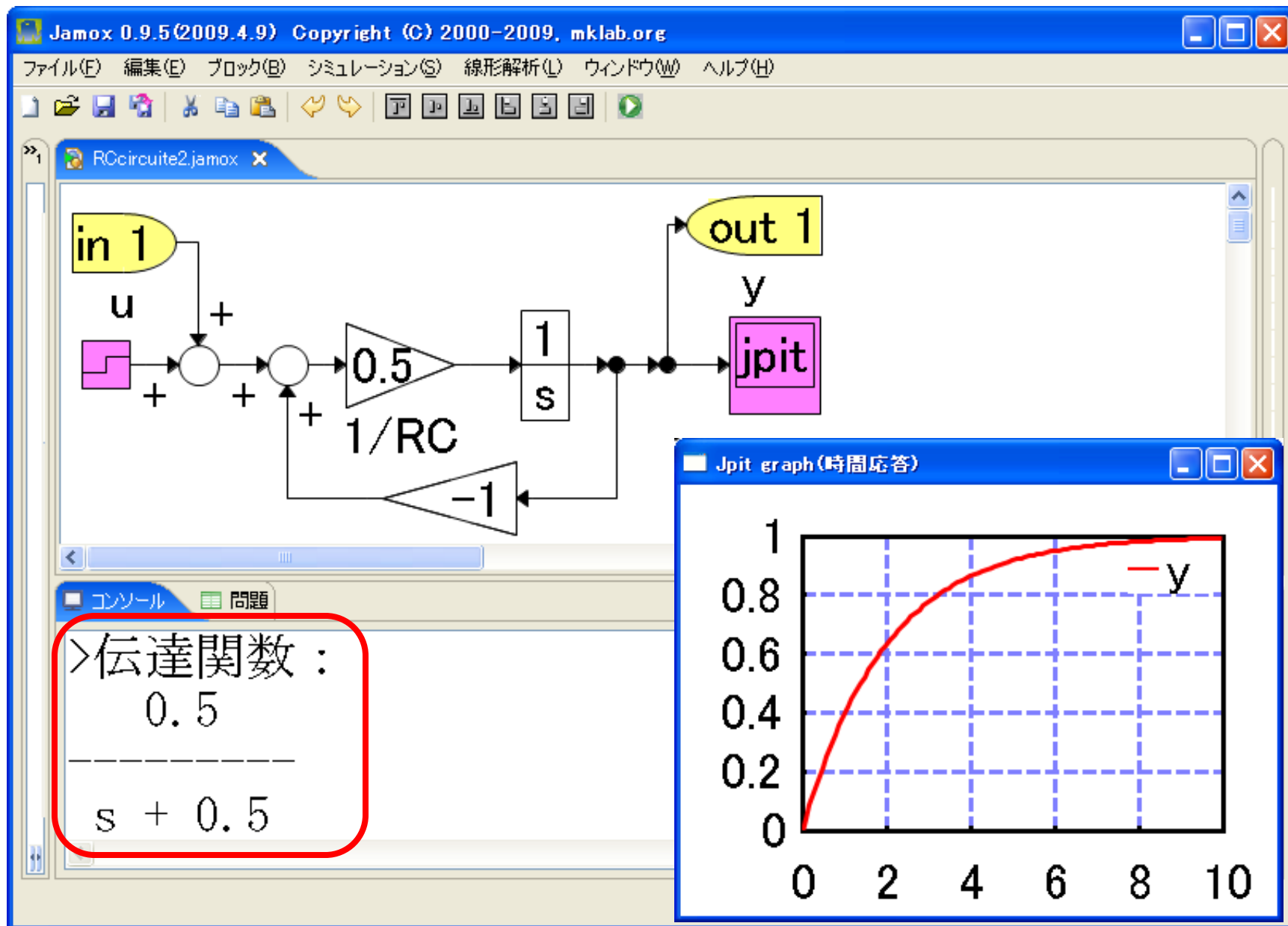
The screenshot shows the Jamox software interface. The main window displays a block diagram of an RC circuit. An input block labeled 'in 1' with the signal 'u' is connected to a summing junction. The output of the summing junction goes to a gain block labeled '0.5'. The output of this block is fed back through a gain block labeled '-1' to the summing junction. A feedback path is also shown with a gain block labeled '1/RC'. A menu is open over the diagram, listing various analysis options such as '伝達関数(T)', '状態空間表現(数値)(S)', etc.

コンソール

>伝達関数:  
0.5  
-----  
s + 0.5

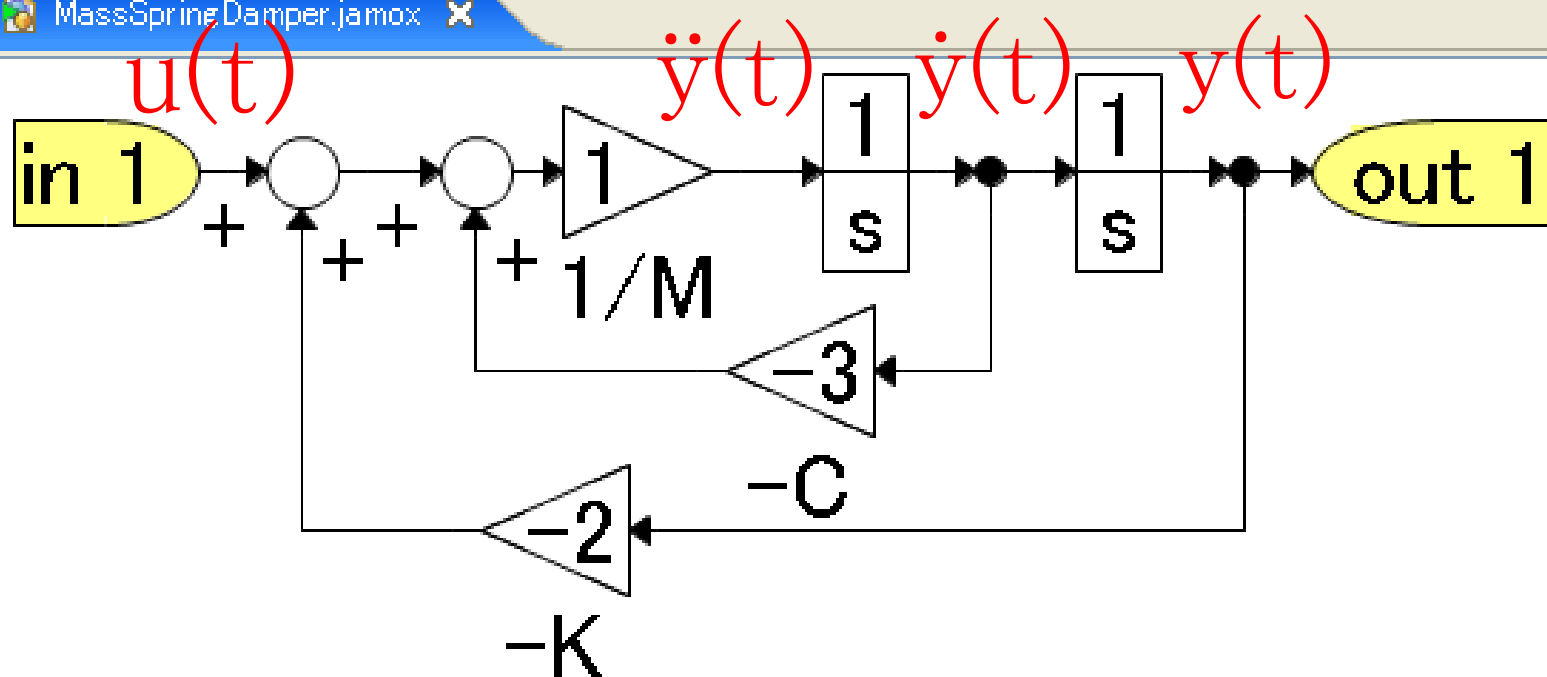
$$G(s) = \frac{1}{RCs + 1}, RC = 2$$

# RC回路の伝達関数と応答



# 質量・ばね・ダンパ系のブロック線図

$$\ddot{y}(t) = \left( -C\dot{y}(t) - Ky(t) + u(t) \right) / M$$



$$M = 1, K = 2, C = 3$$

# 質量・ばね・ダンパ系の伝達関数

The screenshot shows the Jamox software interface with a block diagram of a mass-spring-damper system. The input is labeled 'in 1' and the output is 'out 1'. The system consists of two summing junctions, two integrators (represented by '1/s' blocks), and two gain blocks labeled '-2' and '-3'. The gain '-2' is associated with the parameter '-C' and the gain '-3' is associated with '-3'. A menu is open over the diagram, listing various analysis options.

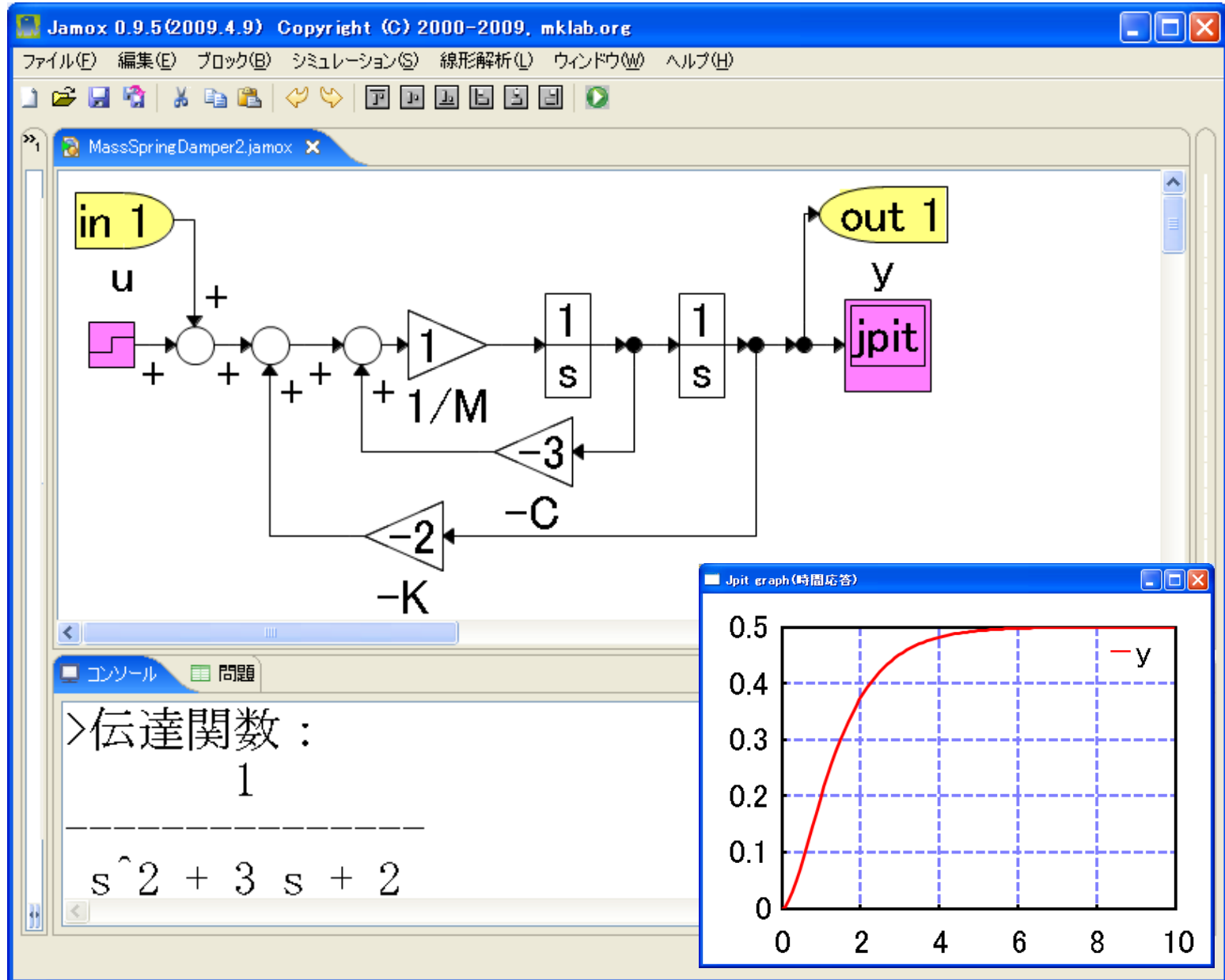
コンソール

>伝達関数：  
1  
-----  
s<sup>2</sup> + 3s + 2

$M = 1, K = 2, C = 3$

$$G(s) = \frac{1}{Ms^2 + Cs + K}$$

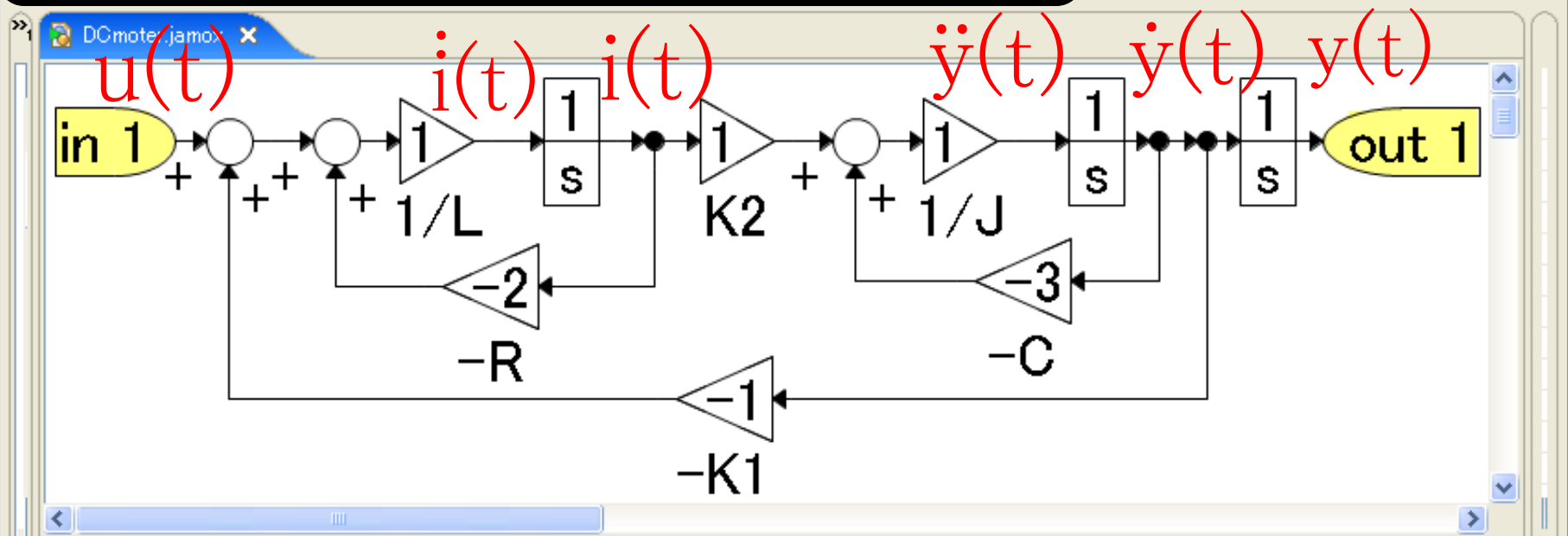
# 質量・ばね・ダンパ系の伝達関数と応答



# DCサーボモータのブロック線図

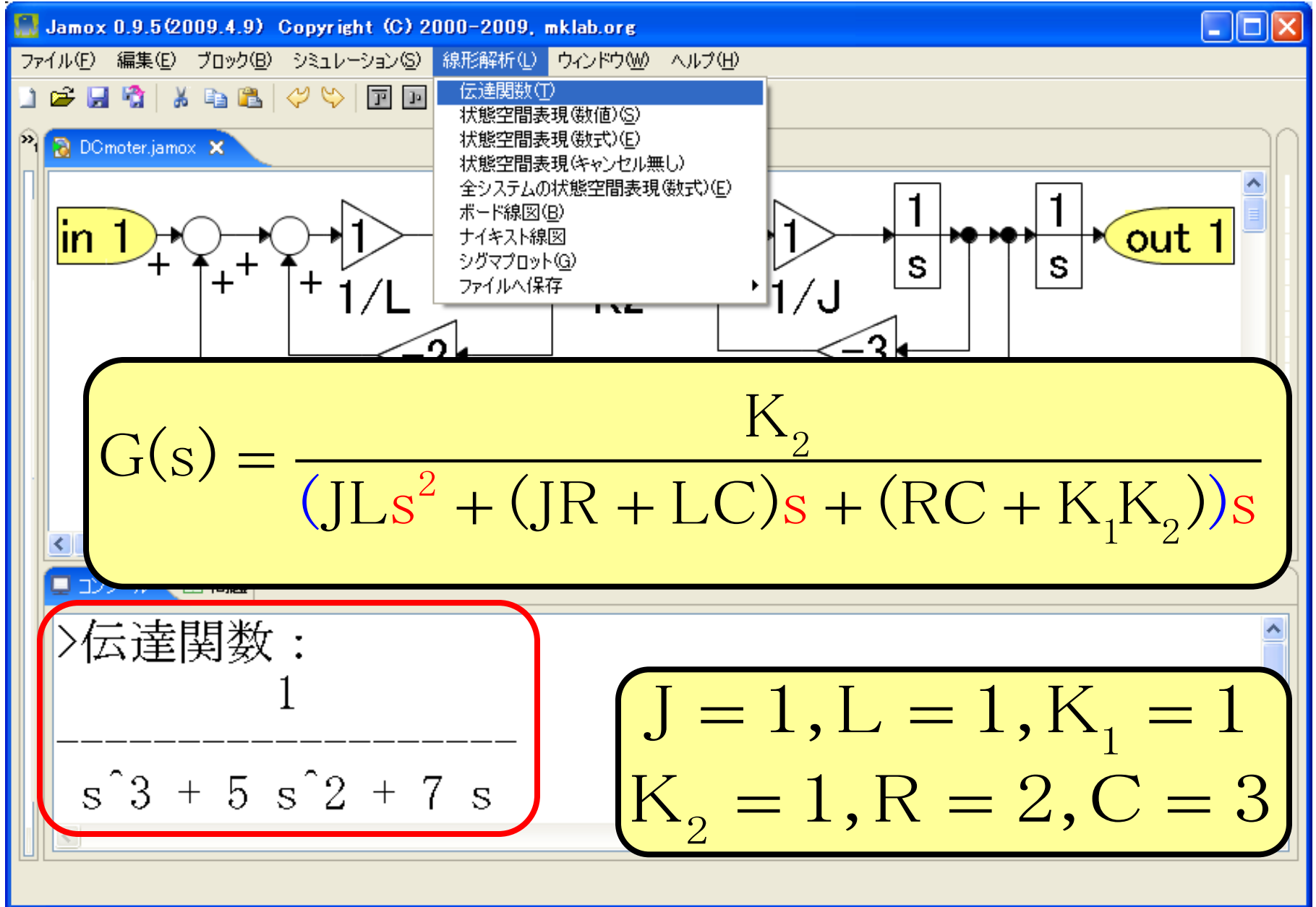
$$\ddot{y}(t) = (K_2 i(t) - C \dot{y}(t)) / J$$

$$i(t) = (-Ri(t) - K_1 \dot{y}(t) + u(t)) / L$$



$$J = 1, L = 1, K_1 = 1, K_2 = 1, R = 2, C = 3$$

# DCサーボモータの伝達関数



Jamox 0.9.5 (2009.4.9) Copyright (C) 2000-2009. mklab.org

ファイル(F) 編集(E) ブロック(B) シミュレーション(S) 線形解析(L) ウィンドウ(W) ヘルプ(H)

