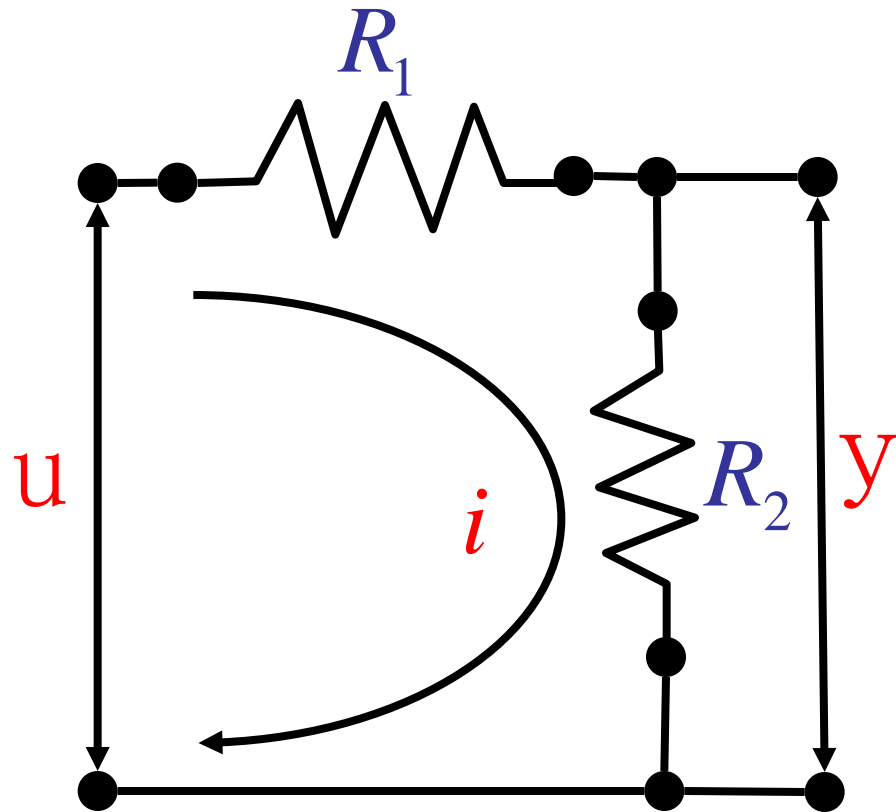


# 第2回

- 微分方程式表現に基づくブロック線図作成
- 伝達関数

# 抵抗回路の方程式表現



関係式:

$$u(t) = i(t) \times (R_1 + R_2)$$

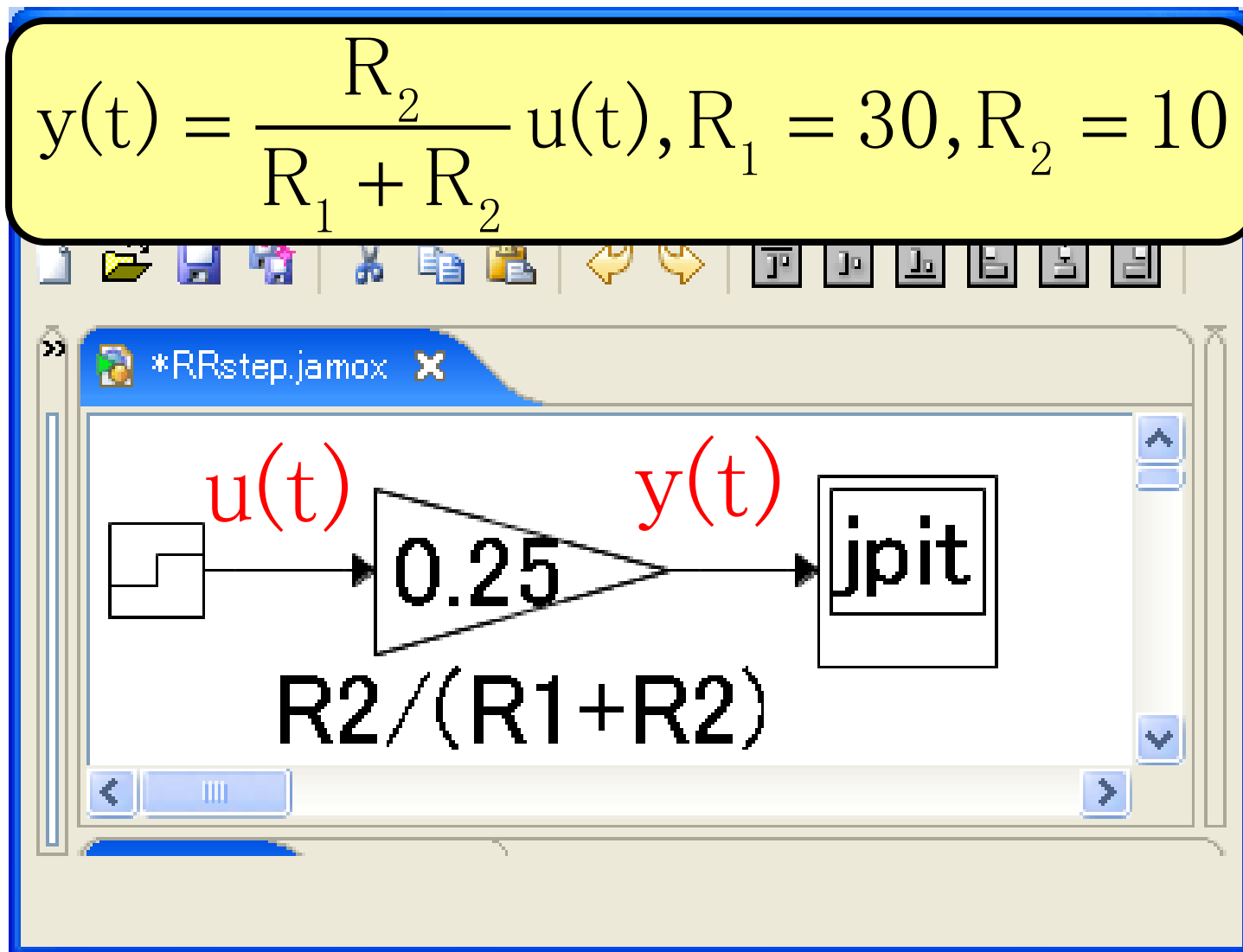
$$y(t) = i(t) \times R_2$$



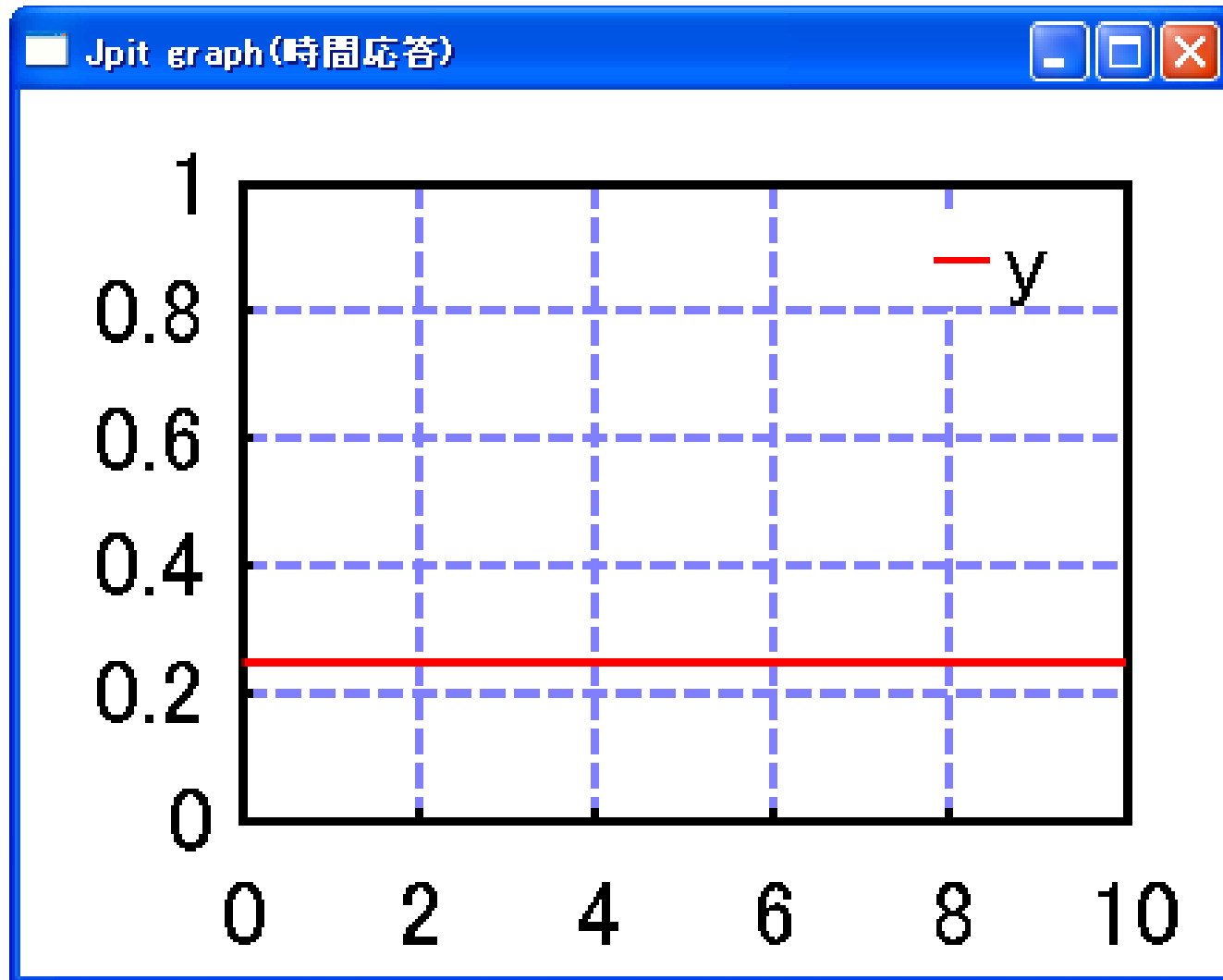
$$y(t) = \frac{R_2}{R_1 + R_2} u(t)$$

# 抵抗回路のブロック線図

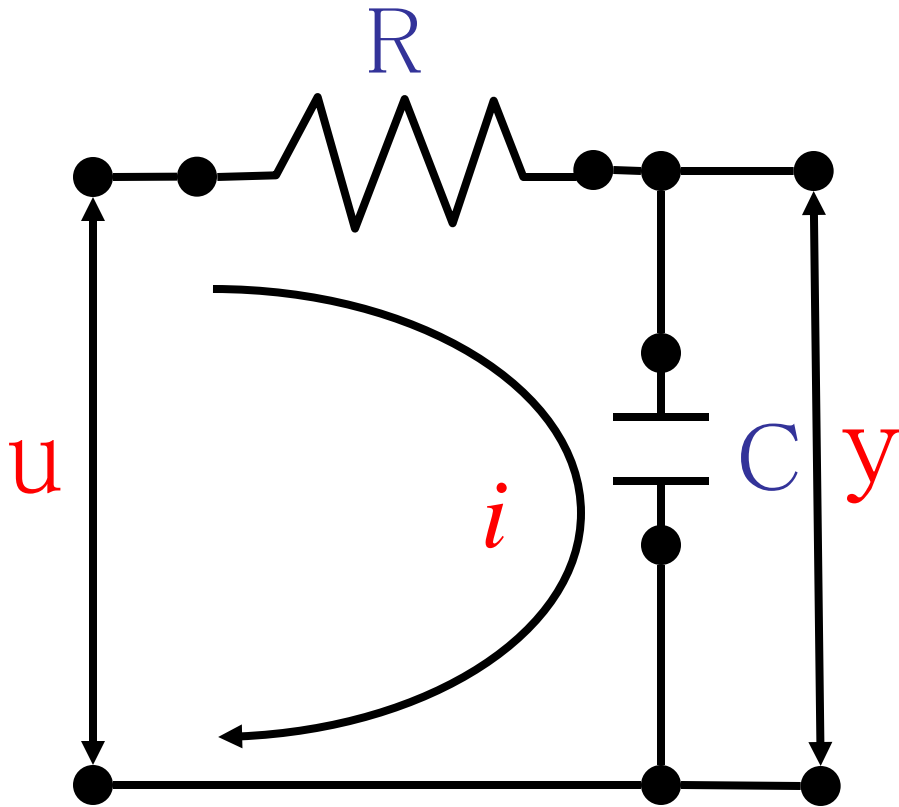
$$y(t) = \frac{R_2}{R_1 + R_2} u(t), R_1 = 30, R_2 = 10$$



# 抵抗回路のステップ応答



# RC回路(微分方程式表現)



關係式:

$$i(t) = C\dot{y}(t)$$

$$u(t) = Ri(t) + y(t)$$



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



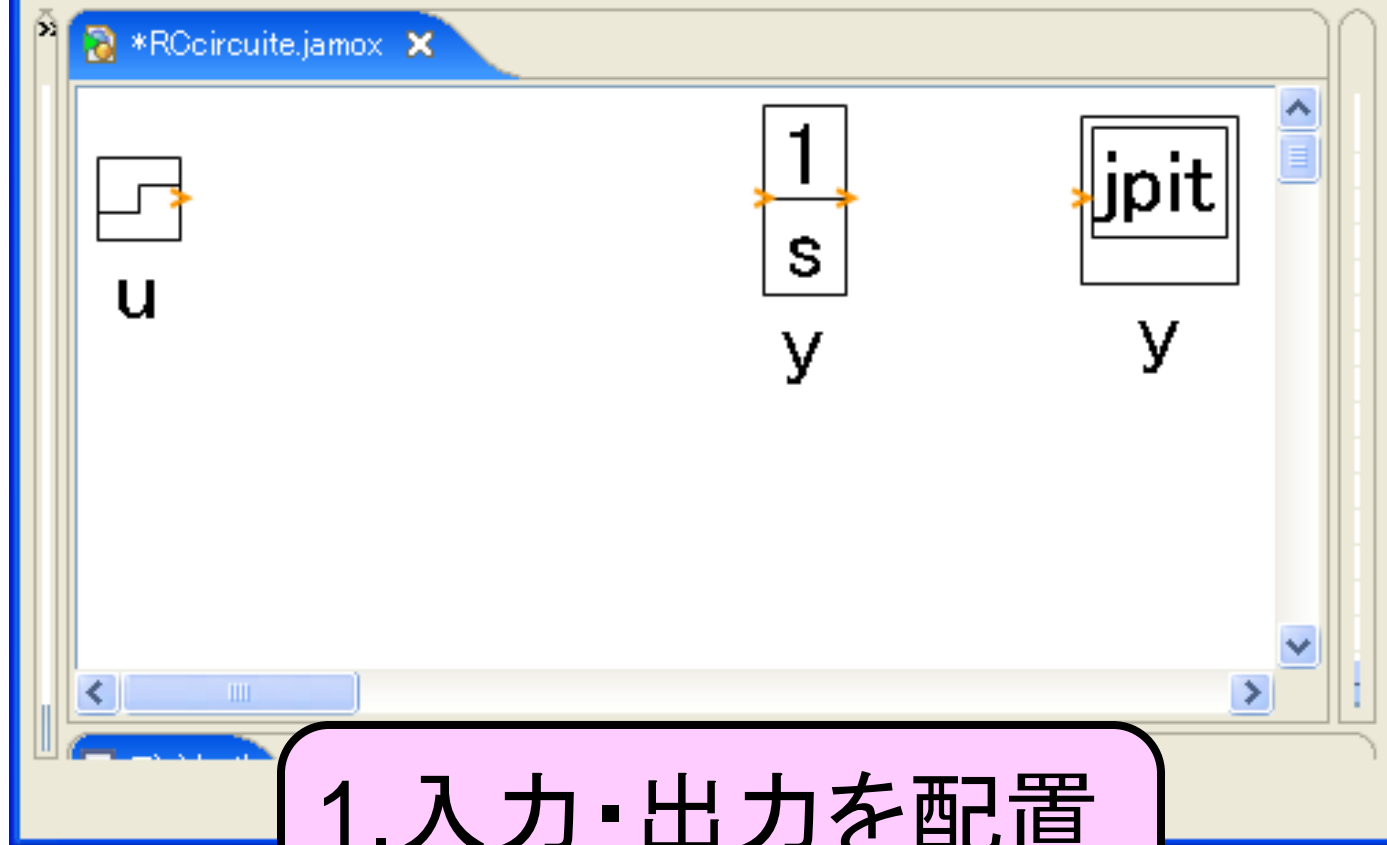
$$\dot{y}(t) = \left(-y(t) + u(t)\right) / RC$$

# 微分方程式に基づくブロック線図の作成

1. 入力・出力を配置  
(変数名を設定)
2. 積分器を配置  
(微分・積分に着目。変数名を設定)
3. 定数を配置  
(変数名を設定)
4. ブロックを接続  
(乗除算を直列結合)
5. ブロックを接続  
(加減算を並列・フィードバック結合)
6. ブロックを整理 (整列、反転、移動)

# RC回路のブロック線図(1/4)

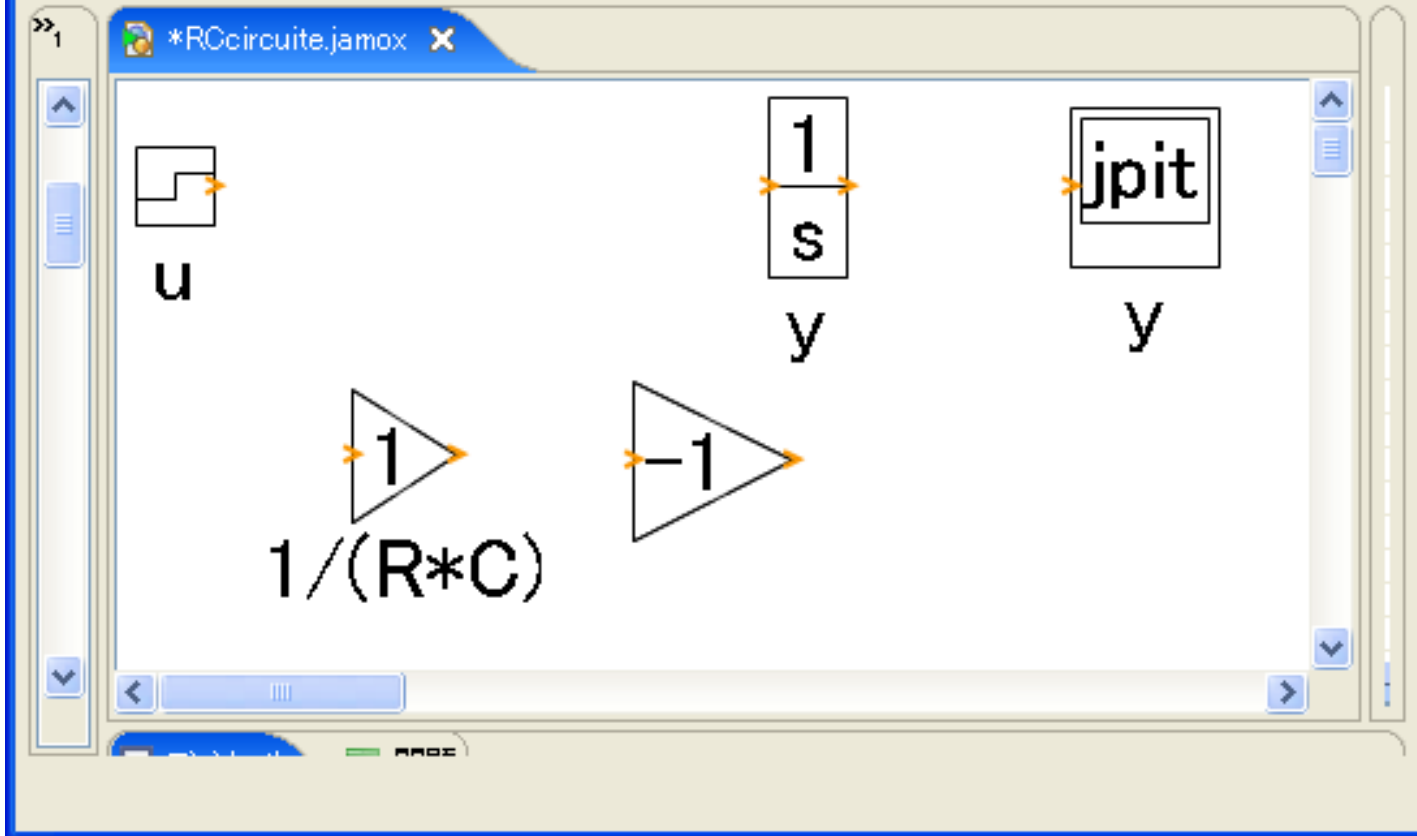
$$\dot{y}(t) = (-y(t) + u(t)) / RC, RC = 2$$