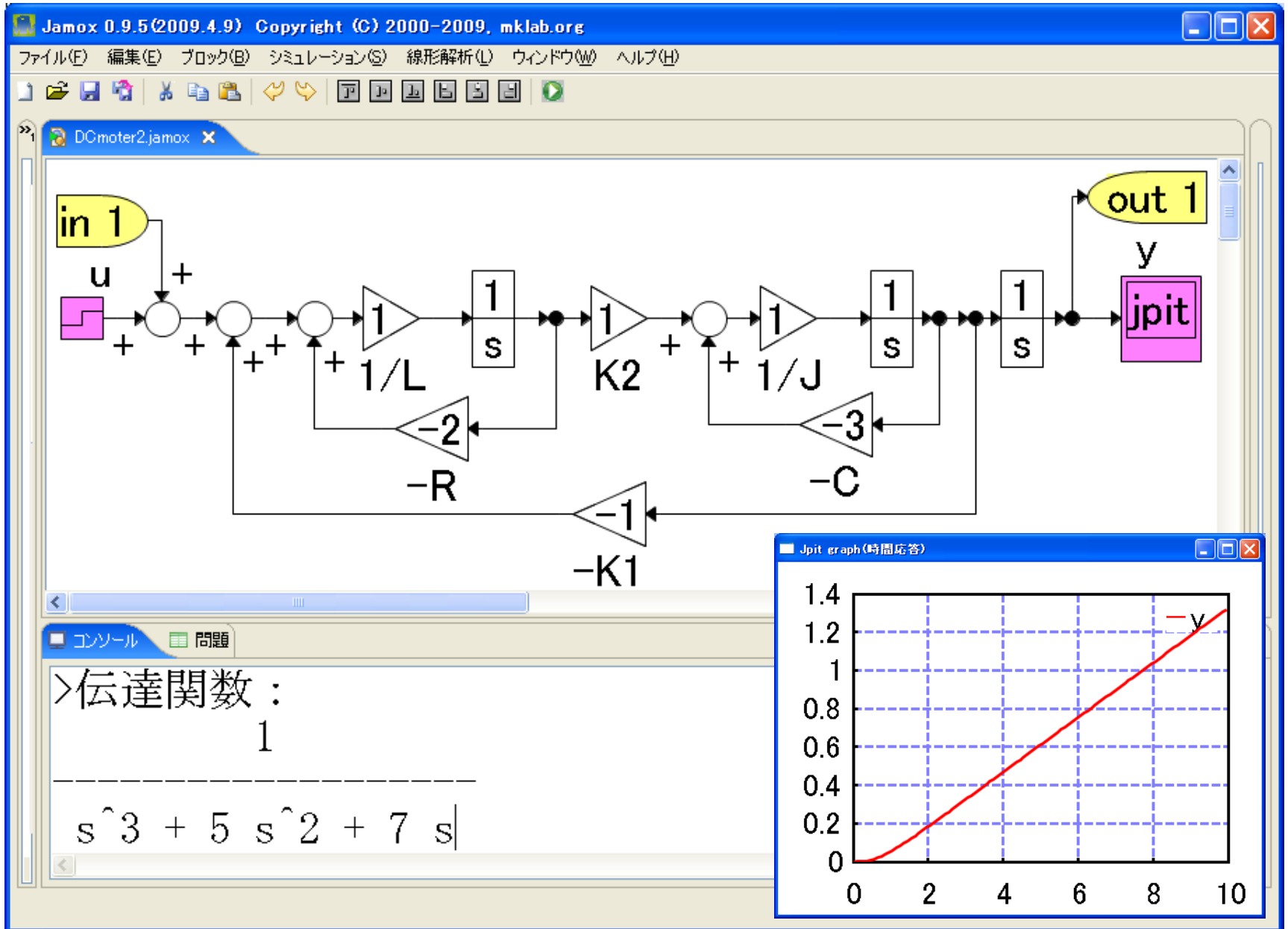
伝達関数(T)  
状態空間表現(数値)(S)  
状態空間表現(数式)(E)  
状態空間表現(キャンセル無し)  
全システムの状態空間表現(数式)(E)  
ボード線図(B)  
ナイキスト線図  
シグマプロット(G)  
ファイルへ保存

$$G(s) = \frac{K_2}{(JLs^2 + (JR + LC)s + (RC + K_1K_2))s}$$

>伝達関数：  
1  
-----  
s^3 + 5 s^2 + 7 s

$$J = 1, L = 1, K_1 = 1$$
$$K_2 = 1, R = 2, C = 3$$

# DCサーボモータの伝達関数と応答



# 結合によるブロック線図の単純化

- 直列結合

結果: 元の伝達関数(G1, G2)の積

$$G = G1 * G2$$

- 並列結合

結果: 元の伝達関数(G1, G2)の和

$$G = G1 + G2$$

- フィードバック結合(ネガティブ)

結果: 元の伝達関数(G1, G2)の式

$$G = G1 / (1 + G1 * G2)$$

# 直列結合による簡単化

The screenshot shows the Jamox software interface. The top part displays a block diagram where two blocks,  $G_1 = \frac{1}{s+1}$  and  $G_2 = \frac{1}{s+2}$ , are connected in series. An orange arrow points to a simplified block  $G_2 * G_1 = \frac{1}{s^2 + 3s + 2}$ . Below this, the console window shows the transfer function matrix calculation:

```
>伝達関数行列:  
=== ( 2 x 2) RaMatrix ===  
[ ( 1) ] [ ( 2) ]  
( 1) [ 1 / (s^2 + 3*s + 2) ]  
[ 0 / 1 ]  
2) [ 0 / 1 ]  
[ 1 / (s^2 + 3*s + 2) ]
```

Three callout boxes highlight specific parts of the diagram and matrix:

- In1 → Out1 伝達関数**: Points to the first block  $G_1$  in the series connection.
- In2 → Out2 伝達関数**: Points to the simplified block  $G_2 * G_1$ .
- In2 → Out2 伝達関数**: Points to the second row of the transfer function matrix, which represents the simplified transfer function.

# 並列結合による簡単化

The screenshot shows the Jamox software interface with three tabs: SerialConnection.jamox, ParallelConnection.jamox, and FeedbackConnection.jamox. The main workspace displays a block diagram where two parallel paths, G1 and G2, are combined into a single block G1+G2. An orange arrow points from the parallel diagram to the simplified block diagram.

Below the diagram, the console window displays the transfer function matrix:

```
>伝達関数行列:  
==== ( 2 x 2) RaMatrix ====  
[ ( 1) ] [ ( 2) ]  
( 1)  $\frac{2*s + 3}{s^2 + 3*s + 2}$  -----  
0 -----  
1
```

Two callout boxes highlight the simplified transfer functions:

- In1 → Out1 伝達関数**:  $\frac{2*s + 3}{s^2 + 3*s + 2}$
- In2 → Out2 伝達関数**:  $\frac{2*s + 3}{s^2 + 3*s + 2}$

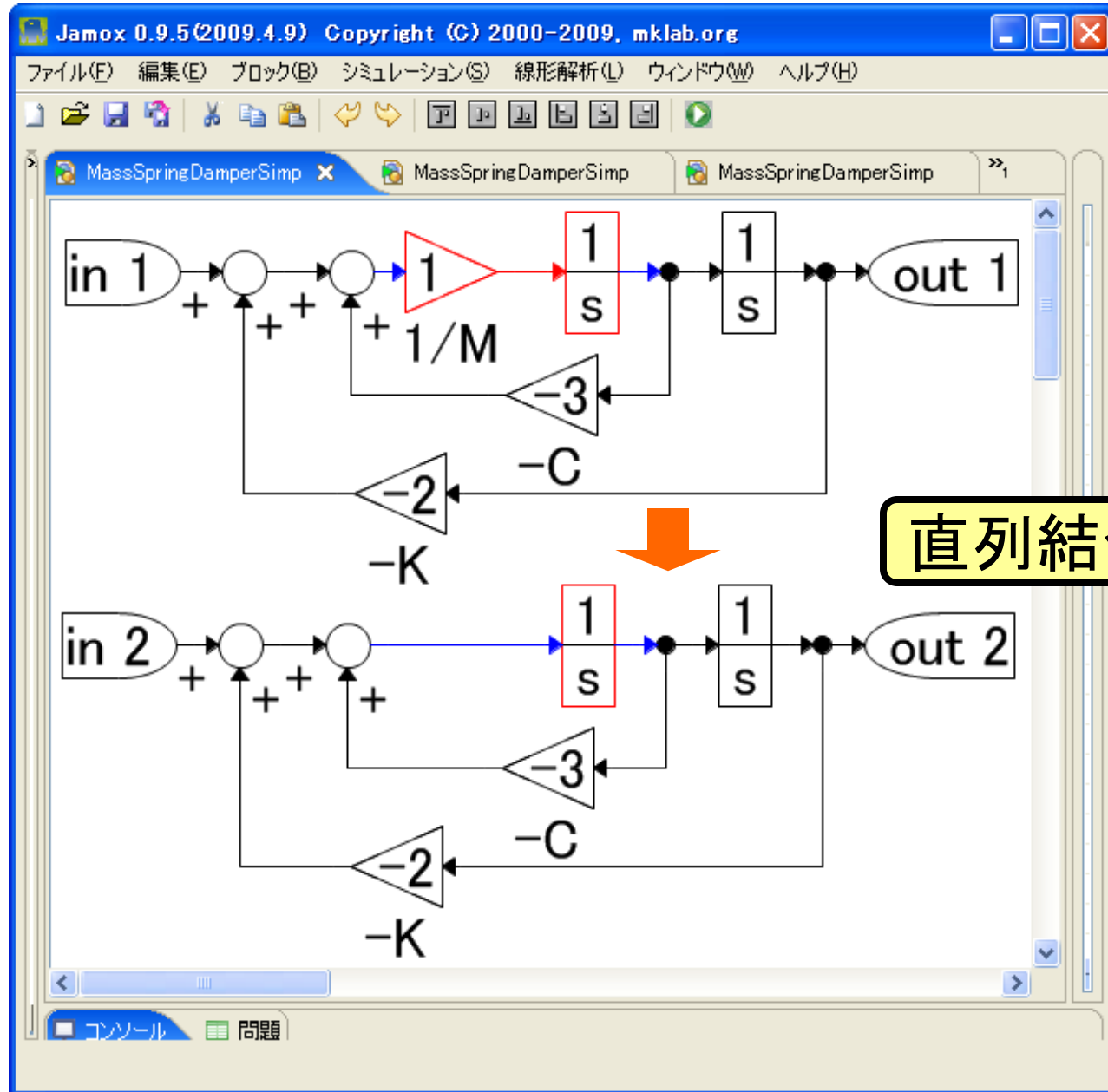
# フィードバック結合による簡単化

The screenshot shows the Jamox software interface. The top window displays a block diagram of a feedback system. An input 'in 1' enters a summing junction with a '+' sign. The output of the summing junction goes to a block labeled 'G1' with transfer function  $\frac{1}{s+1}$ . The output of 'G1' goes to a summing junction with a '-' sign. The output of this second summing junction goes to a block labeled 'G2' with transfer function  $\frac{1}{s+2}$ . The output of 'G2' is fed back to the first summing junction. The output of the first summing junction is 'out 1'. An orange arrow points to the right, where the simplified transfer function is shown: 'in 2' enters a block with transfer function  $\frac{s+2}{s^2+3s+3}$ , and the output is 'out 2'. Below the diagram is a console window with the following text:

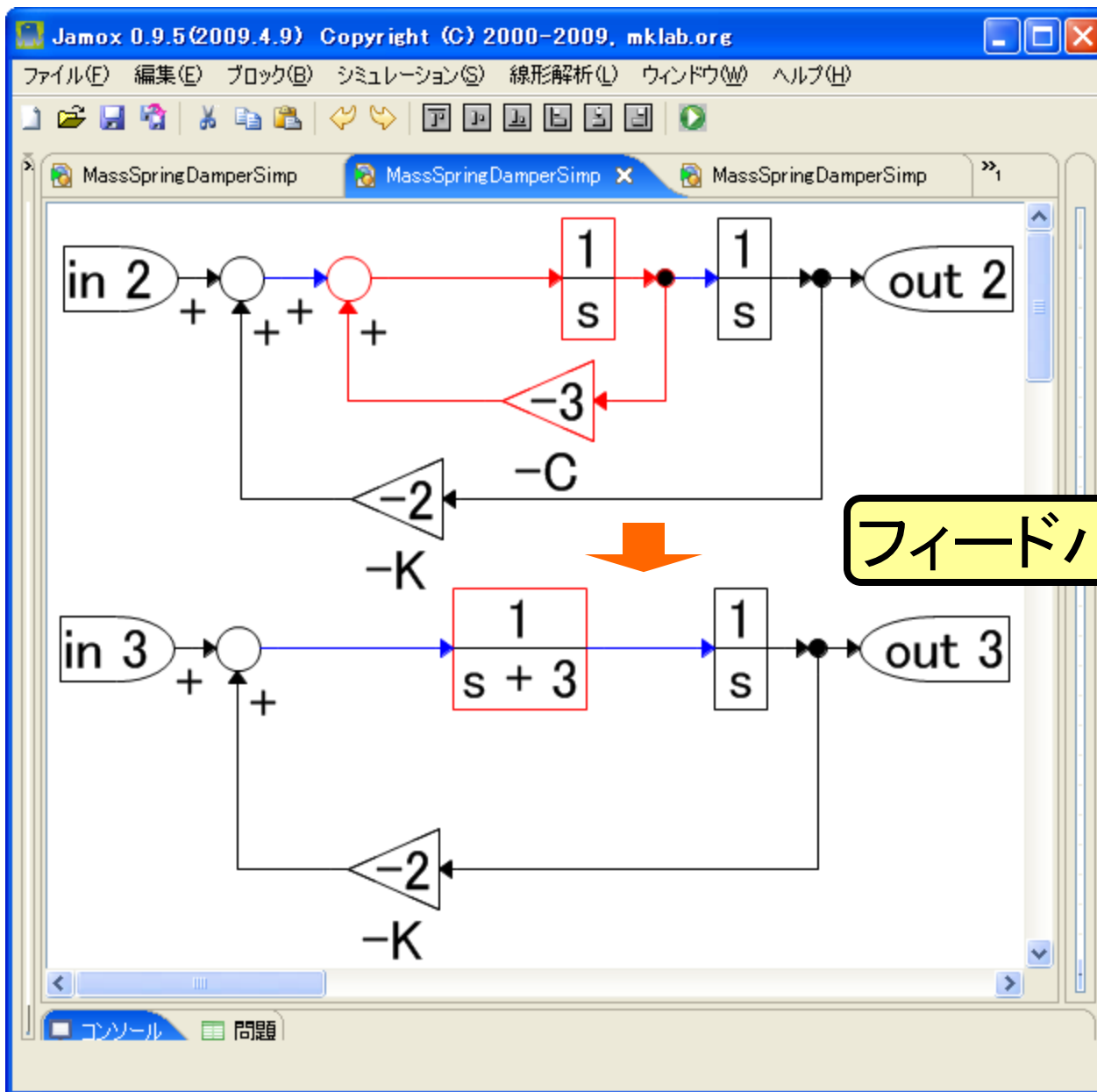
```
>伝達関数行列:  
=== ( 2 x 2) RaMatrix ===  
[ ( 1) ] [ ( 2) ]  
[ s + 2 ] [ 0 ]  
( 1) -----  
[ s^2 + 3*s + 3 ] [ 1 ]  
[ 0 ]  
-----  
[ 1 ]  
[ s + 2 ]  
-----  
[ s^2 + 3*s + 3 ]
```

Two callout boxes highlight specific parts of the transfer function matrix. The first callout, labeled 'In1 → Out1 伝達関数', points to the element  $\frac{s+2}{s^2+3s+3}$  in the first row, first column. The second callout, labeled 'In2 → Out2 伝達関数', points to the element  $\frac{s+2}{s^2+3s+3}$  in the second row, second column.