1. 入力・出力を配置
2. 積分器を配置

# RC回路のブロック線図(2/4)

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

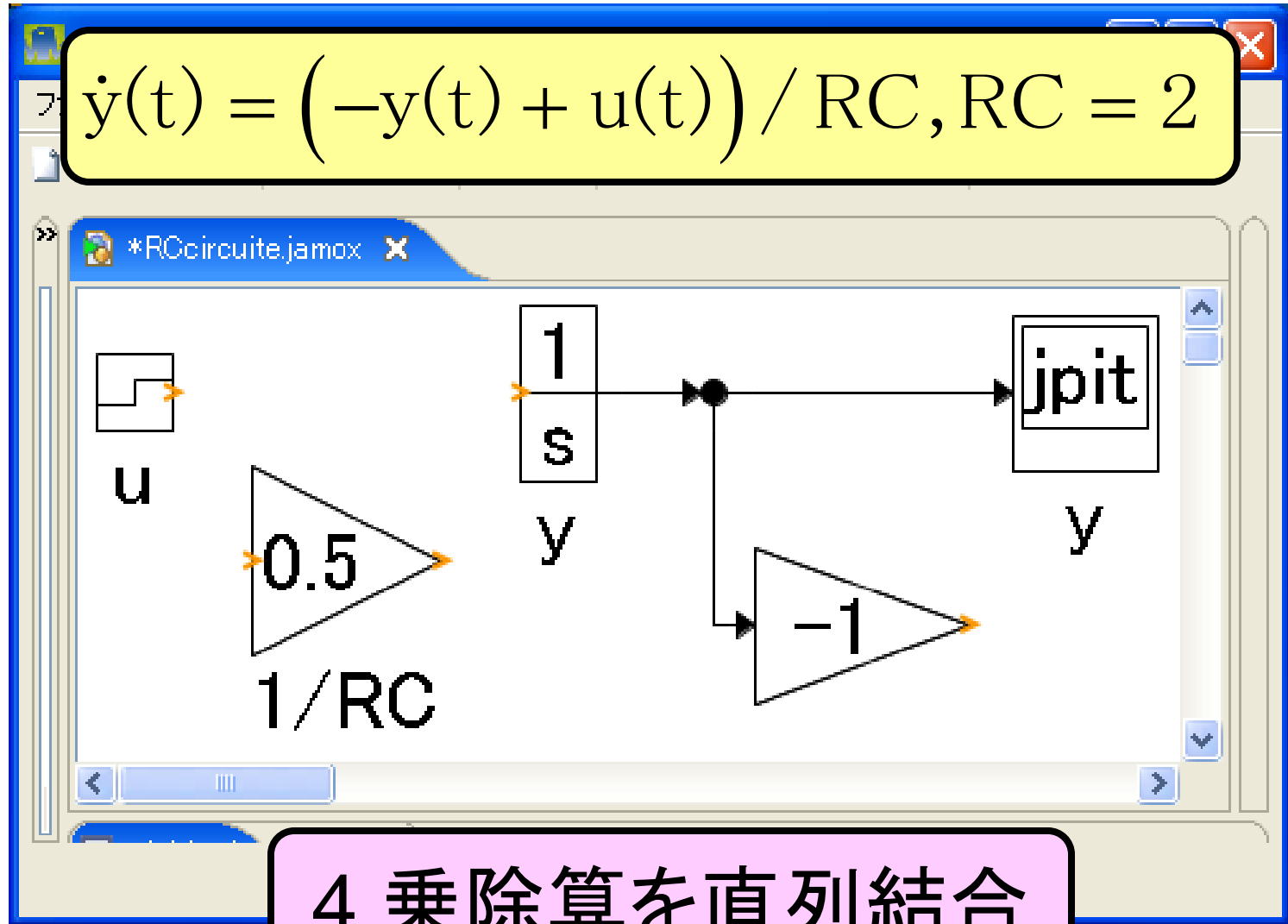


3.定数を配置



# RC回路のブロック線図(3/4)

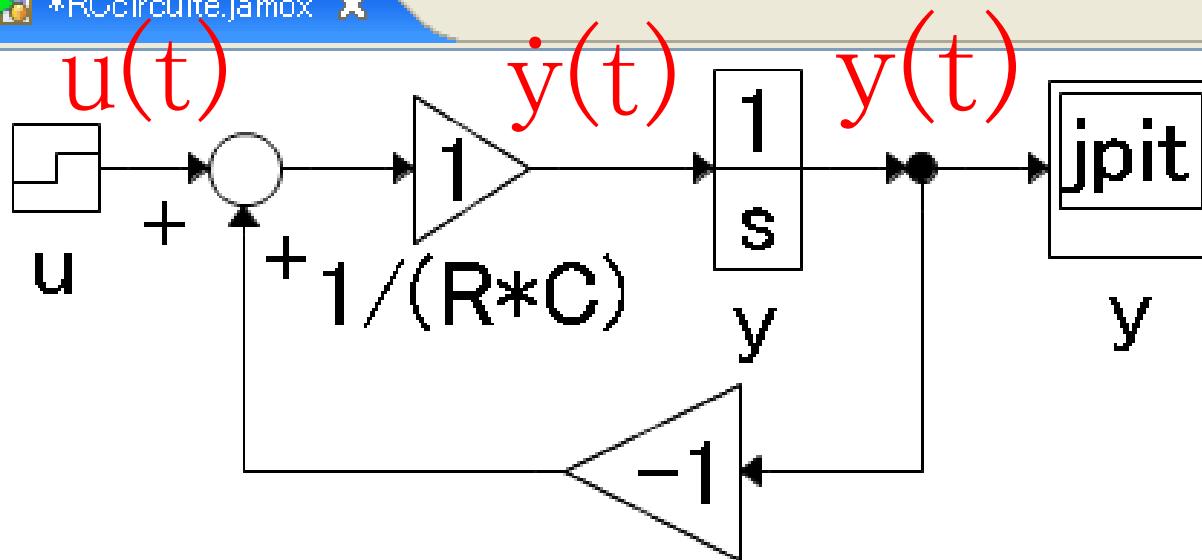
$$\dot{y}(t) = (-y(t) + u(t)) / RC, RC = 2$$



4.乗除算を直列結合

# RC回路のブロック線図(4/4)

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



5. 加減算を並列・フィードバック結合

# RC回路のステップ応答

初期条件:  $y(0) = 0$   
積分器のパラメータ設定

パラメータ設定:

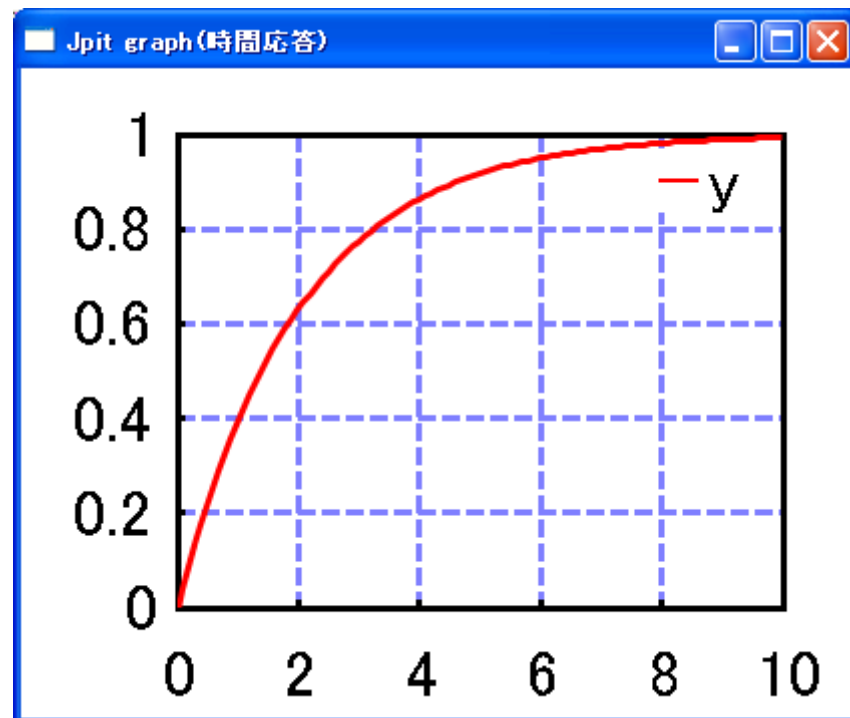
基本 システム 図 ポート

初期状態(initialState) [0]'

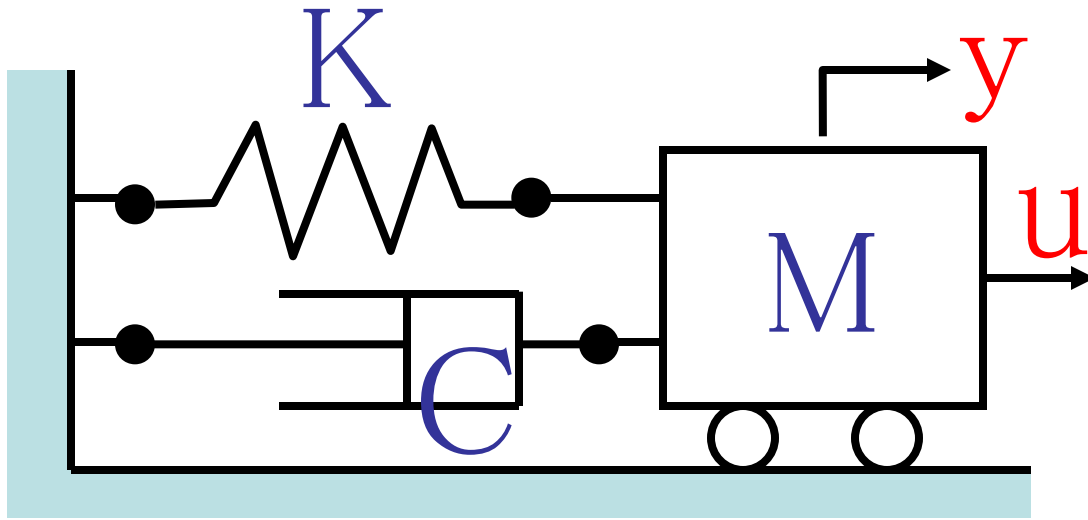
タグ(tag) x

状態番号(stateNumber) 1

了解 キャンセル 適用



# 質量・ばね・ダンパ系（微分方程式表現）



関係式：

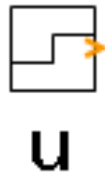
$$M\ddot{y}(t) = -C\dot{y}(t) - Ky(t) + u(t)$$



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

# 質量・ばね・ダンパ系のブロック線図(1/4)

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



1. 入力・出力を配置
2. 積分器を配置

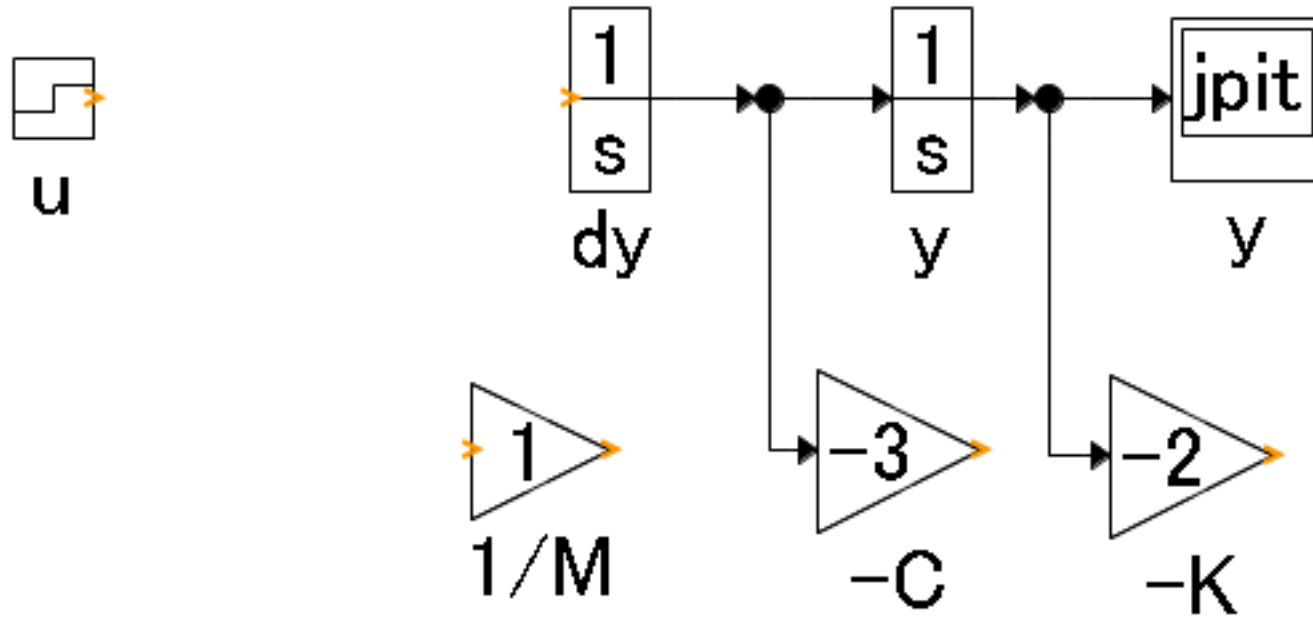
# 質量・ばね・ダンパ系のブロック線図(2/4)

The screenshot shows the Jamox software interface with the following elements:

- Window title: Jamox 0.9.5(2009.4.9) Copyright (C) 2000-2009, mklab.org
- Equation in a yellow box: 
$$\ddot{y}(t) = \left( -C\dot{y}(t) - Ky(t) + u(t) \right) / M$$
- Block diagram components:
  - Input block:  $u$
  - Integrator blocks:  $\frac{1}{s}$  (labeled  $dy$  and  $y$ )
  - Gain blocks:  $\frac{1}{M}$ ,  $-K$ , and  $-C$
  - Output block:  $y$  (labeled  $jpit$ )
- Bottom text in a pink box: 3.定数を配置

# 質量・ばね・ダンパ系のブロック線図(3/4)

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



4.乗除算を直列結合

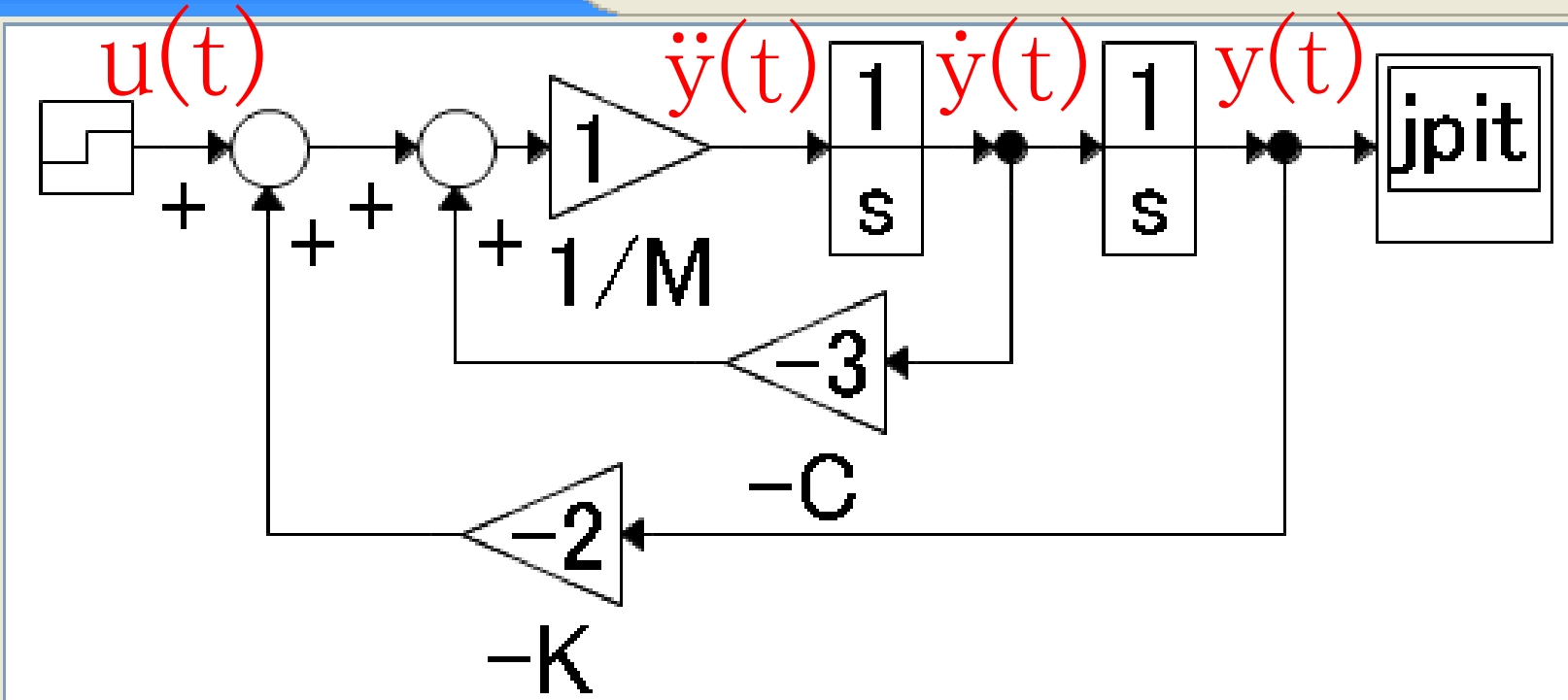
# 質量・ばね・ダンパ系のブロック線図(4/4)

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

ファイル

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

MassSpringDamper.jamox



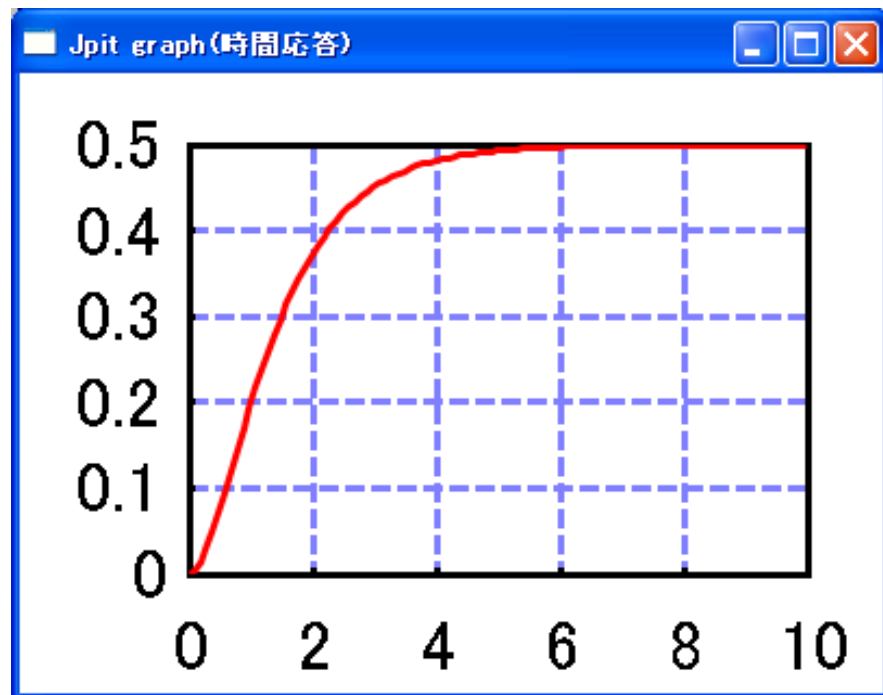
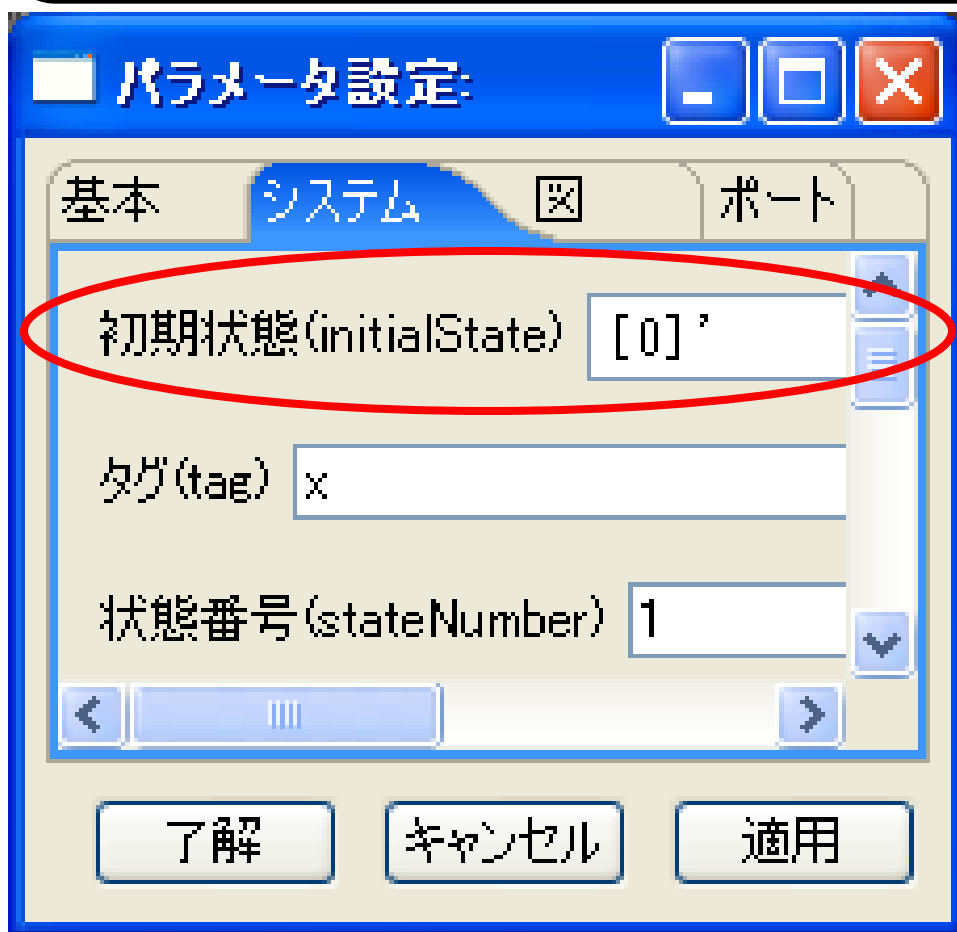
The block diagram illustrates the implementation of the differential equation  $\ddot{y}(t) = (-C\dot{y}(t) - Ky(t) + u(t)) / M$ . It starts with an input  $u(t)$  entering a summing junction. A feedback signal from the output  $y(t)$  is multiplied by  $-K$  and added to the sum. Another feedback signal from the first derivative  $\dot{y}(t)$  is multiplied by  $-C$  and added to the sum. The result is then multiplied by  $1/M$ . This signal is integrated (represented by a  $1/s$  block) to produce  $\dot{y}(t)$ , which is then integrated again (another  $1/s$  block) to produce the final output  $y(t)$ . The output  $y(t)$  is also fed into a 'jpit' block. The diagram uses red text to label the signals  $u(t)$ ,  $\dot{y}(t)$ , and  $y(t)$ .