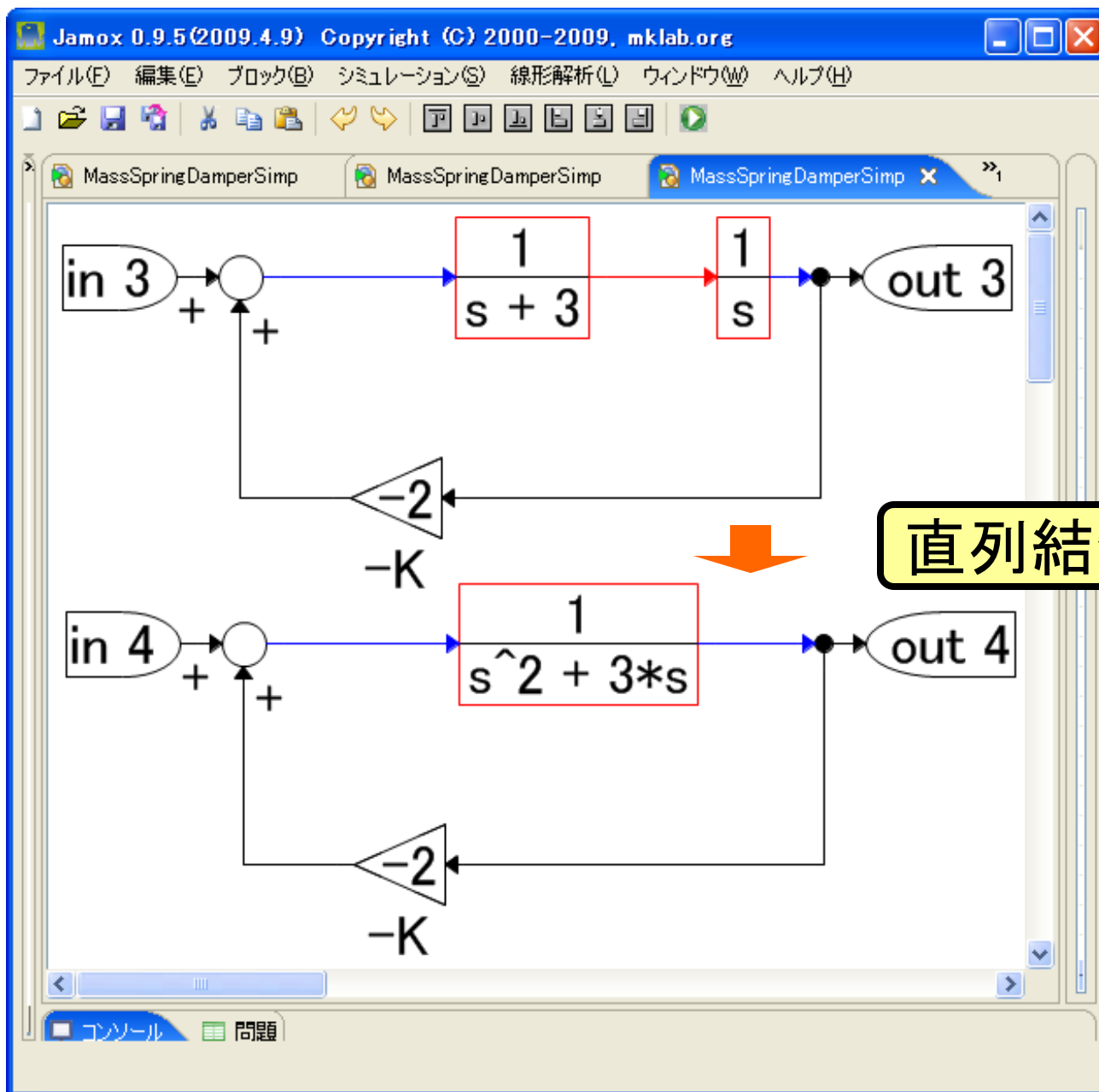
# 質量・ばね・ダンパ系のブロック線図の単純化(1/4)



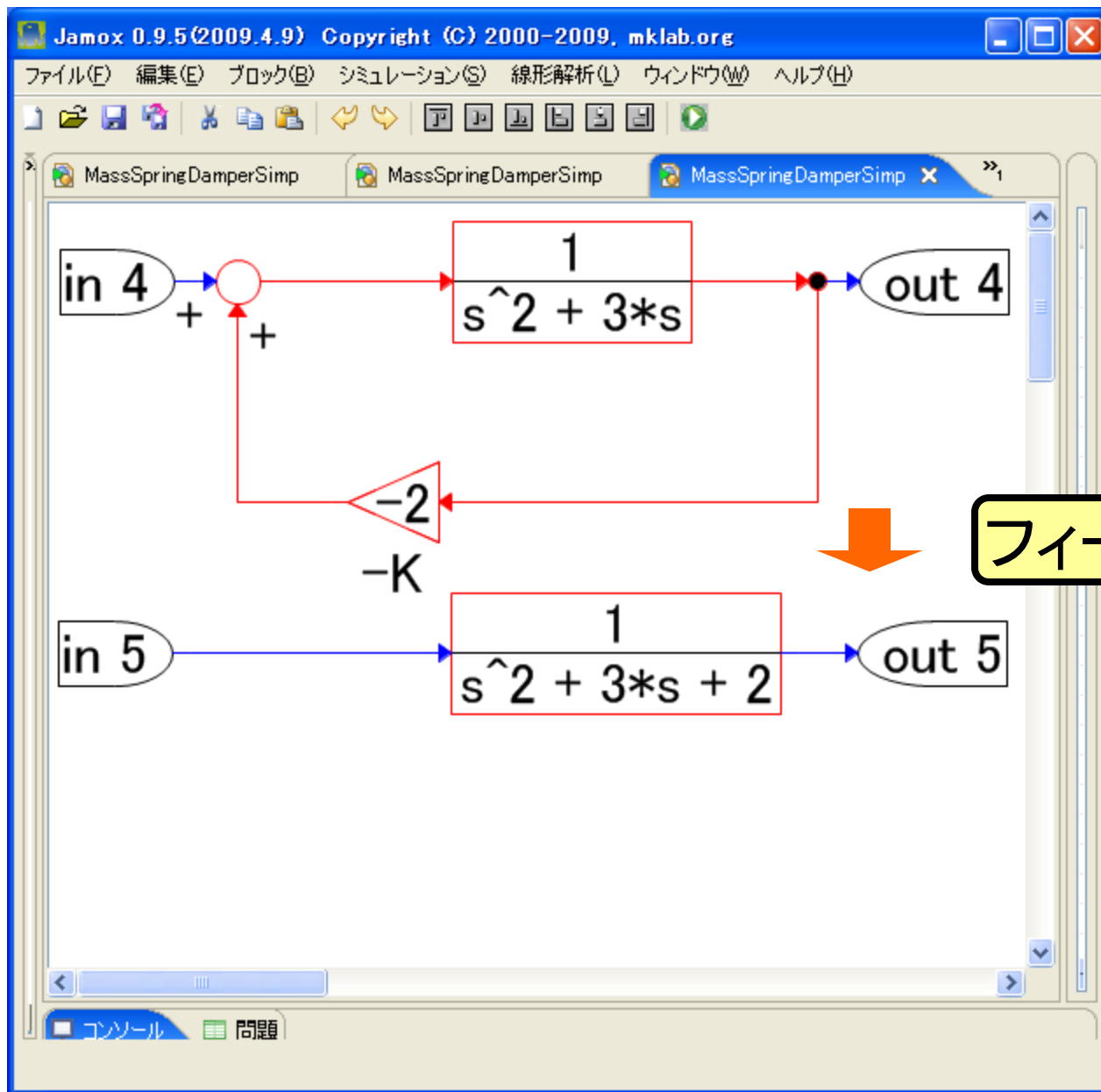
# 質量・ばね・ダンパ系のブロック線図の簡単化(2/4)



# 質量・ばね・ダンパ系のブロック線図の簡単化(3/4)



# 質量・ばね・ダンパ系のブロック線図の簡単化(4/4)



# ブロック線図の等価変換

1. 2個のブロックの入れ替え
2. 2個の加え合わせ点の入れ替え
3. 2個の引き出し点の入れ替え
4. ブロックを加え合せ点の前へ移動
5. ブロックを加え合せ点の後へ移動
6. ブロックを引き出し点の前へ移動
7. ブロックを引き出し点の後ろへ移動

# 2個のブロックの入れ替え

Jamox 0.9.5 (2009.4.9) Copyright (C) 2000-2009. mklab.org

ファイル(F) 編集(E) ブロック(B) シミュレーション(S) 線形解析(L) ウィンドウ(W) ヘルプ(H)

ExchangedBlocks.jamox x

The diagram illustrates the exchange of two transfer function blocks, G1 and G2, in a series connection. On the left, the original system has an input 'in 1' connected to block G1 (transfer function  $\frac{1}{s+1}$ ), which is then connected to block G2 (transfer function  $\frac{1}{s+2}$ ), resulting in output 'out 1'. On the right, the swapped system has an input 'in 2' connected to block G2 (transfer function  $\frac{1}{s+2}$ ), which is then connected to block G1 (transfer function  $\frac{1}{s+1}$ ), resulting in output 'out 2'. A red double-headed arrow indicates the exchange of the two blocks.

in 1 →  $\frac{1}{s+1}$  (G1) →  $\frac{1}{s+2}$  (G2) → out 1

in 2 →  $\frac{1}{s+2}$  (G2) →  $\frac{1}{s+1}$  (G1) → out 2

コンソール 問題

>伝達関数行列:

```
=== ( 2 x 2) RaMatrix ===  
[ ( 1) ]  
[ ( 2) ]  
-----  
-----  
( 1) 1  
-----  
s^2 + 3*s + 2  
-----  
0  
( 2) -----  
1  
-----  
1  
-----  
s^2 + 3*s + 2
```

# 2個の加え合せ点の入れ替え

Jamox 0.9.5 (2009.4.9) Copyright (C) 2000-2009. mklab.org

ファイル(F) 編集(E) ブロック(B) シミュレーション(S) 線形解析(L) ウィンドウ(W) ヘルプ(H)

\*ExchangedSummingPoints.jamox x

in 1 →  $\frac{1}{s+2}$  (G2) → + →  $\frac{1}{s+1}$  (G1) → + → out 1

in 2 →  $\frac{1}{s+2}$  (G2) → + →  $\frac{1}{s+1}$  (G1) → + → out 2

コンソール 問題

>伝達関数行列 :

===	( 2 x 2 )	RaMatrix	===
	[ ( 1 ) ]	[ ( 2 ) ]	
( 1 )	$\frac{s^2 + 5*s + 5}{s^2 + 3*s + 2}$	0	-----
		1	
( 2 )	0	$\frac{s^2 + 5*s + 5}{s^2 + 3*s + 2}$	-----
	1		

# 2個の引き出し点の入れ替え

Jamox 0.9.5(2009.4.9) Copyright (C) 2000-2009. mklab.org

ファイル(F) 編集(E) ブロック(B) シミュレーション(S) 線形解析(L) ウィンドウ(W) ヘルプ(H)

ExchangedBranchPoints.jamox

in 1 →  $\frac{1}{s+2}$  (G2) → + → out 1  
 in 1 →  $\frac{1}{s+1}$  (G1) → + → out 1

in 2 →  $\frac{1}{s+2}$  (G2) → + → out 2  
 in 2 →  $\frac{1}{s+1}$  (G1) → + → out 2

コンソール 問題

>伝達関数行列:

=== ( 2 x 2) RaMatrix ===  
 [ ( 1) ] [ ( 2) ]  
 ( 1)  $\frac{s^2 + 5*s + 5}{s^2 + 3*s + 2}$  -----  
 0 1

( 2) -----  
 0  $\frac{s^2 + 5*s + 5}{s^2 + 3*s + 2}$

# ブロックを加え合せ点の前へ移動

Jamox 0.9.5 (2009.4.9) Copyright (C) 2000-2009. mklab.org

ファイル(F) 編集(E) ブロック(B) シミュレーション(S) 線形解析(L) ウィンドウ(W) ヘルプ(H)

ExchangedBlock.SummingPoint2.jamox

in 1 →  $\frac{s+2}{s+1}$  (G1) → + → out 1

in 2 →  $\frac{s+1}{s+2}$  (1/G1) → + →  $\frac{s+2}{s+1}$  (G1) → out 2

**逆数を追加**

コンソール 問題

>伝達関数行列:

===	$\begin{pmatrix} 2 & x & 2 \\ & (1) & \end{pmatrix}$	RaMatrix	===
( 1)	$\frac{s+3}{s^2+3*s+2}$		$\frac{0}{1}$
( 2)	$\frac{0}{1}$		$\frac{s+3}{s^2+3*s+2}$

# ブロックを加え合せ点の後へ移動

Jamox 0.9.5(2009.4.9) Copyright (C) 2000-2009. mklab.org

ファイル(F) 編集(E) ブロック(B) シミュレーション(S) 線形解析(L) ウィンドウ(W) ヘルプ(H)

ExchangedBlockSummingPoint1.jamox

in 1

out 1

1

$s + 2$

G2

+

+

1

$s + 1$

G1

in 2

out 2

1

$s + 1$

G1

+

+

1

$s + 2$

G2

1

$s + 1$

G1

同一ブロックを追加

コンソール

問題

>伝達関数行列:

=== ( 2 x 2) RaMatrix ===

[ ( 1) ]

[ ( 2) ]

( 1)  $\frac{s + 3}{s^2 + 3*s + 2}$   $\frac{0}{1}$

( 2)  $\frac{0}{1}$   $\frac{s + 3}{s^2 + 3*s + 2}$

# ブロックを引き出し点の前へ移動

The screenshot shows the Jamox software interface. The top part displays a block diagram with two input nodes, 'in 1' and 'in 2', and two output nodes, 'out 1' and 'out 2'. The diagram illustrates the movement of a block  $G_1$  (represented as  $\frac{1}{s+1}$ ) from its original position to a junction point before a summing junction. An orange arrow points to the new configuration. The bottom part of the screenshot shows the console window with the transfer function matrix:

コンソール 問題

>伝達関数行列:  
 === ( 2 x 2) RaMatrix ===  
 [ ( 1) [ ( 2) ]  
 s + 3 0  
 -----  
 ( 1) s^2 + 3\*s + 2 1  
 ( 2) 0 s + 3  
 -----  
 1 s^2 + 3\*s + 2

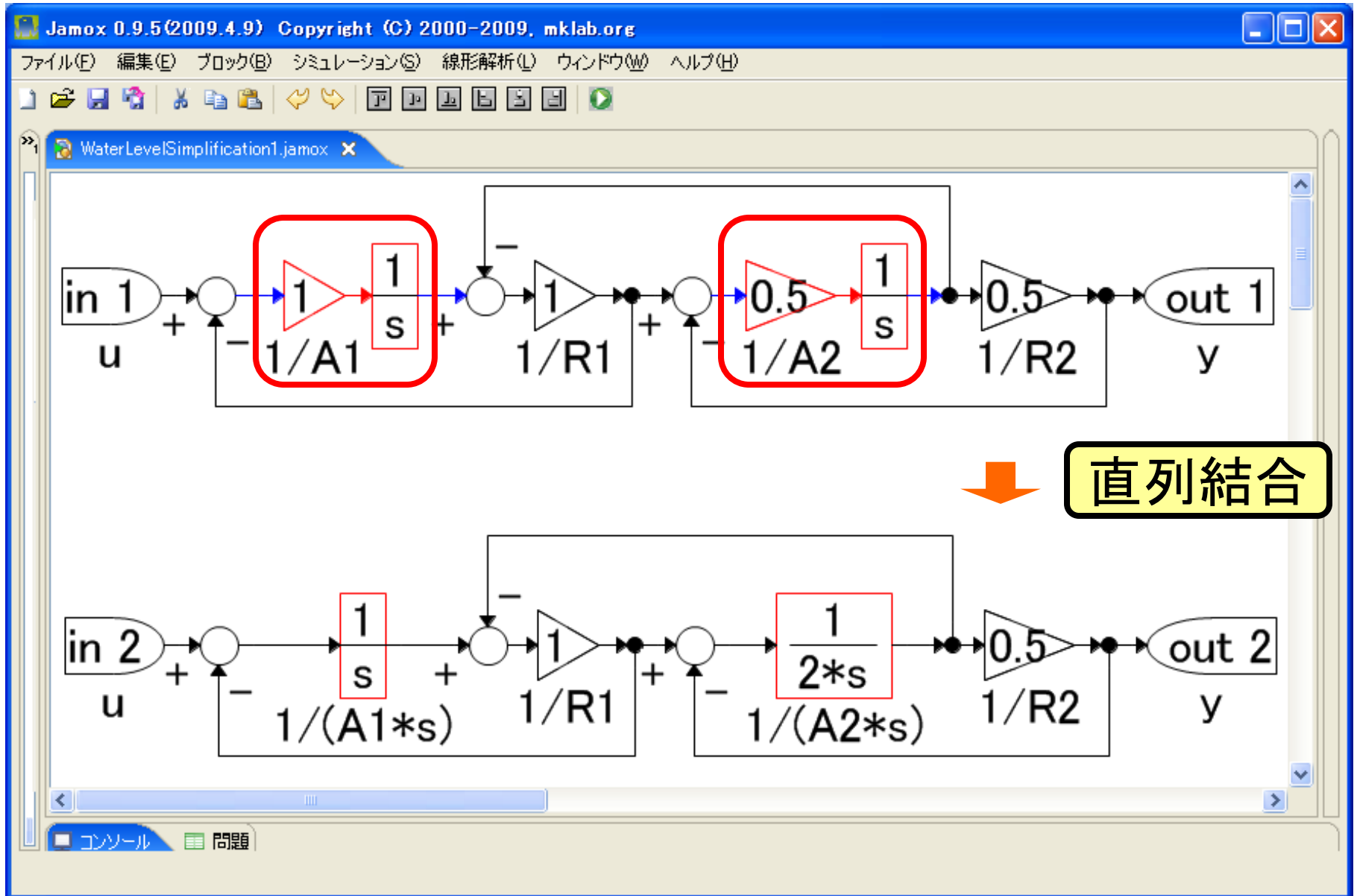
同一ブロックを追加

# ブロックを引き出し点の後ろへ移動

The screenshot shows the Jamox software interface. The main window displays a block diagram with two input nodes, 'in 1' and 'in 2', and two output nodes, 'out 1' and 'out 2'. A block labeled 'G1' with transfer function  $\frac{s+2}{s+1}$  is shown in two positions. An orange arrow points from the top position (before the summing junction) to the bottom position (after the summing junction). A green block labeled '1/G1' with transfer function  $\frac{s+1}{s+2}$  is also shown. A yellow callout box with red text says '逆数を追加' (Add inverse). The console window at the bottom shows the transfer function matrix:

```
>伝達関数行列:  
=== ( 2 x 2) RaMatrix ===  
[[ ( 1) ] [( 2) ]  
( 1)  $\frac{2*s + 3}{s + 1}$  0  
      -----  
      s + 1  
      1  
( 2) 0  $\frac{2*s + 3}{s + 1}$   
      -----  
      1
```

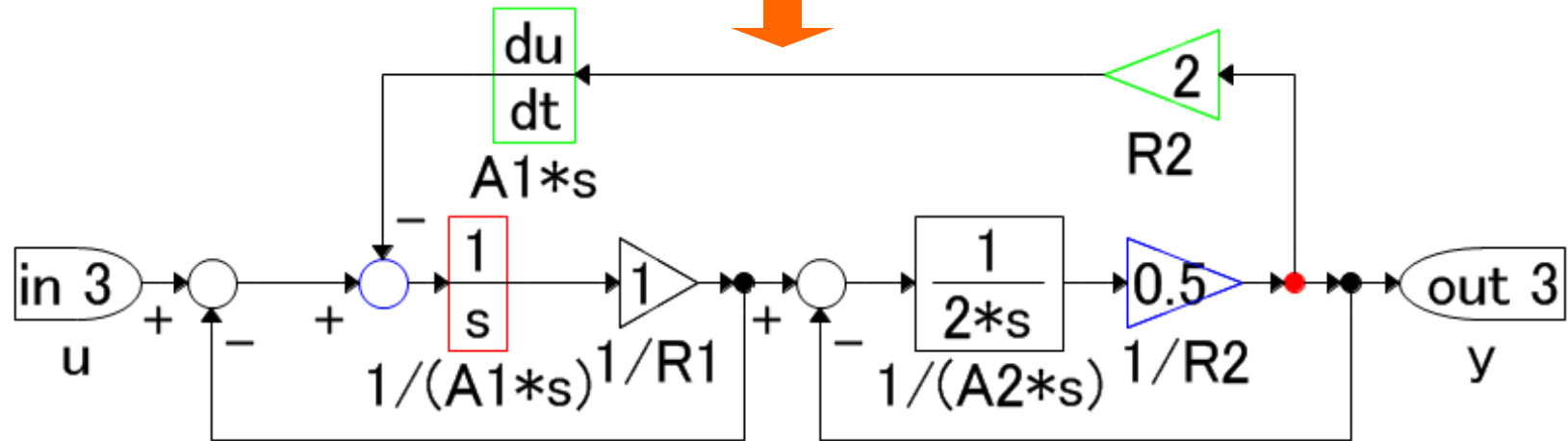
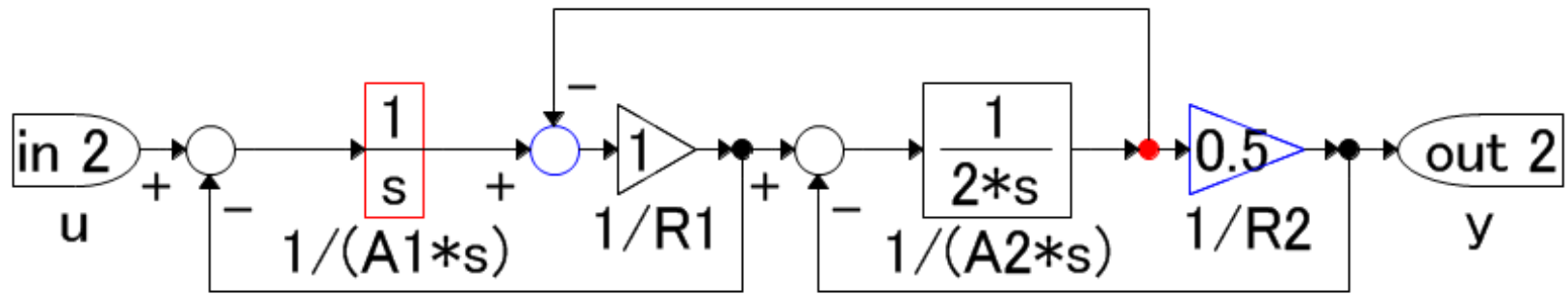
# 液面系のブロック線図の簡単化(1/7)