## 5. 加減算を並列・フィードバック結合

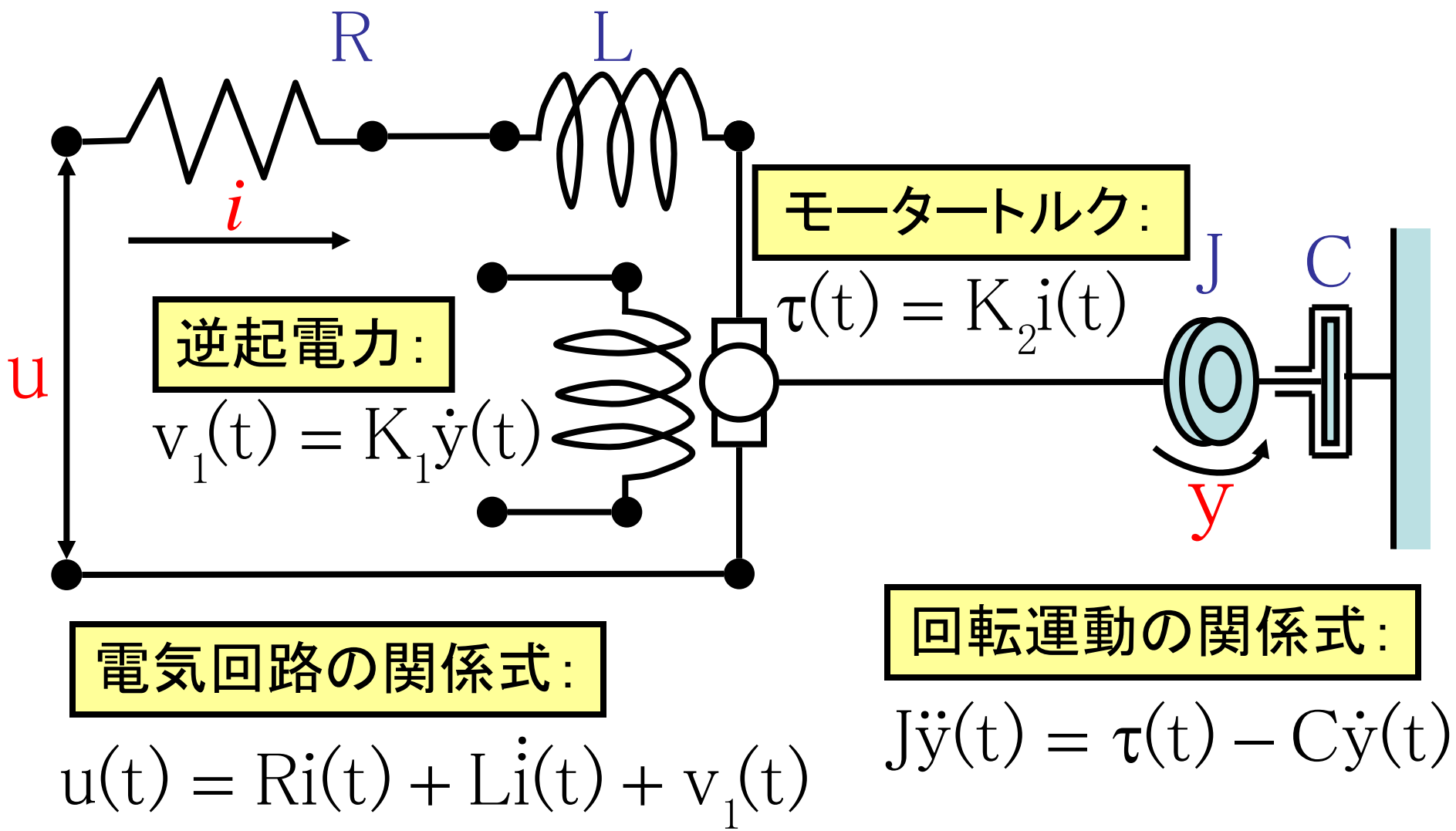


# 質量・ばね・ダンパ系のステップ応答

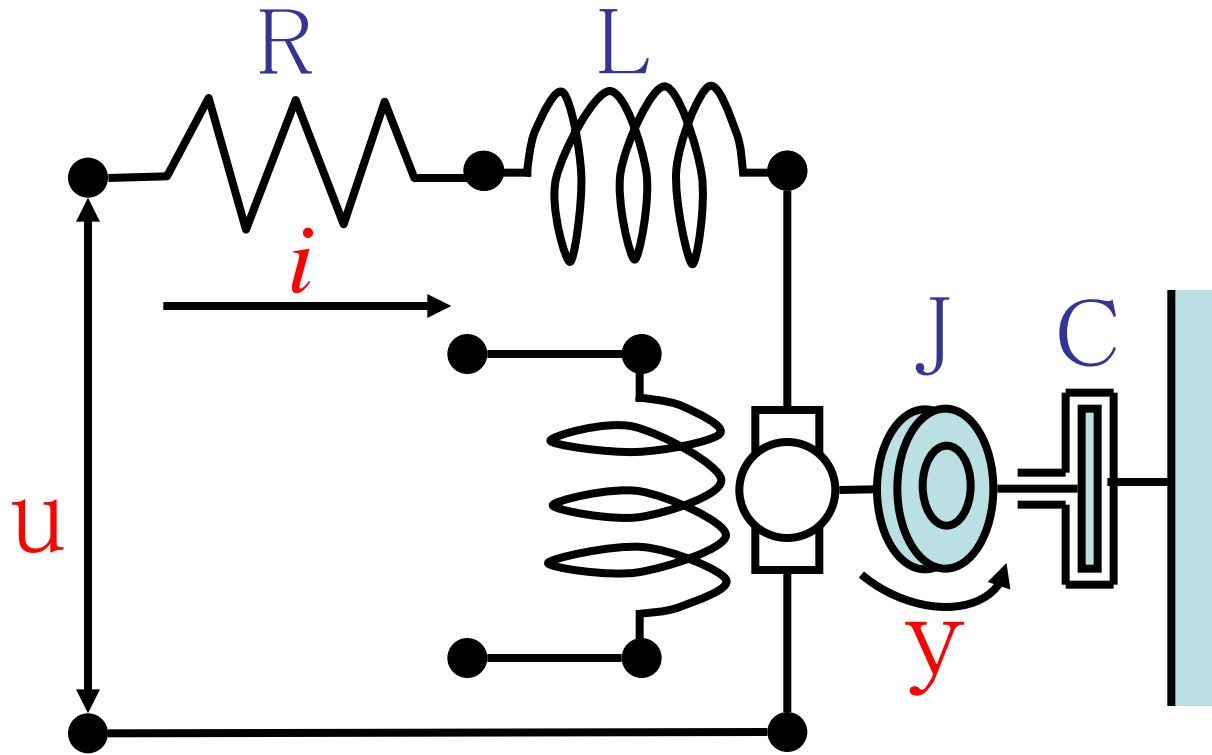
初期条件:  $y(0) = 0, \dot{y}(0) = 0$   
積分器のパラメータ設定



# DCサーボモータ



# DCサーボモータの微分方程式表現



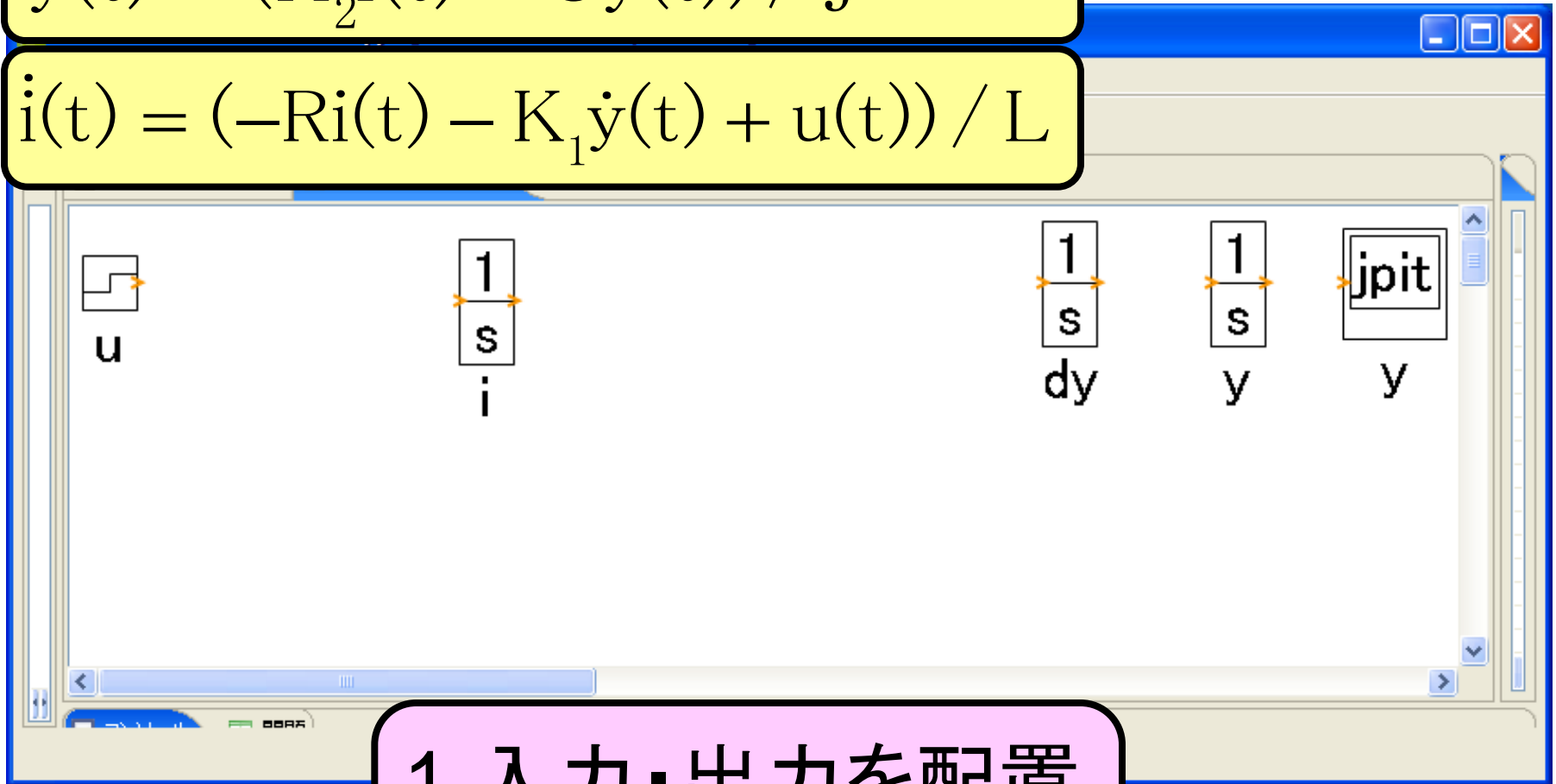
$$\ddot{y}(t) = (K_2 i(t) - C \dot{y}(t)) / J$$