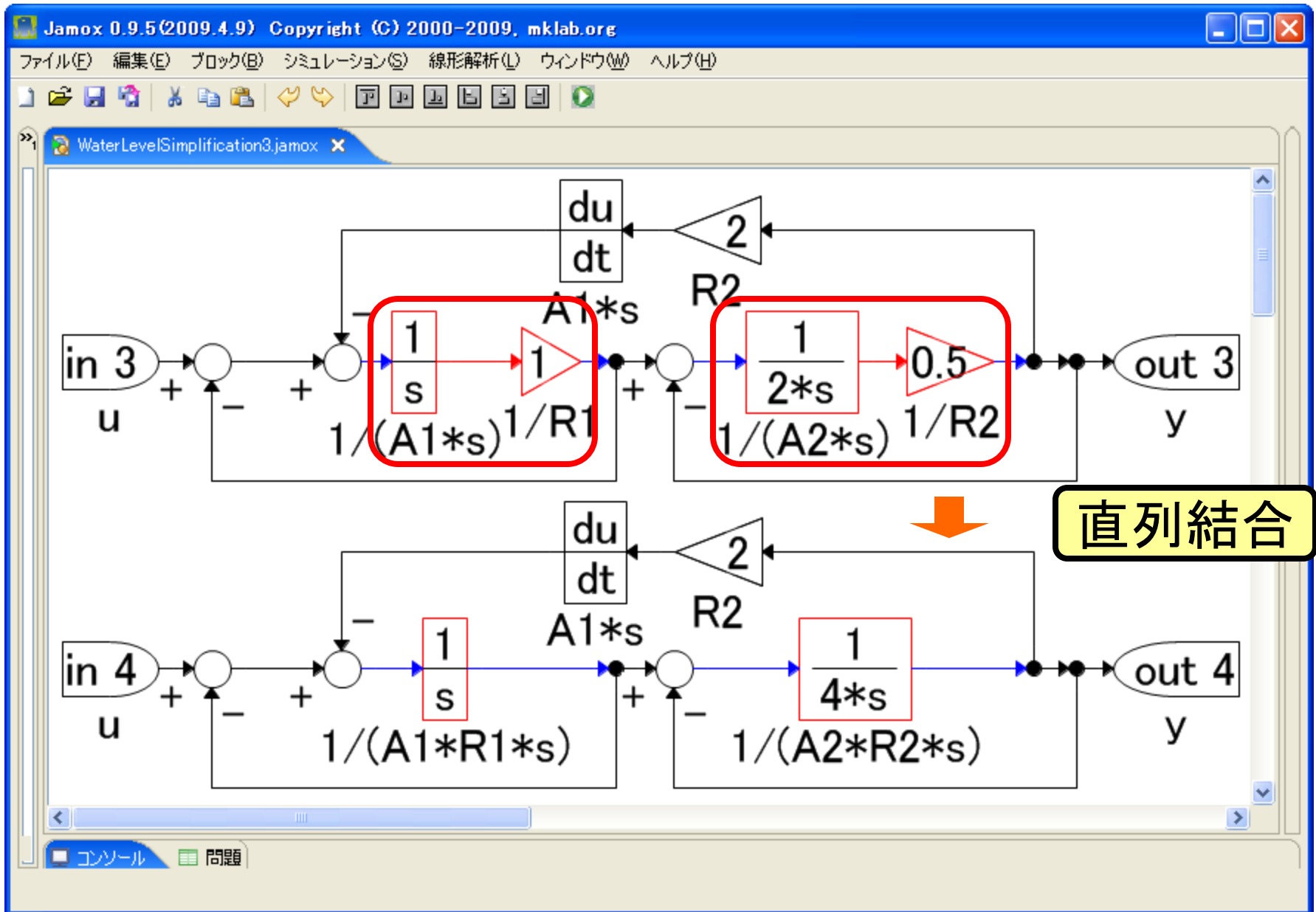
# 液面系のブロック線図の簡単化(2/7)

ブロックを加え合せ点の前へ移動

ブロックを引き出し点の後へ移動



# 液面系のブロック線図の簡単化(3/7)



# 液面系のブロック線図の簡単化(4/7)

Jamox 0.9.5 (2009.4.9) Copyright (C) 2000-2009. mklab.org

ファイル(F) 編集(E) ブロック(B) シミュレーション(S) 線形解析(L) ウィンドウ(W) ヘルプ(H)

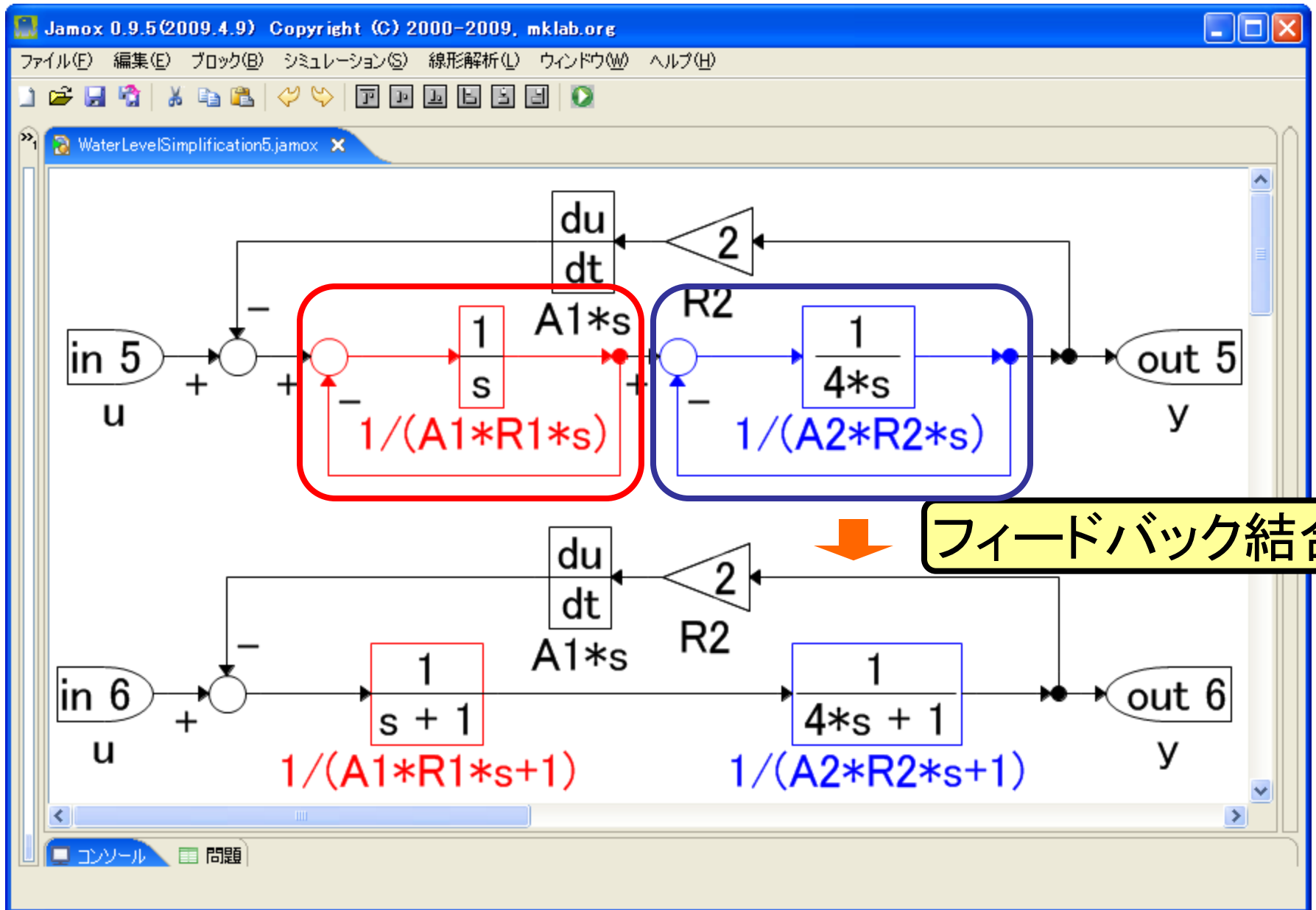
**加え合せ点の入れ替え**      **引き出し点の入れ替え**

The diagram illustrates the simplification of a liquid level system's block diagram. It shows two versions of the system, connected by a downward-pointing orange arrow.

**Top Diagram (Original):** The input  $u$  (labeled "in 4") enters a summing junction (red circle) with a positive sign. The output of this junction goes to another summing junction (blue circle) with a positive sign. The output of the blue junction goes through a block  $\frac{1}{s}$  to a summing junction (black circle) with a positive sign. This junction also receives a negative feedback signal from a block  $\frac{1}{A_1 R_1 s}$ . The output of this junction goes to a summing junction (black circle) with a positive sign. This junction also receives a negative feedback signal from a block  $\frac{1}{A_2 R_2 s}$ . The output of this junction goes through a block  $\frac{1}{4s}$  to a summing junction (red circle) with a positive sign. This junction also receives a negative feedback signal from a block  $\frac{2}{du/dt}$ . The output of this junction goes to a summing junction (blue circle) with a positive sign. The output of this junction is the system output  $y$  (labeled "out 4").

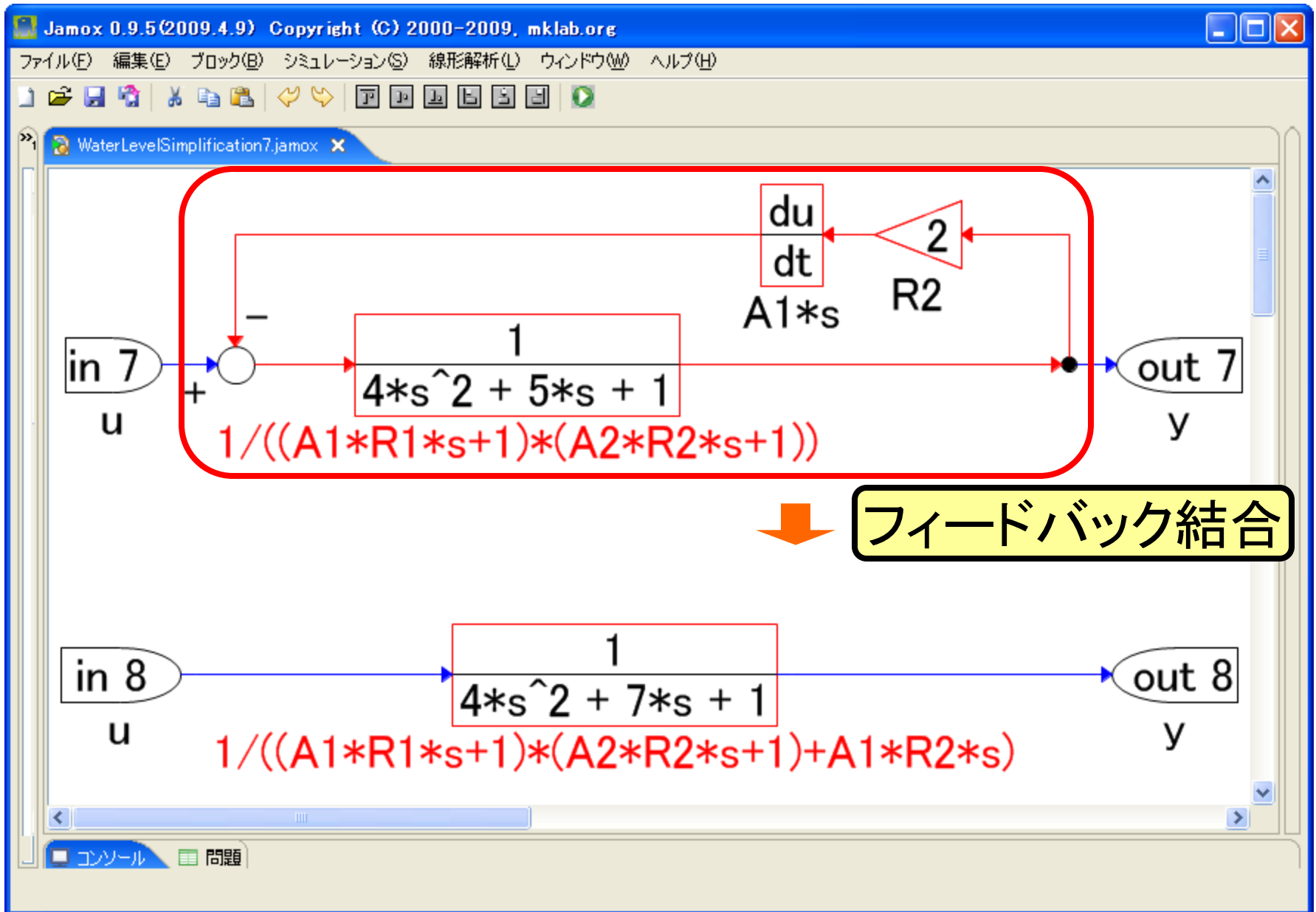
**Bottom Diagram (Simplified):** The input  $u$  (labeled "in 5") enters a summing junction (blue circle) with a positive sign. The output of this junction goes to a summing junction (red circle) with a positive sign. The output of the red junction goes through a block  $\frac{1}{s}$  to a summing junction (black circle) with a positive sign. This junction also receives a negative feedback signal from a block  $\frac{1}{A_1 R_1 s}$ . The output of this junction goes to a summing junction (black circle) with a positive sign. This junction also receives a negative feedback signal from a block  $\frac{1}{A_2 R_2 s}$ . The output of this junction goes through a block  $\frac{1}{4s}$  to a summing junction (blue circle) with a positive sign. This junction also receives a negative feedback signal from a block  $\frac{2}{du/dt}$ . The output of this junction goes to a summing junction (red circle) with a positive sign. The output of this junction is the system output  $y$  (labeled "out 5").

# 液面系のブロック線図の簡単化(5/7)

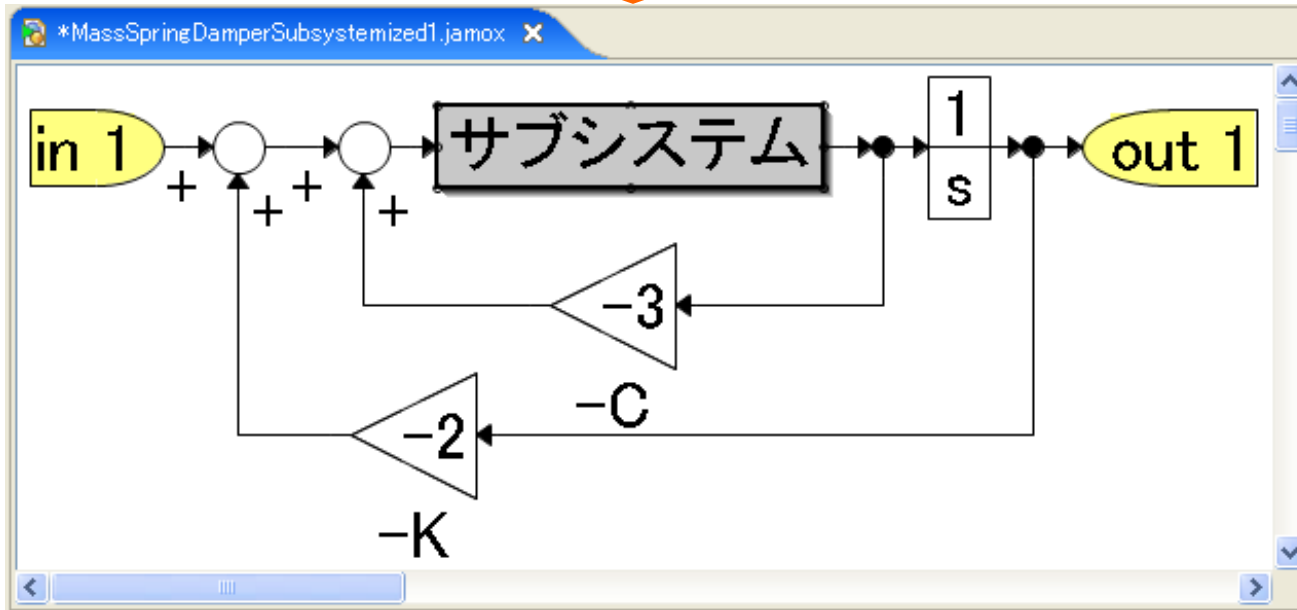
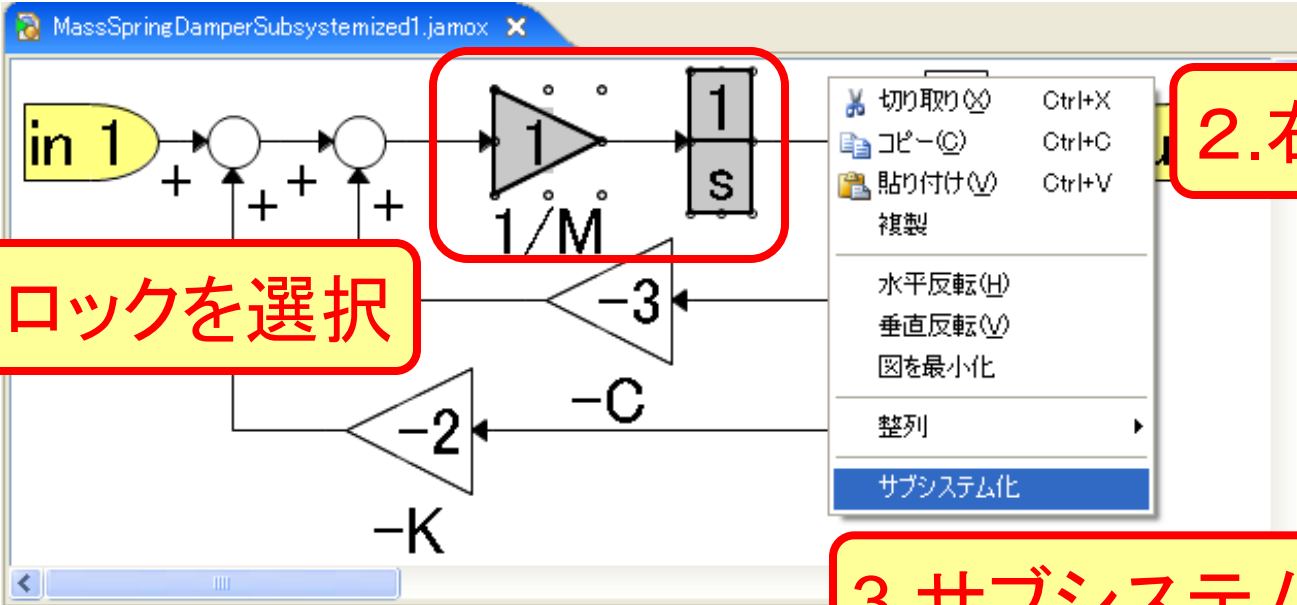




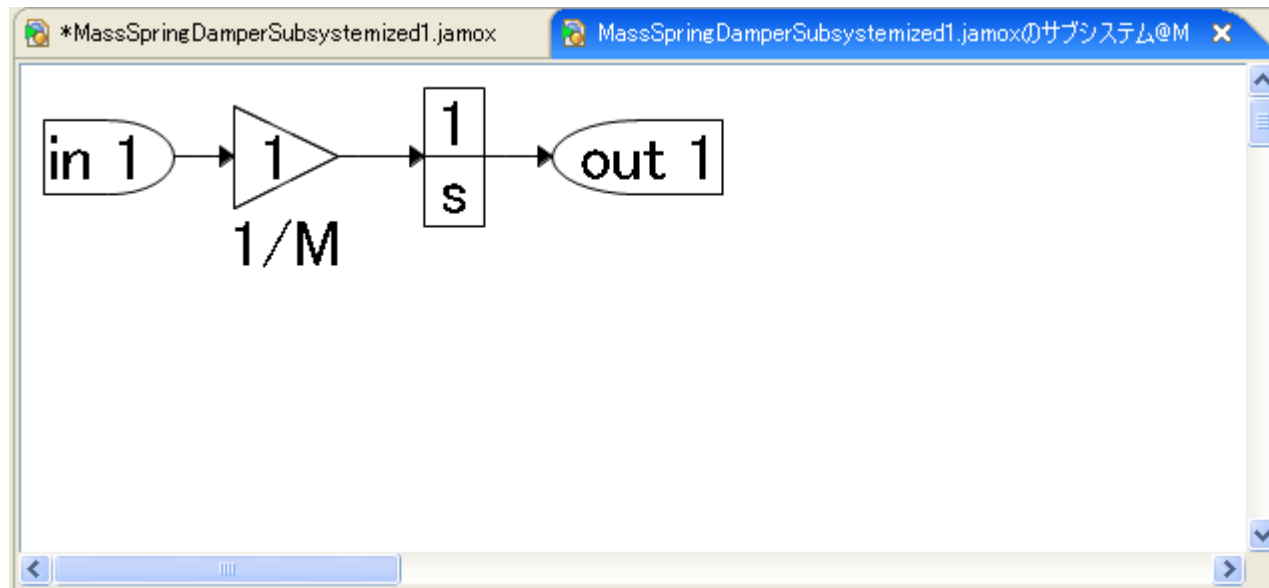
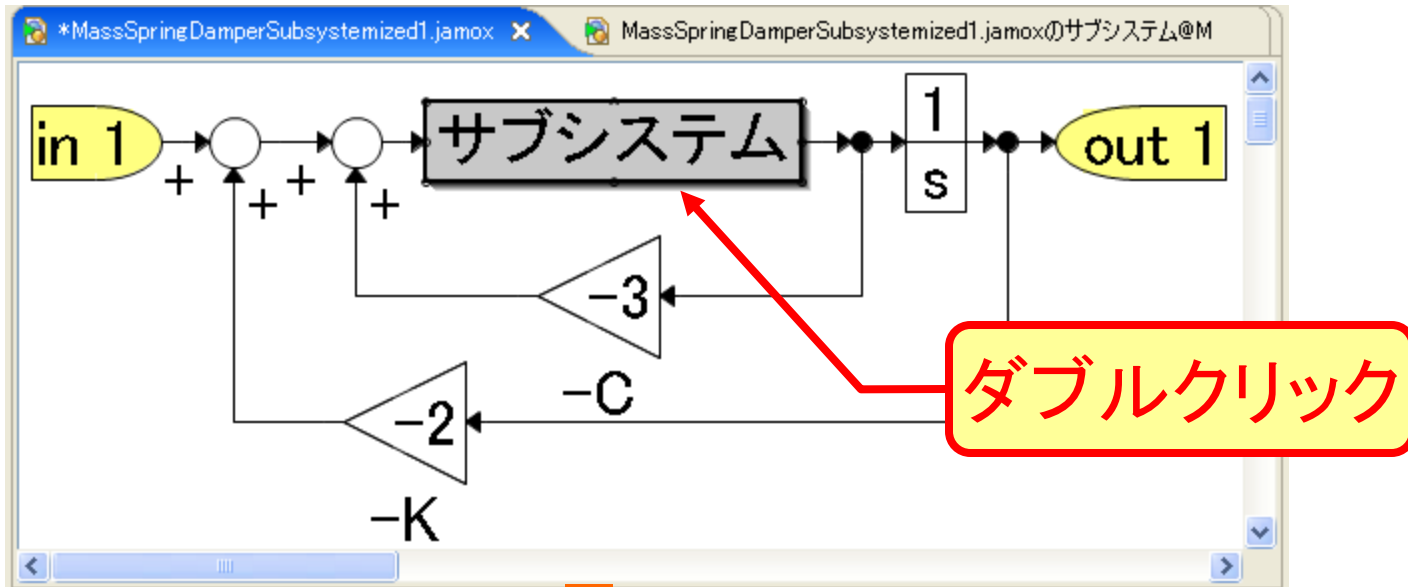
# 液面系のブロック線図の簡単化(7/7)



# サブシステム化によるブロック線図の簡単化



# サブシステムの表示



# サブシステムのパラメータ設定

1.ブロックを選択

2.右クリック

3.パラメータ設定選択

MassSpringDamperSubsystemized1.jamox

切り取り(Ctrl+X)  
コピー(Ctrl+C)  
貼り付け(Ctrl+V)  
複製

水平反転(H)  
垂直反転(V)  
図を最小化

影をつける  
名前を本体に表示  
ポート名を表示

サブシステムを開く  
ライブラリとして保存

削除(D) Ctrl+D  
パラメータ設定(P) Alt+Enter

in 1

+

+

+

サブシステム

$1/(M*s)$

-2

-C

-K

2.右クリック

3.パラメータ設定選択



パラメータ設定:1...

基本 システム 図 ポート

ブロック名

状態

入力数:1 出力数:1 状態数:1

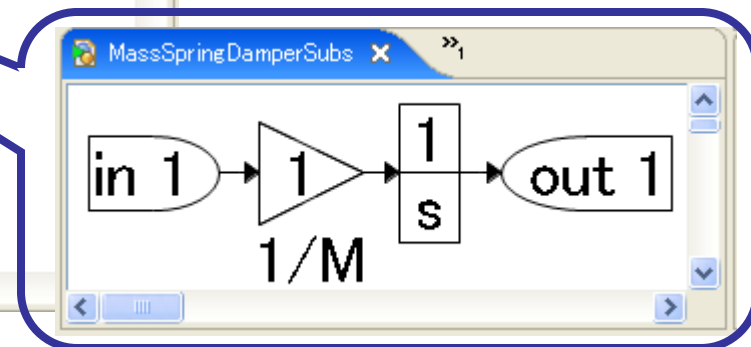
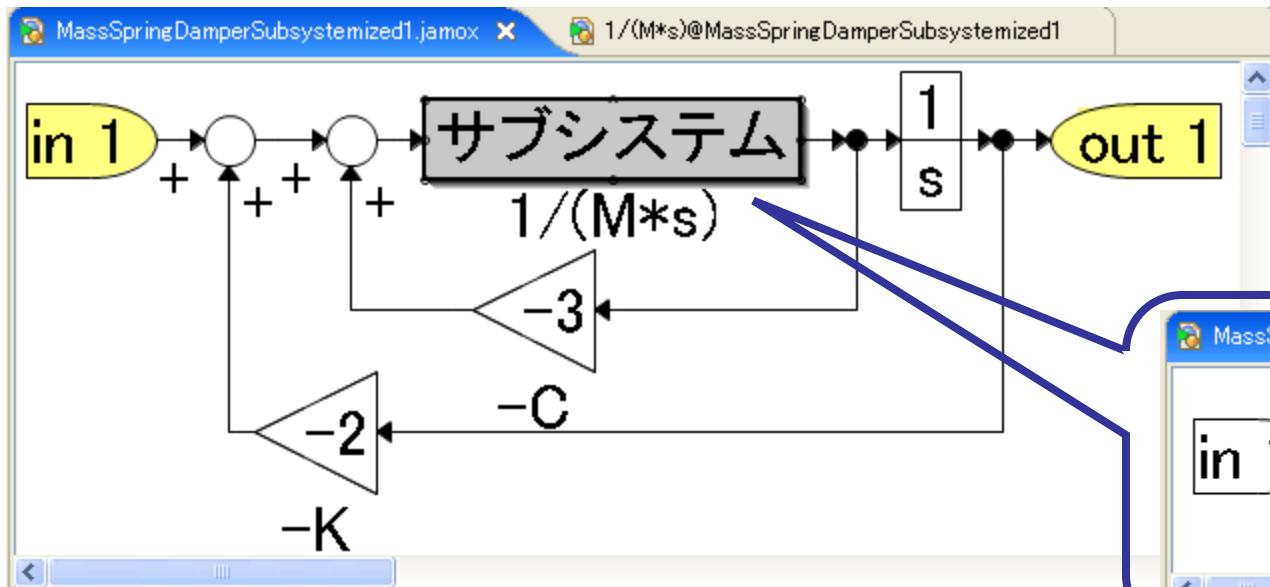
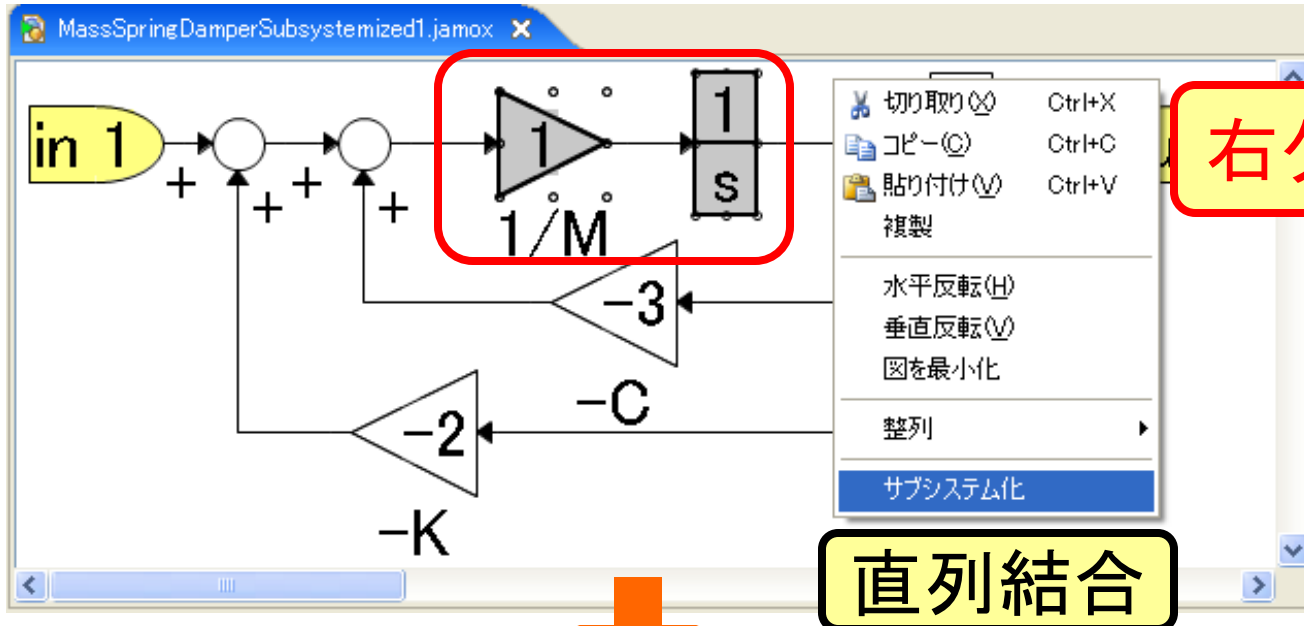
ステータス: NORMAL

ブロックの説明

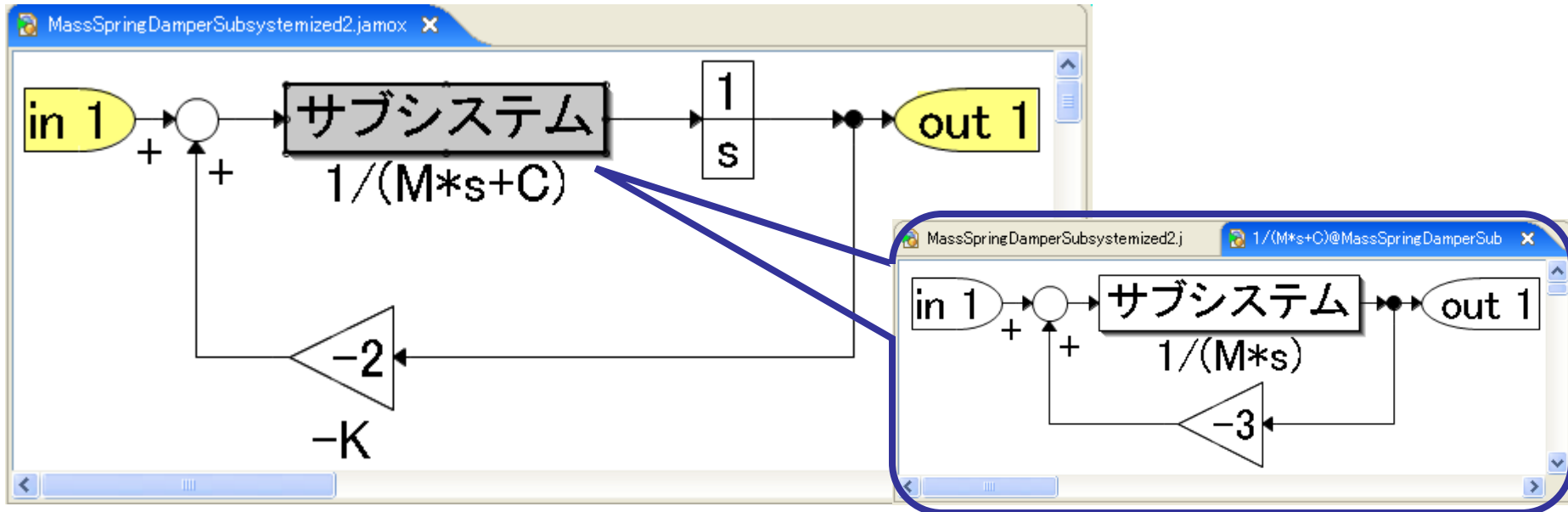
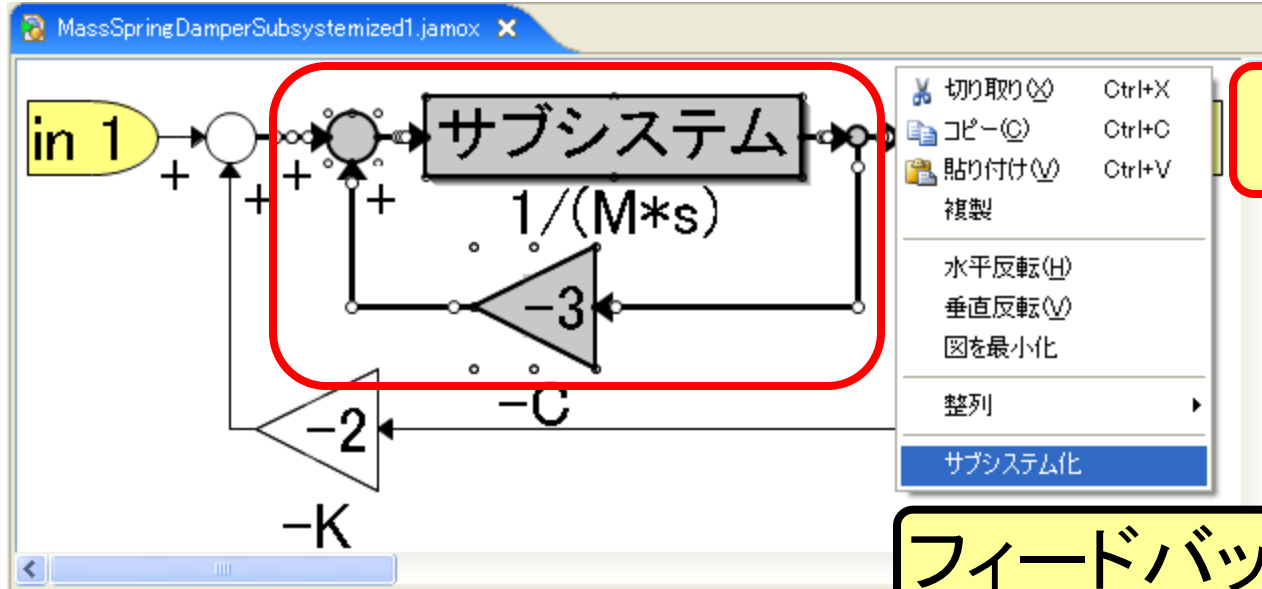
サブシステムを表すブロックです。

了解 キャンセル 適用

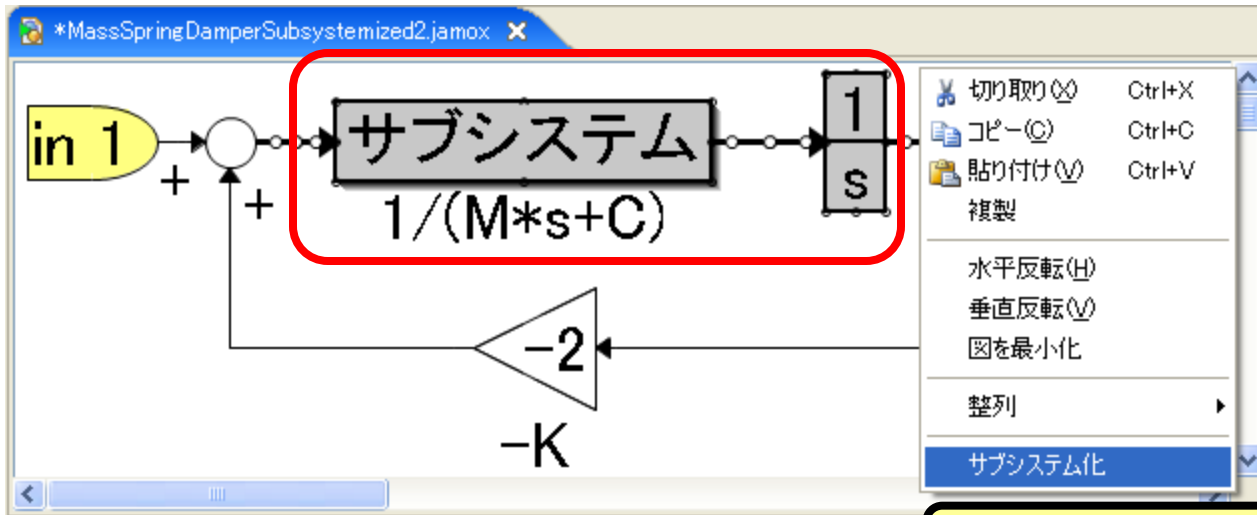
# 質量・ばね・ダンパ系のサブシステム化(1/4)



# 質量・ばね・ダンパ系のサブシステム化(2/4)

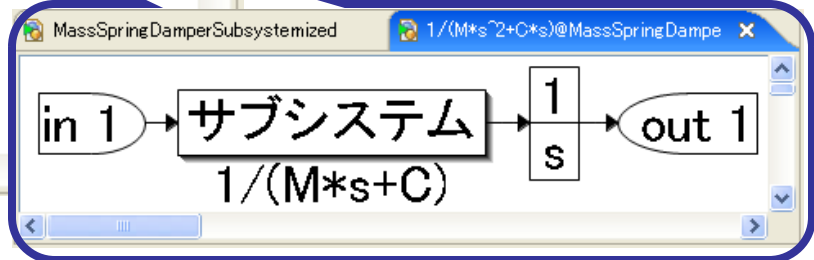
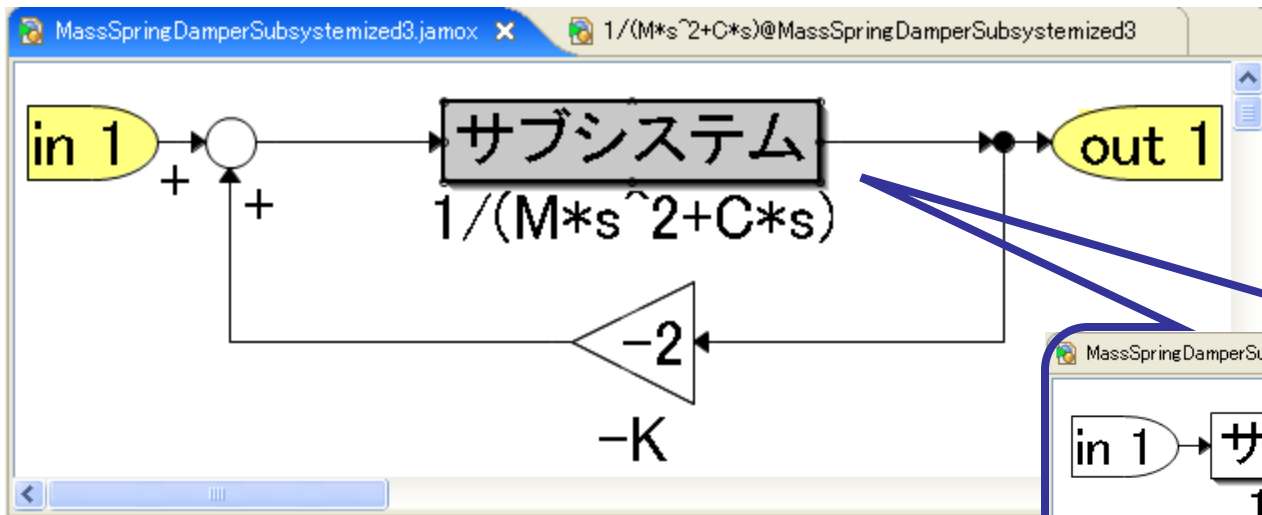


# 質量・ばね・ダンパ系のサブシステム化(3/4)

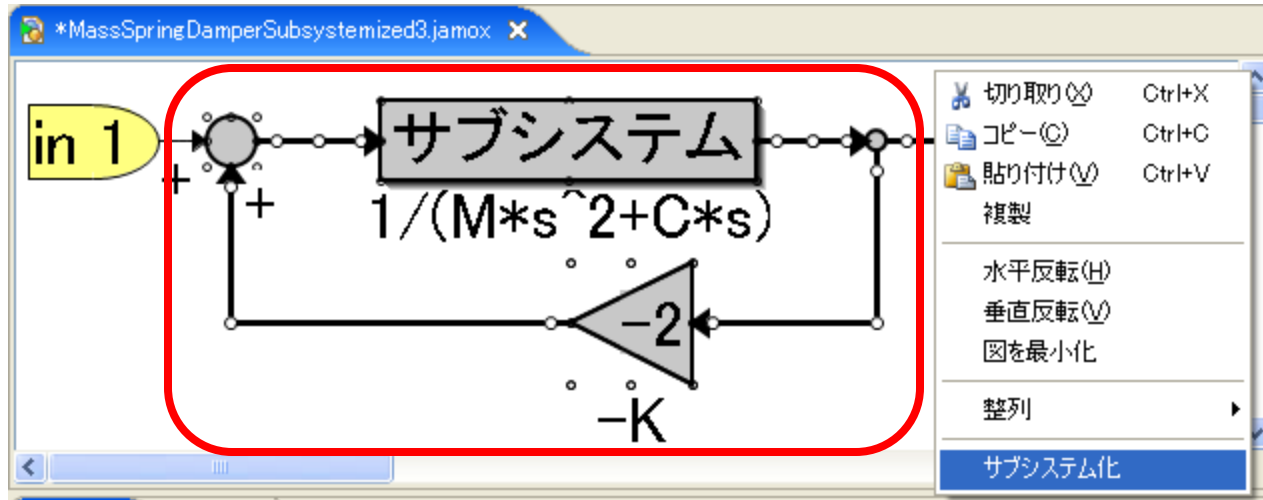


右クリック

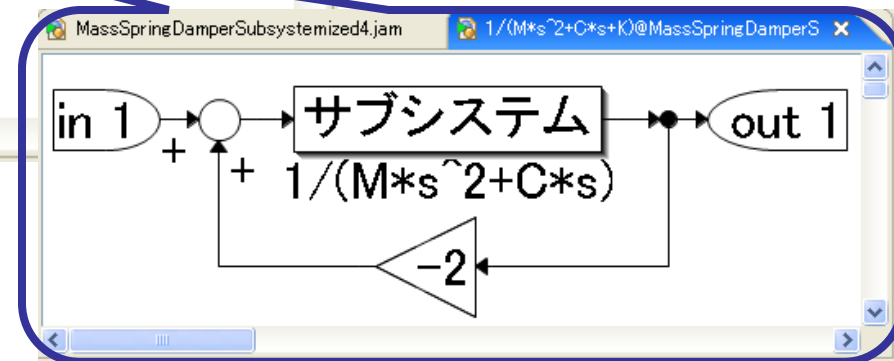
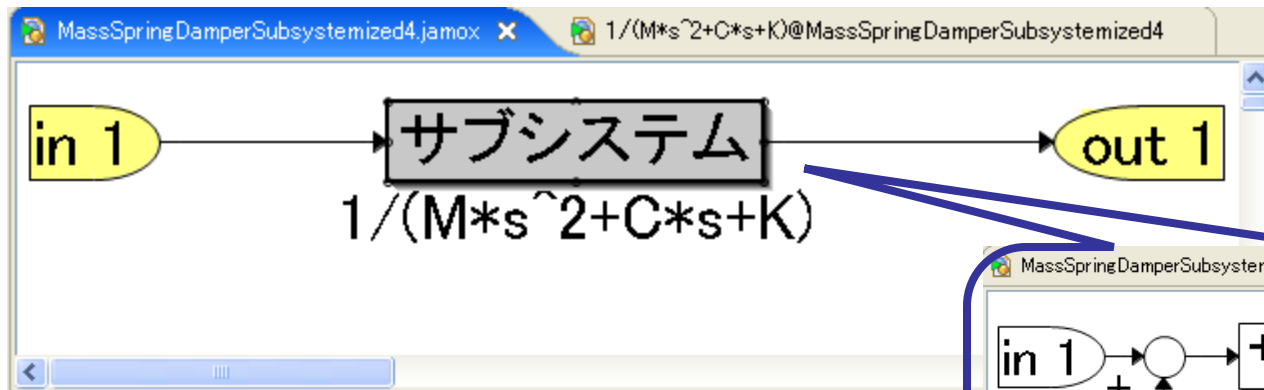
直列結合



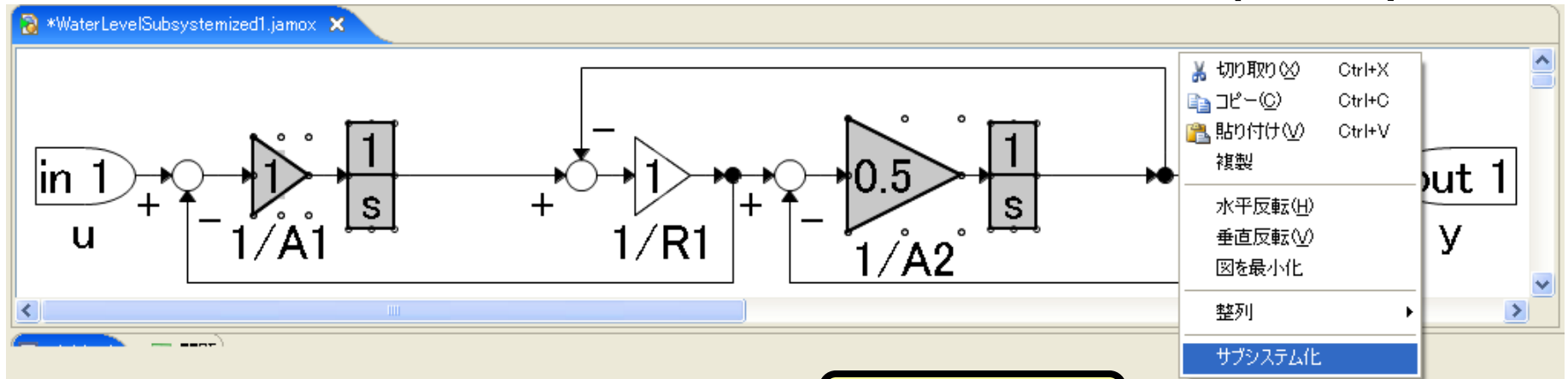
# 質量・ばね・ダンパ系のサブシステム化(4/4)



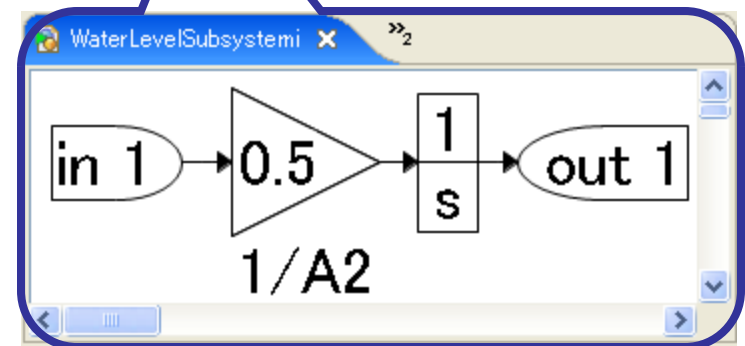
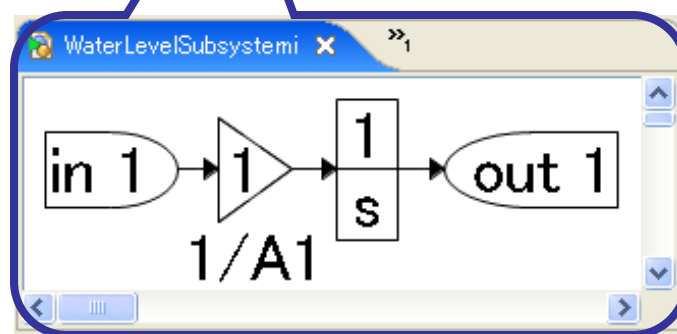
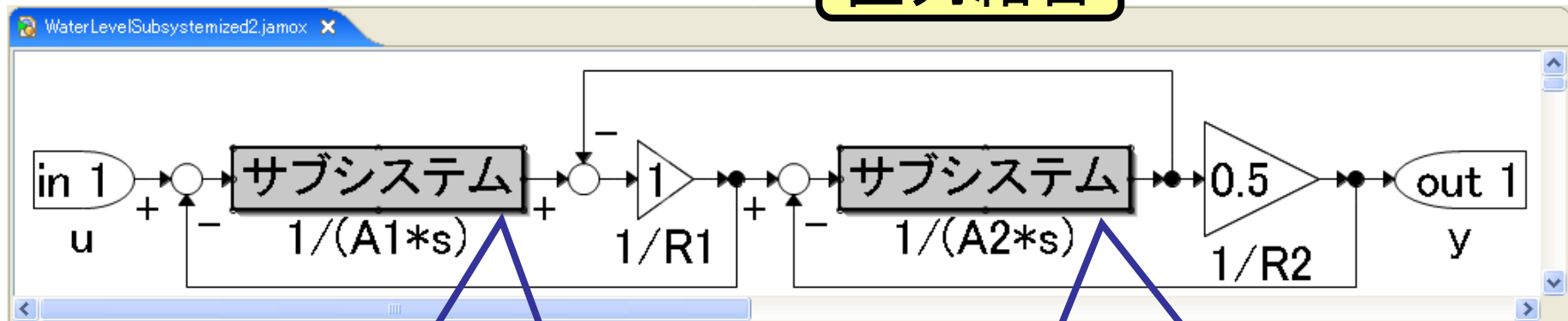
フィードバック結合



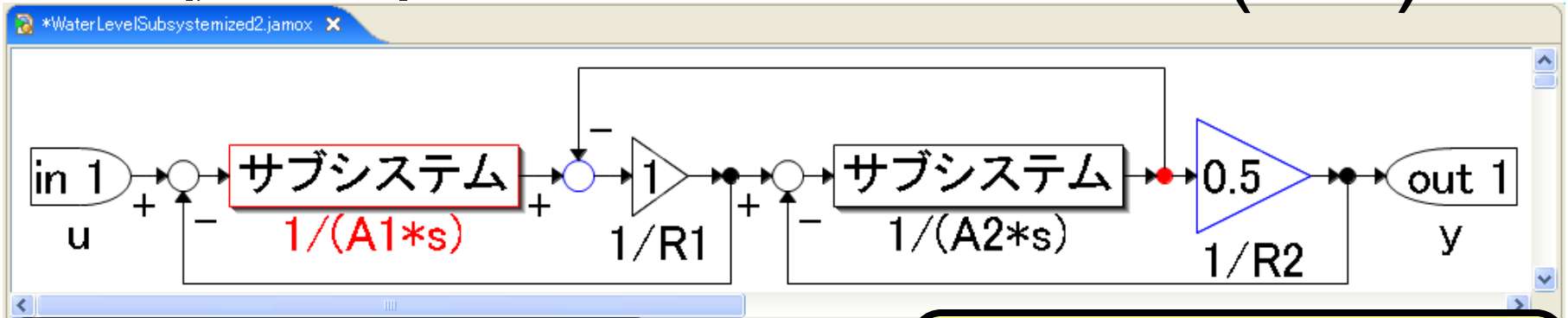
# 液面系のサブシステム化(1/7)



直列結合



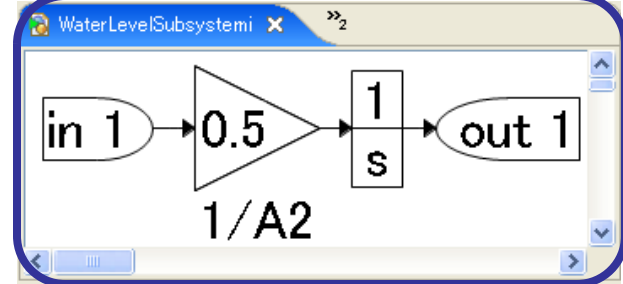
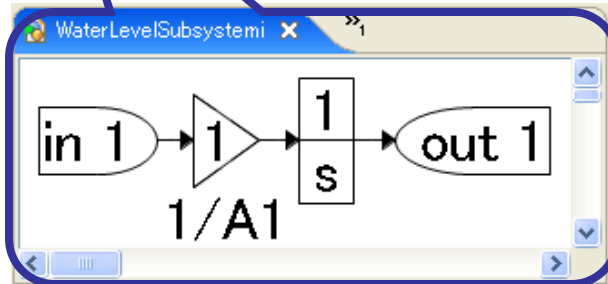
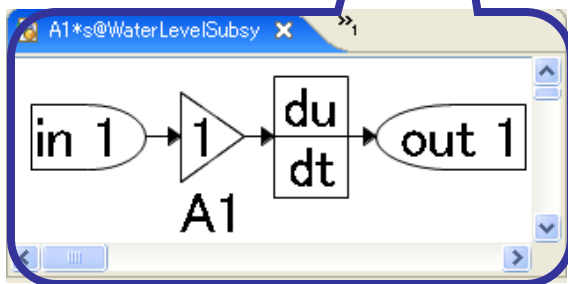
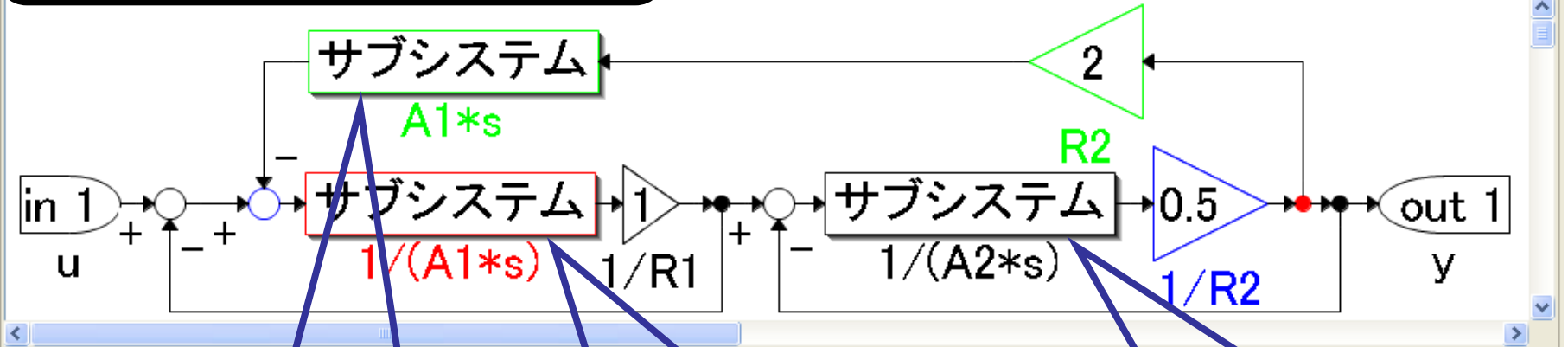
# 液面系のサブシステム化(2/7)



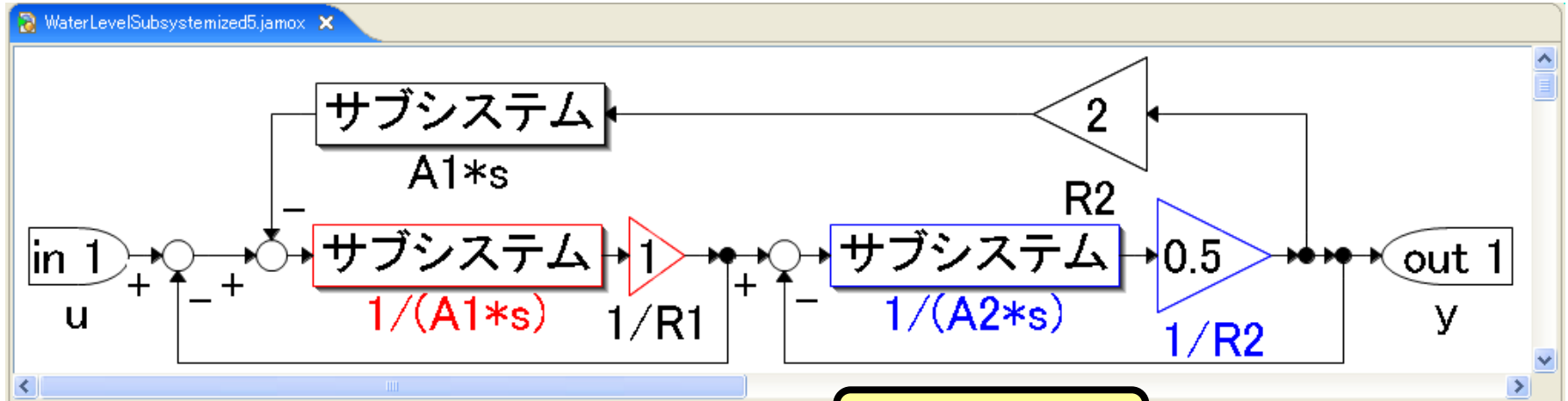
ブロックと加え合せ点の  
前へ移動



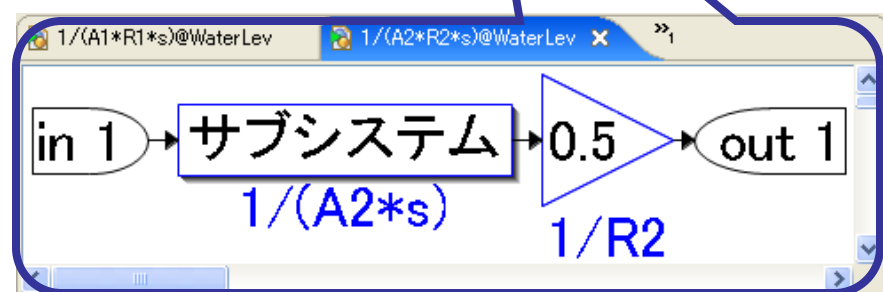
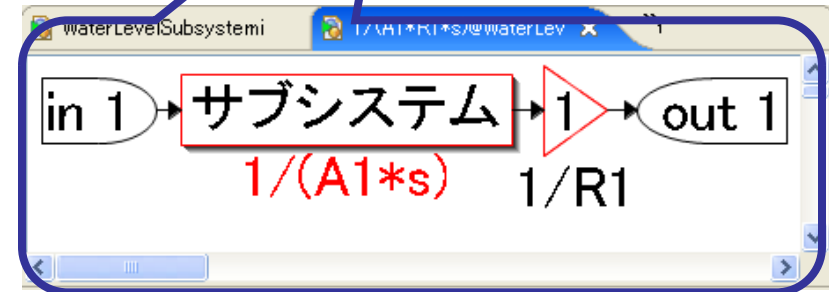
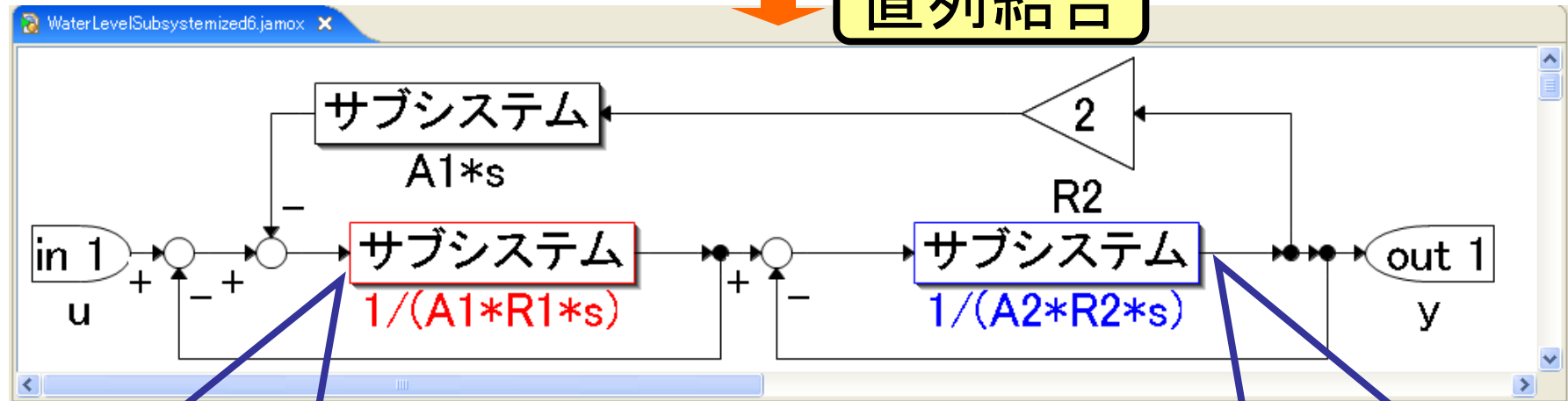
ブロックと引き出し点の  
後へ移動



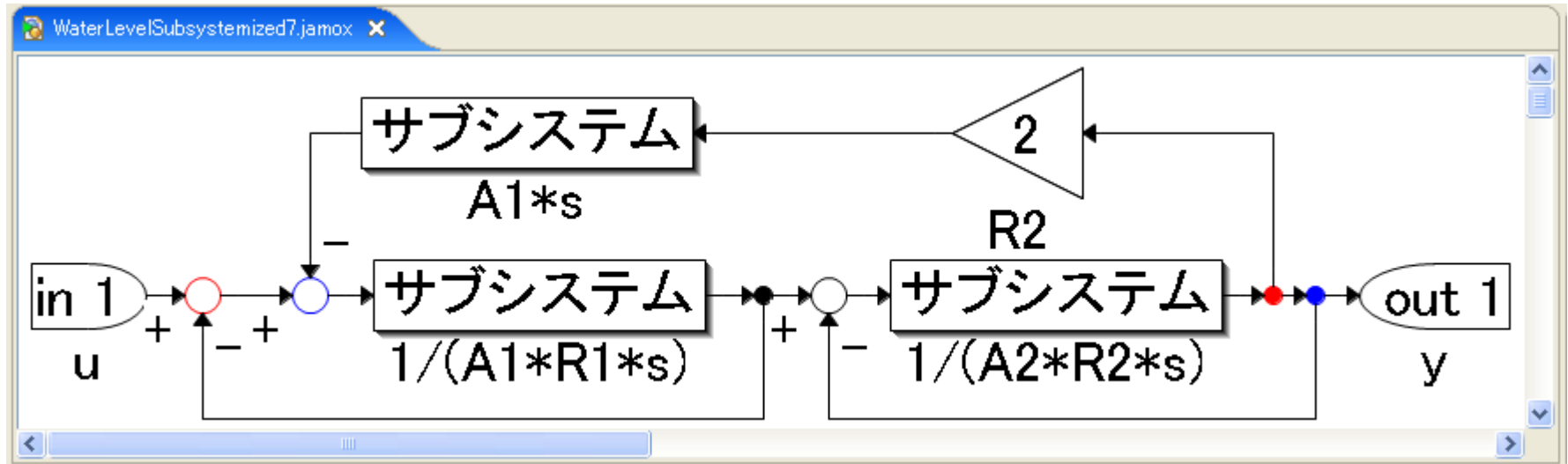
# 液面系のサブシステム化(3/7)



直列結合



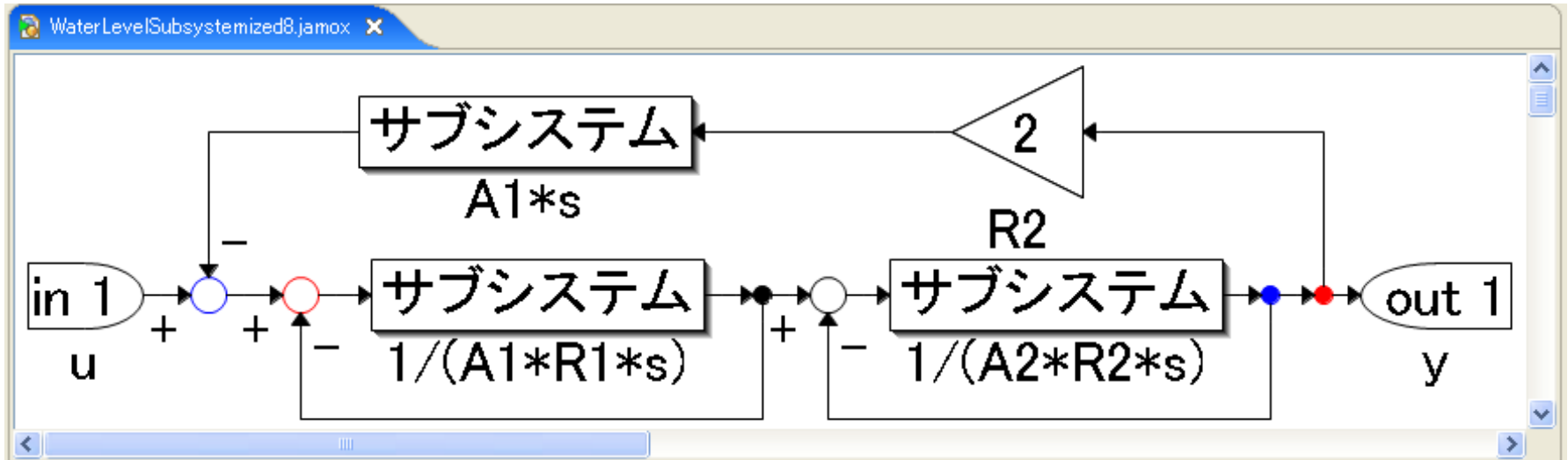
# 液面系のサブシステム化(4/7)



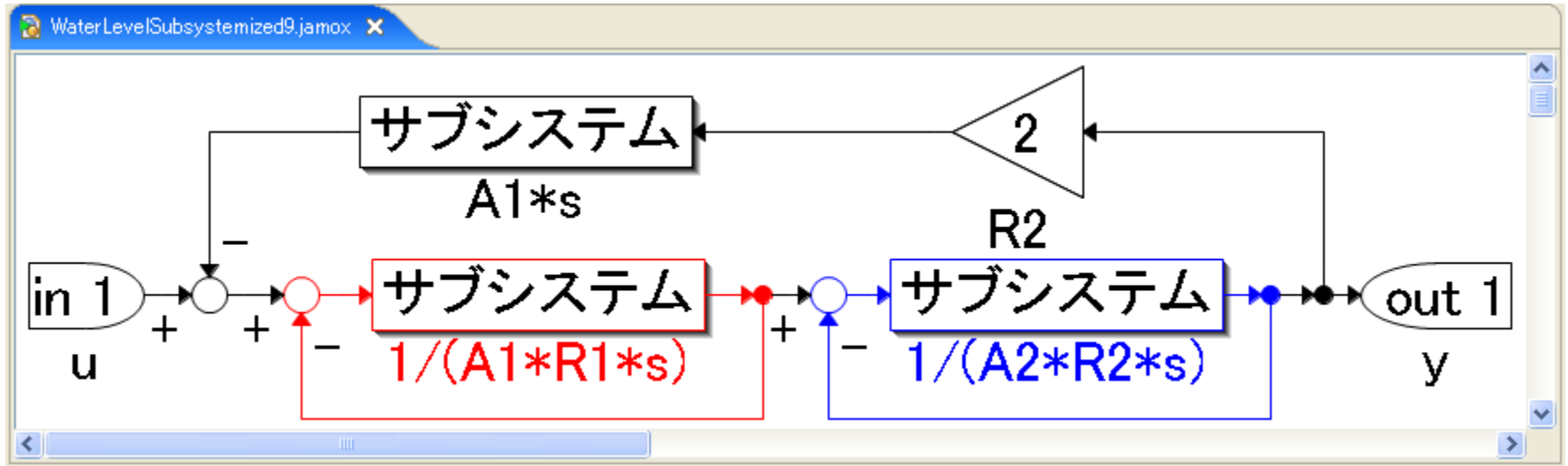
加え合せ点の入れ替え



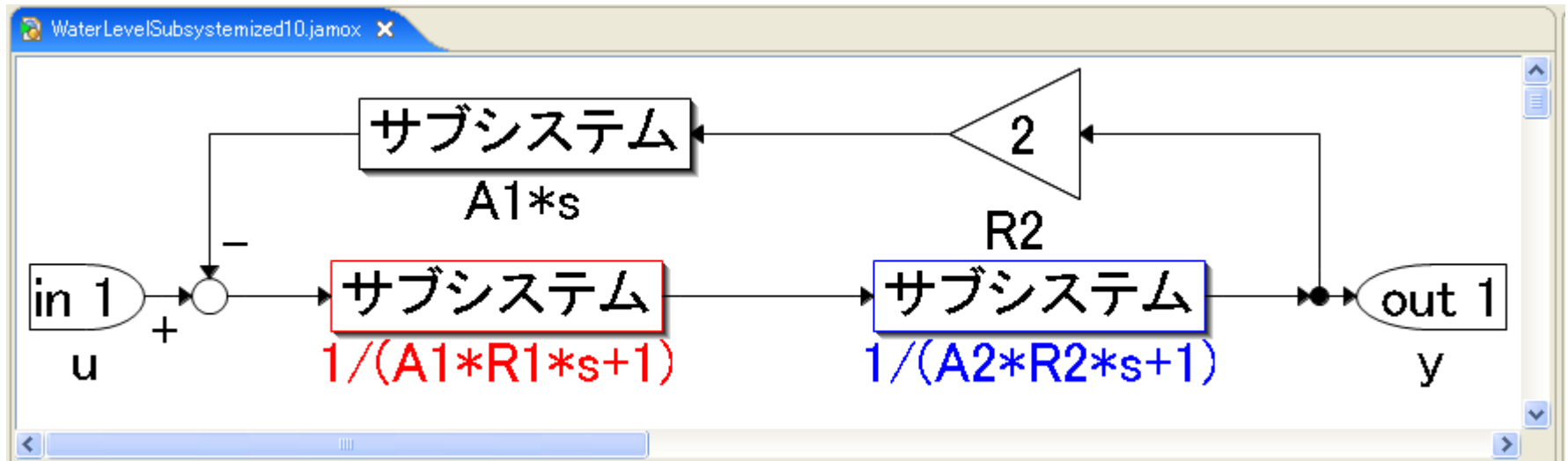
引き出し点の入れ替え



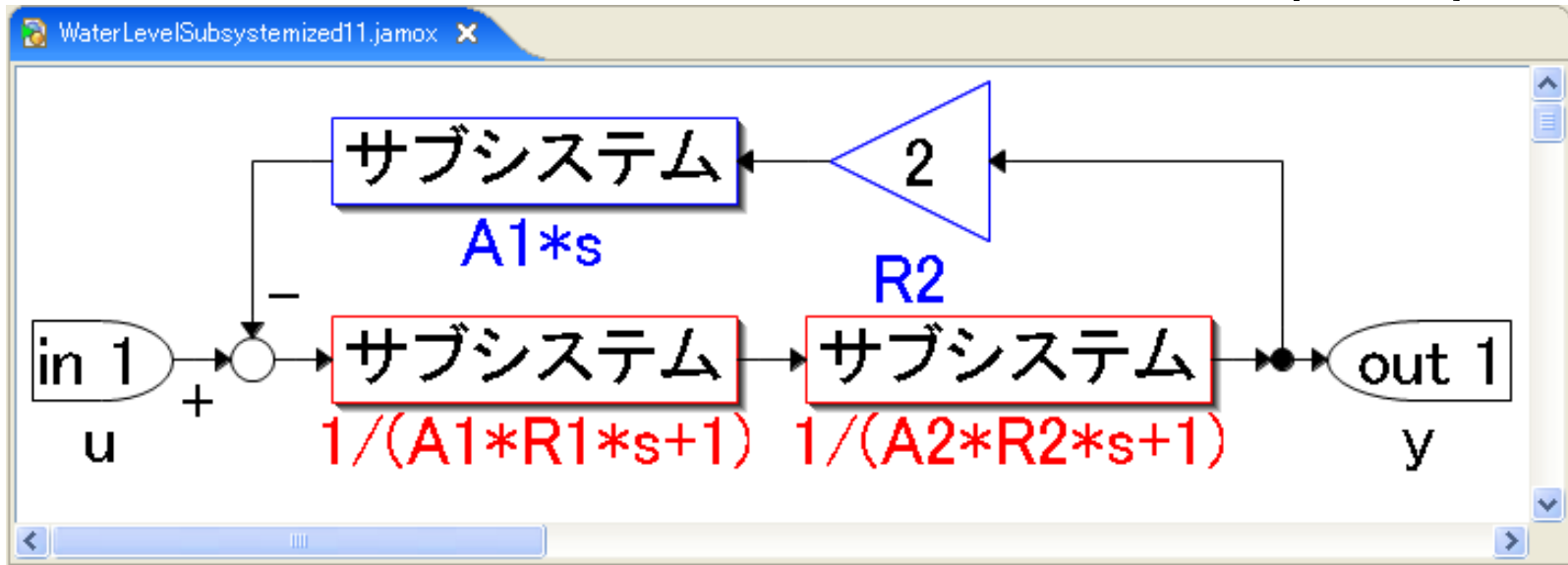
# 液面系のサブシステム化(5/7)



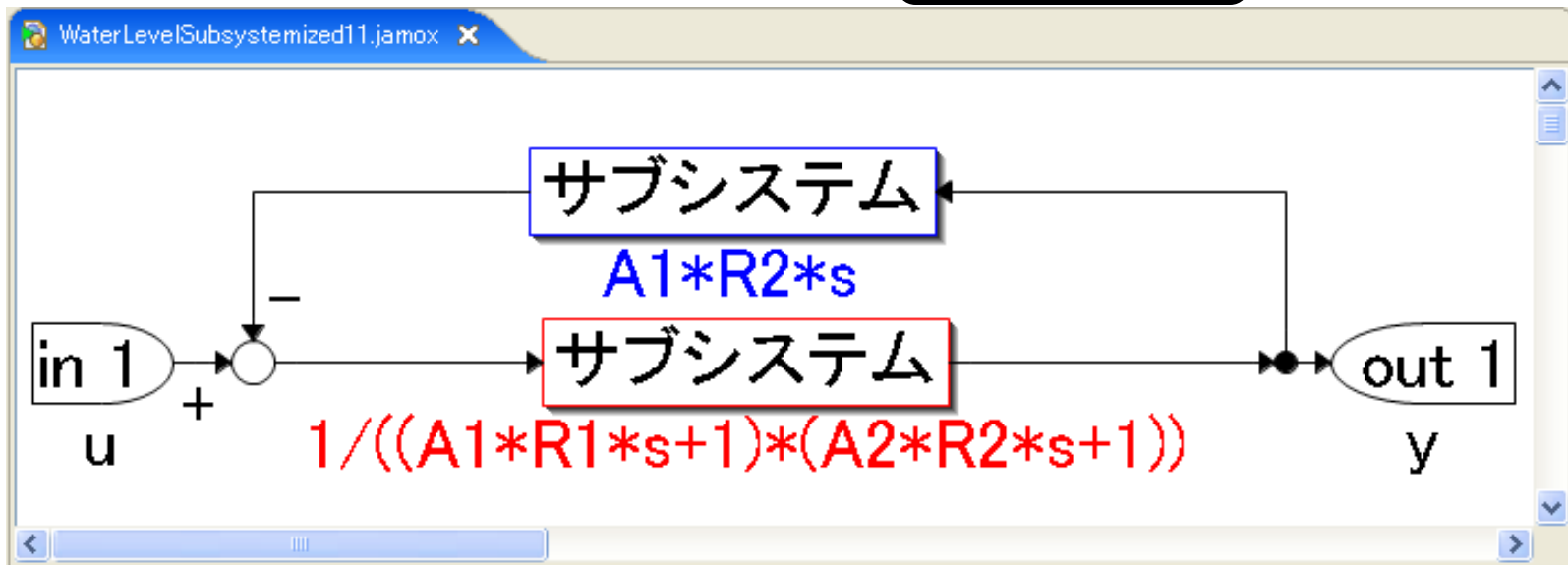
↓ フィードバック結合



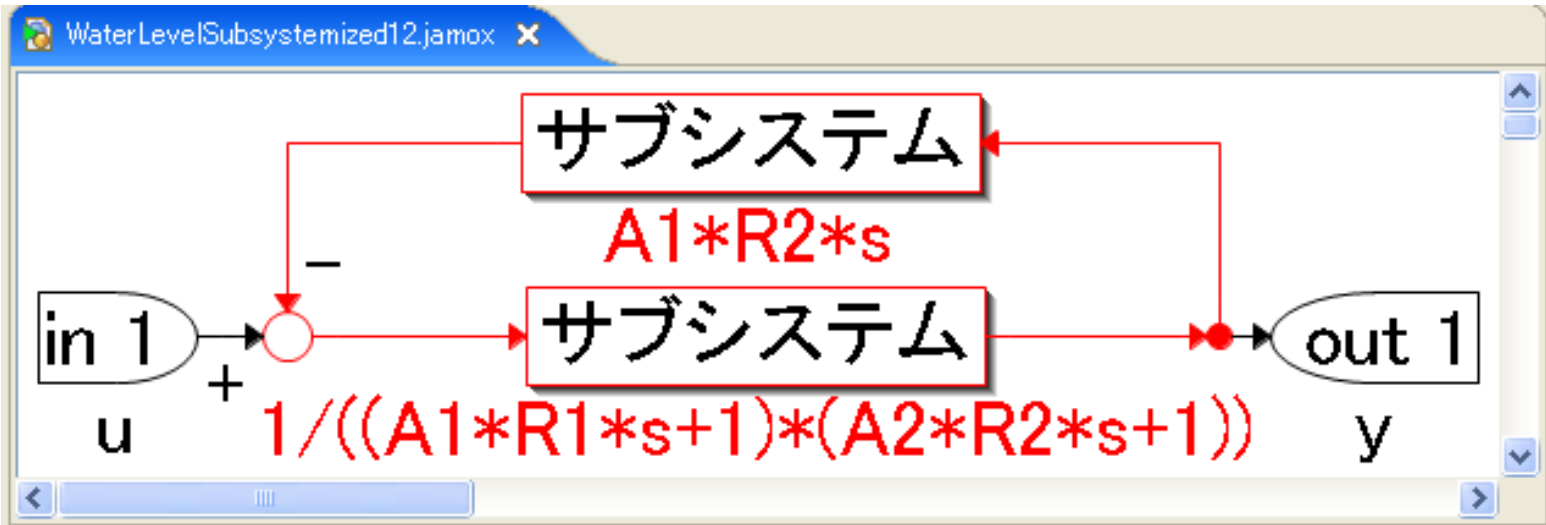
# 液面系のサブシステム化(6/7)



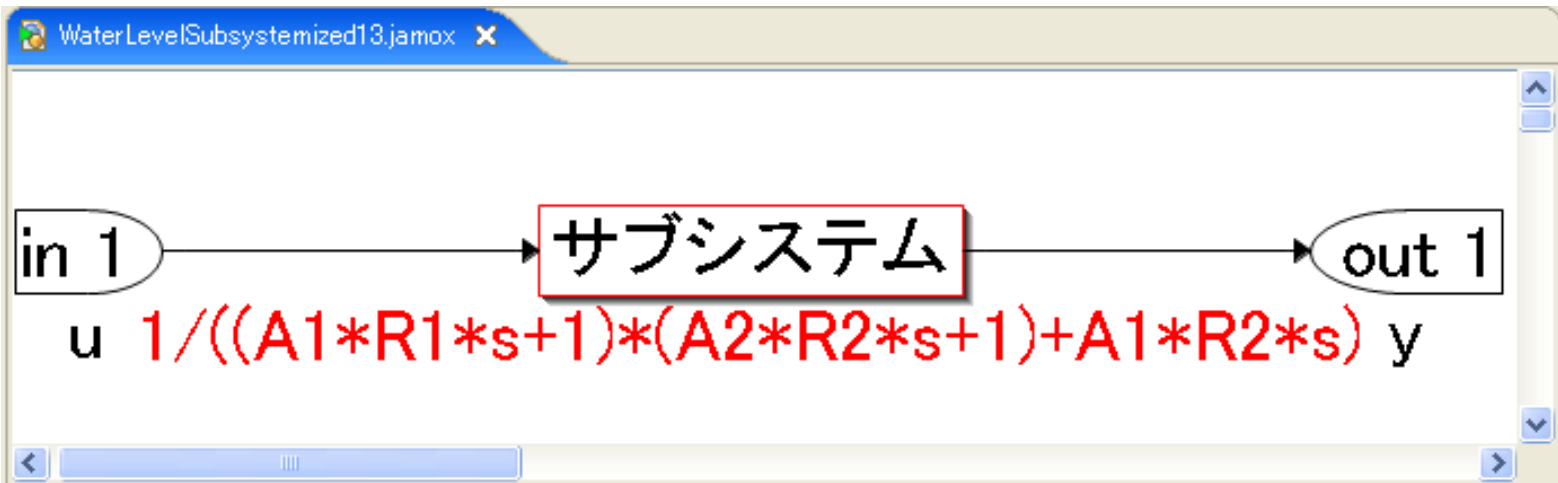
直列結合



# 液面系のサブシステム化(7/7)

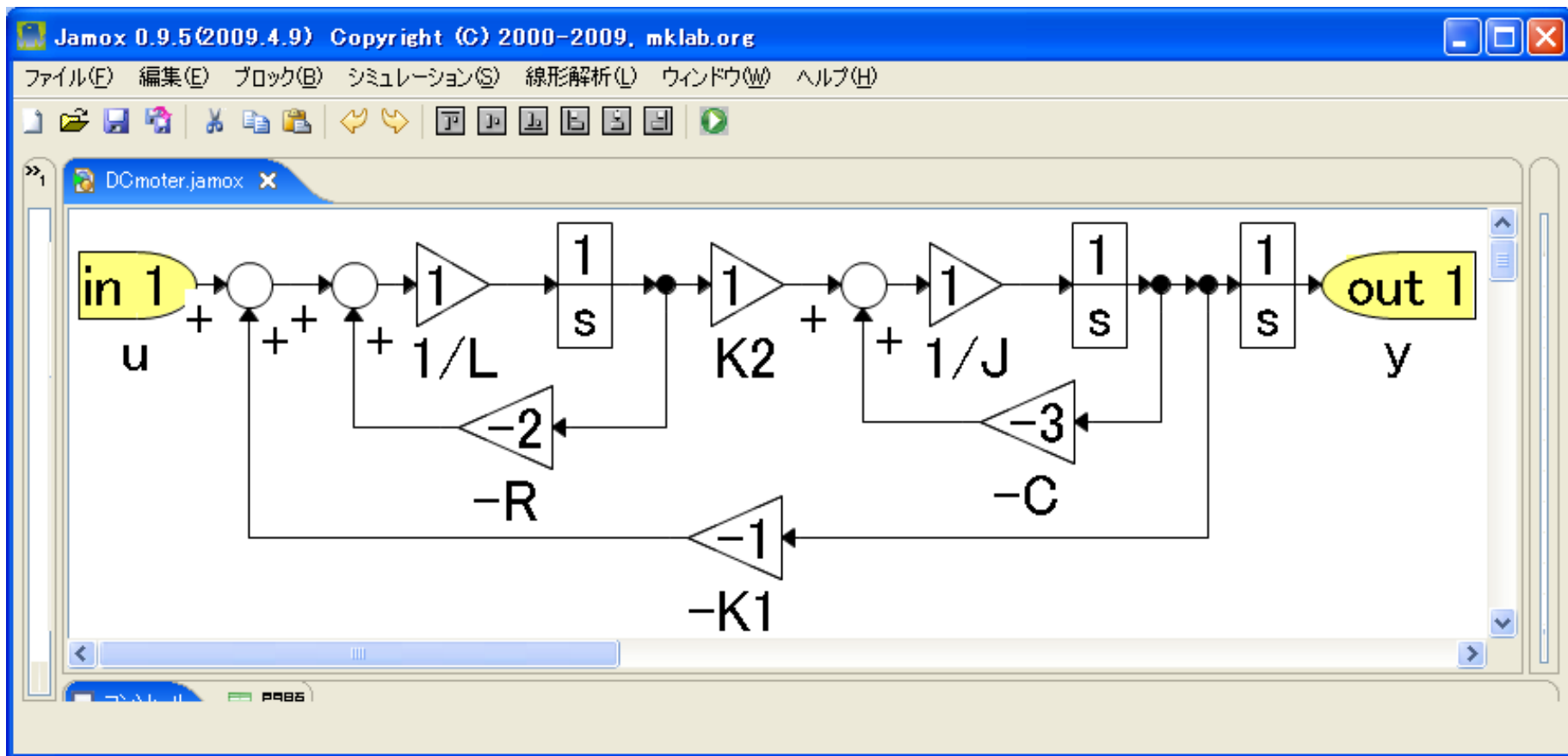


↓ フィードバック結合



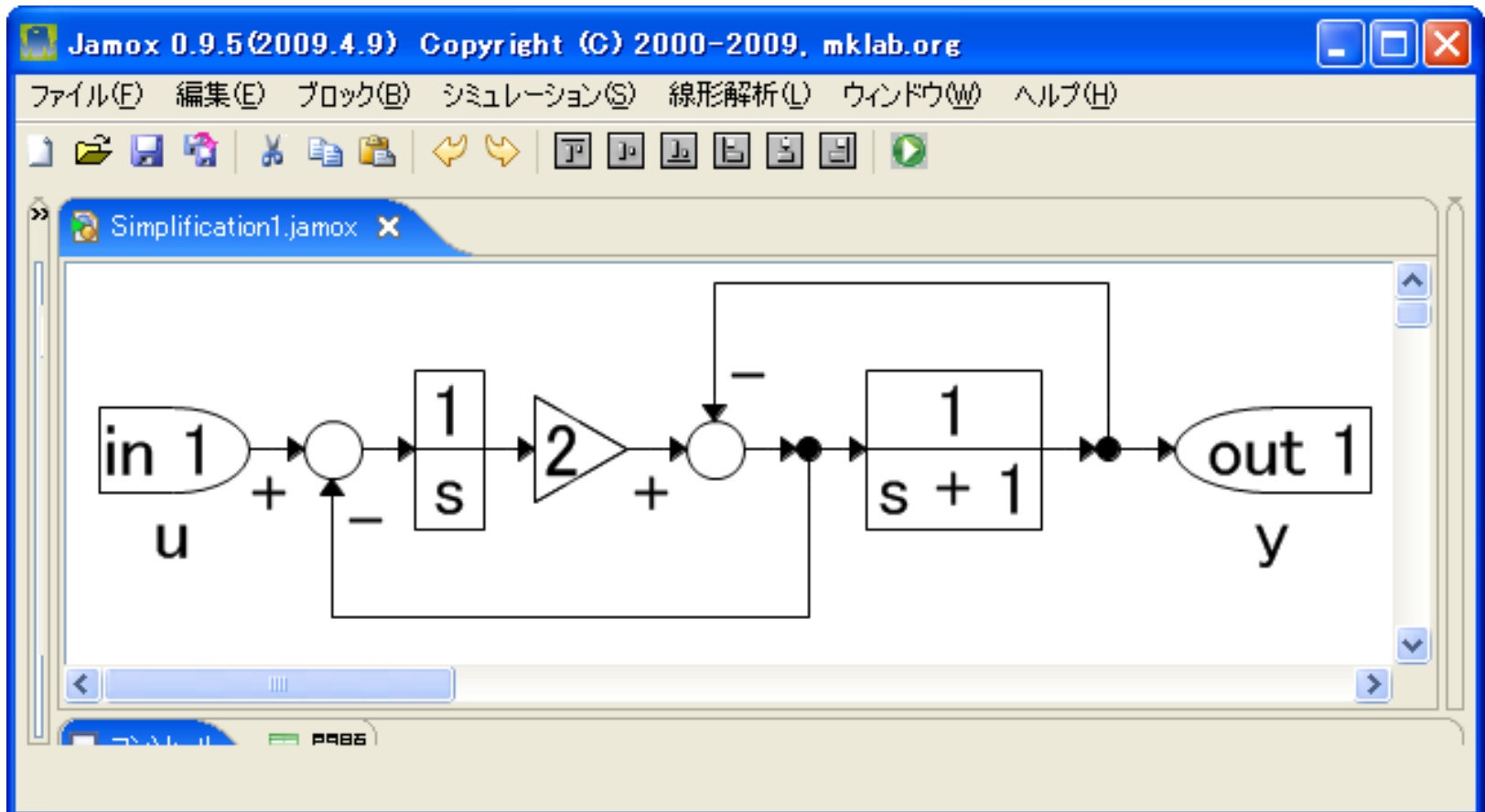
# 演習1: ブロック線図の簡単化(1)

- ブロック線図を簡単化し、 $u$ から $y$ までの伝達関数を求めよ。ただし、変換方法、変換前後のブロック線図と伝達関数を全て示すこと。



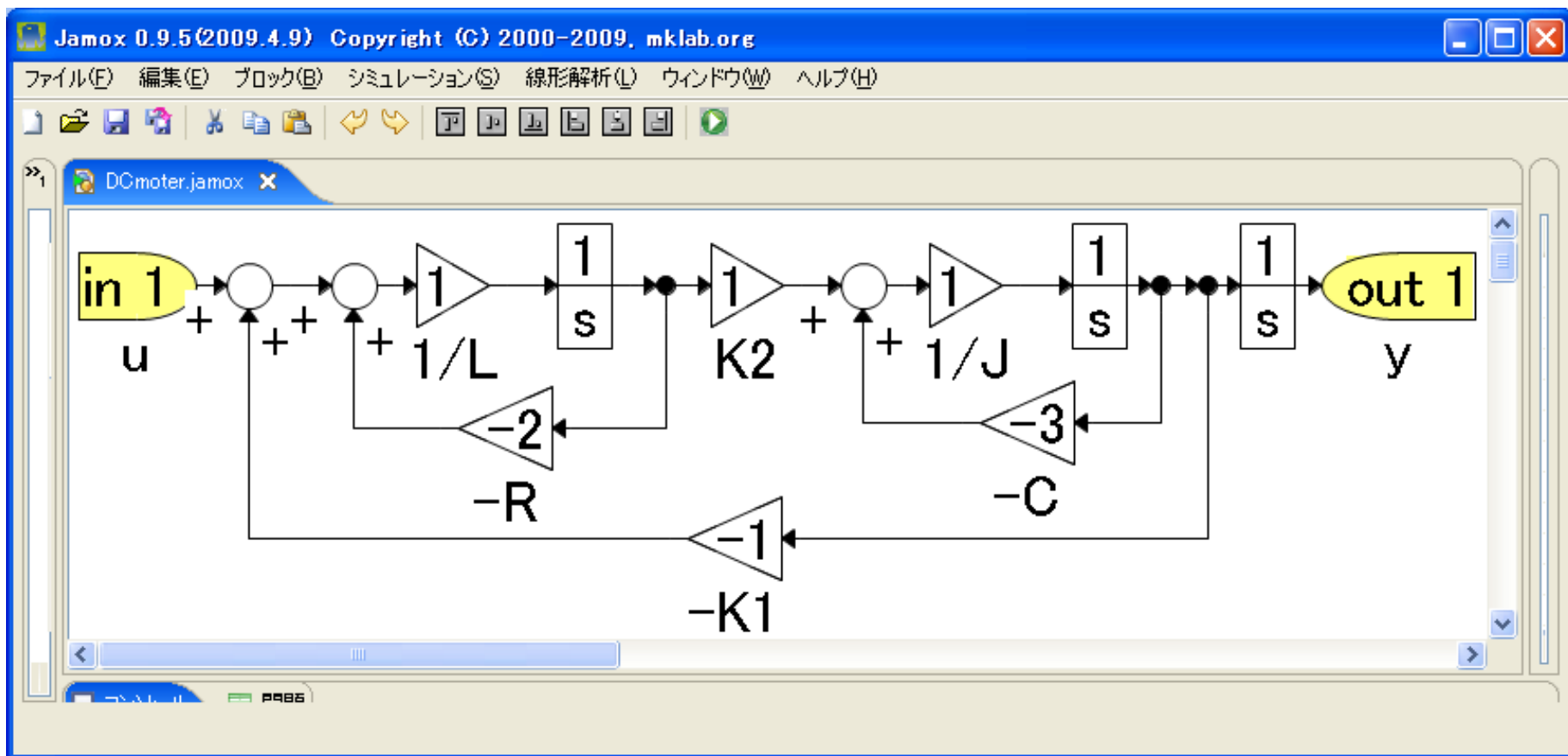
# 演習2: ブロック線図の簡単化(2)

- ブロック線図を簡単化し、 $u$ から $y$ までの伝達関数を求めよ。ただし、変換方法、変換前後のブロック線図と伝達関数を全て示すこと。



# 演習3: ブロック線図のサブシステム化(1)

- ブロック線図をサブシステム化し、 $u$ から $y$ までの伝達関数を求めよ。ただし、サブシステム化の過程を全て示すこと。



# 演習4：ブロック線図のサブシステム化(2)

- ブロック線図をサブシステム化し、 $u$ から $y$ までの伝達関数を求めよ。ただし、サブシステム化の過程を全て示すこと。