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

# DCサーボモータのブロック線図(1/4)

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

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

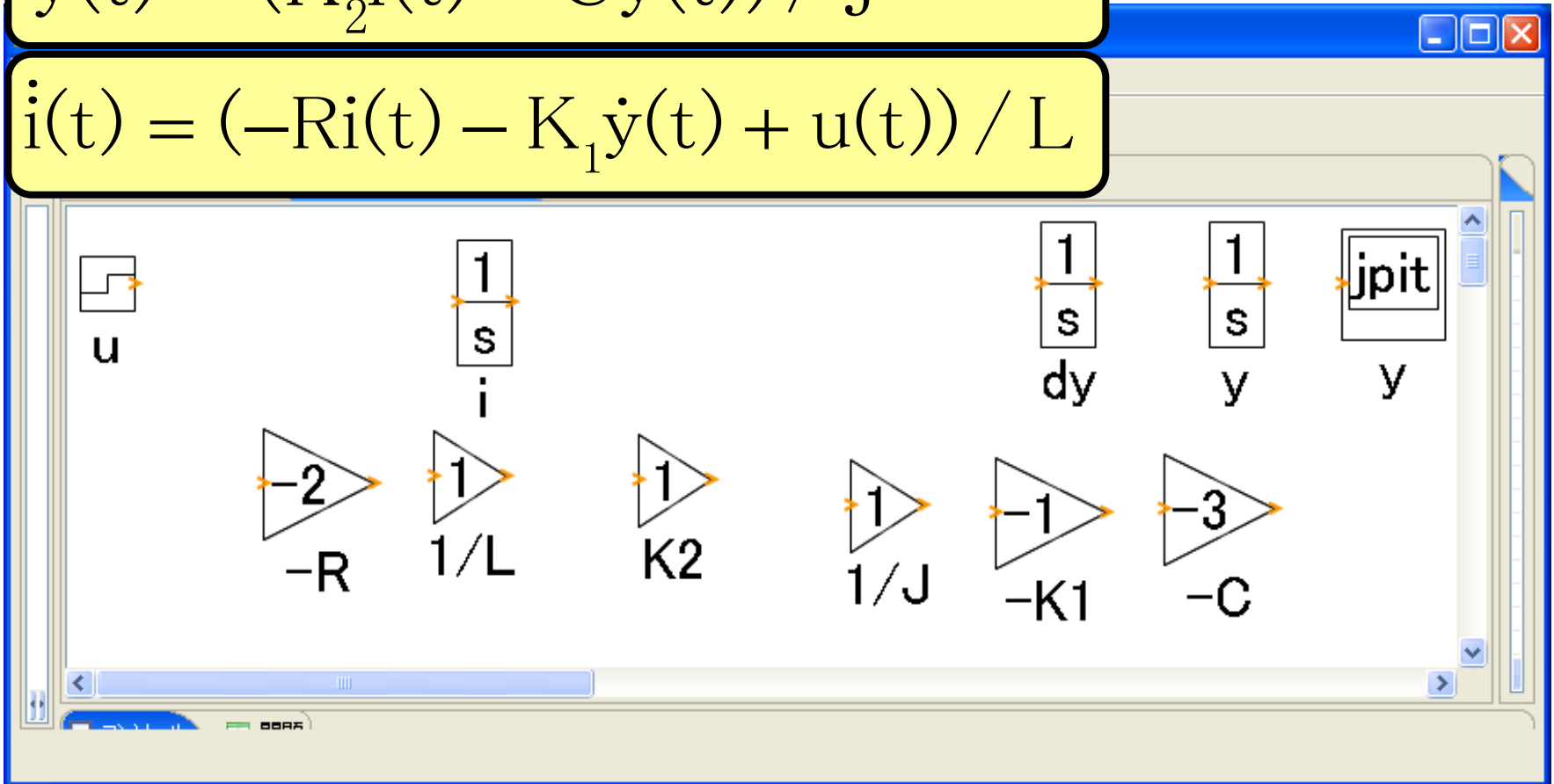


1. 入力・出力を配置
2. 積分器を配置

# DCサーボモータのブロック線図(2/4)

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

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

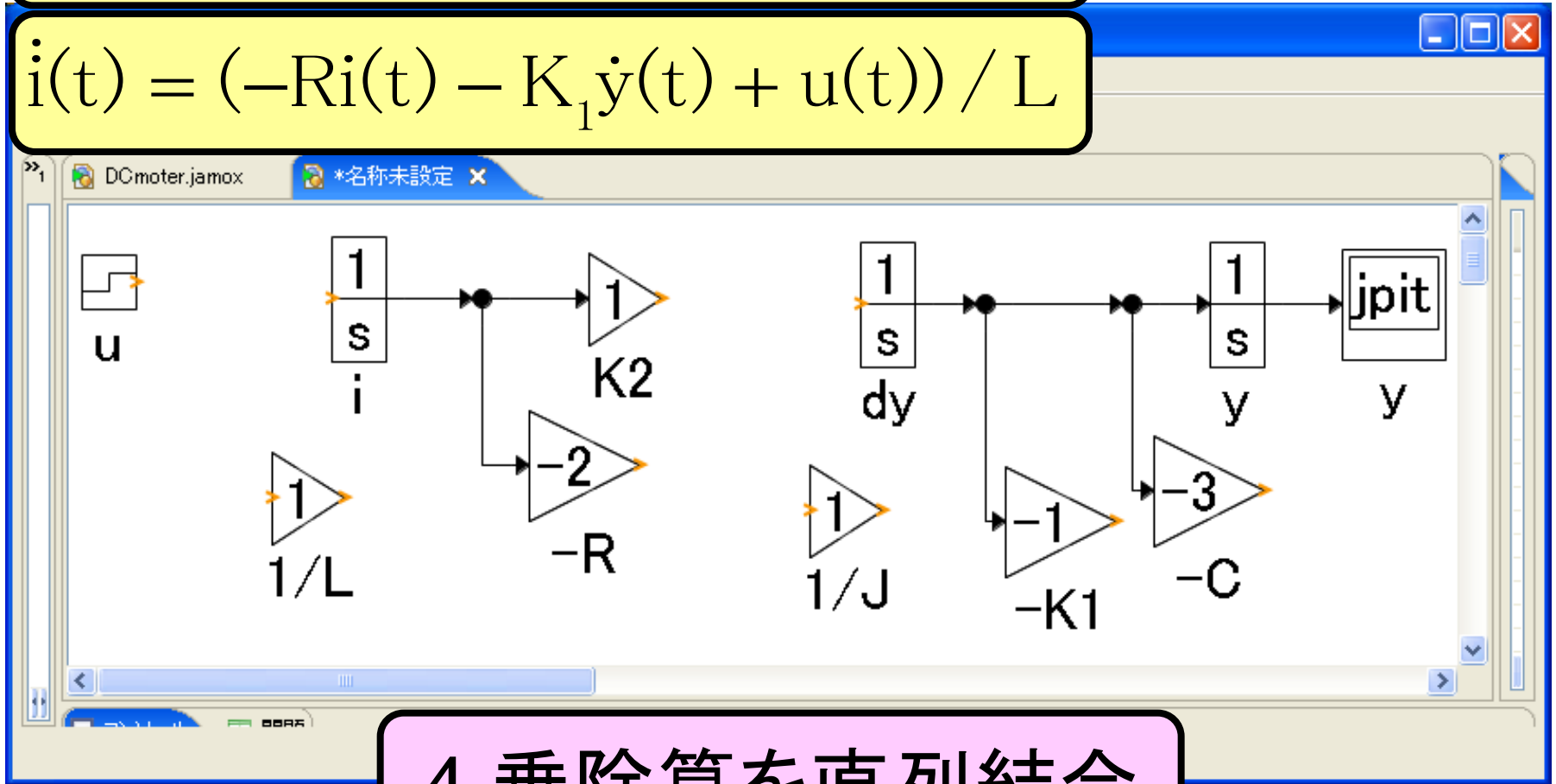


3.定数を配置

# DCサーボモータのブロック線図(3/4)

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

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

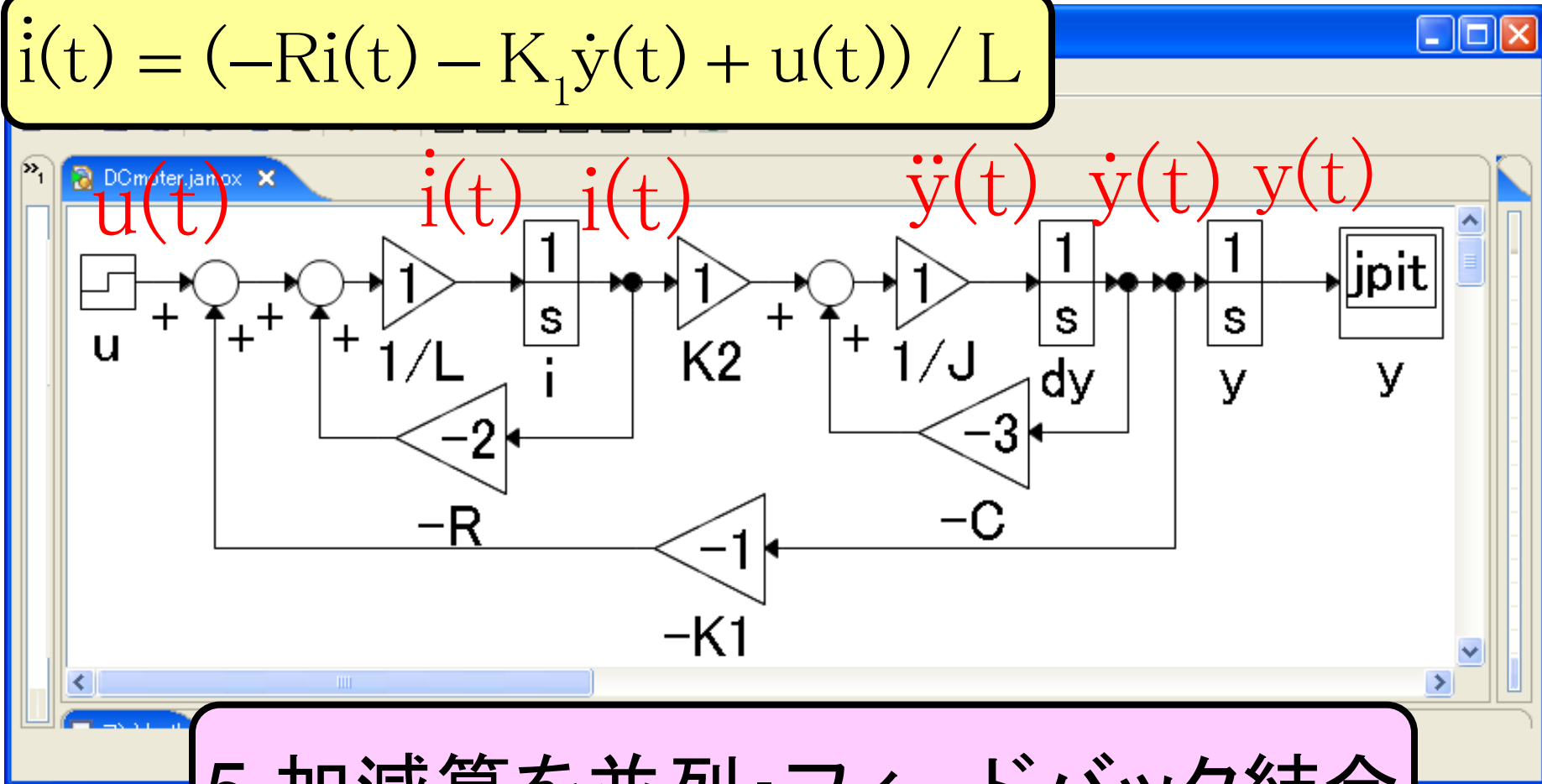


4.乗除算を直列結合

# DCサーボモータのブロック線図(4/4)

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

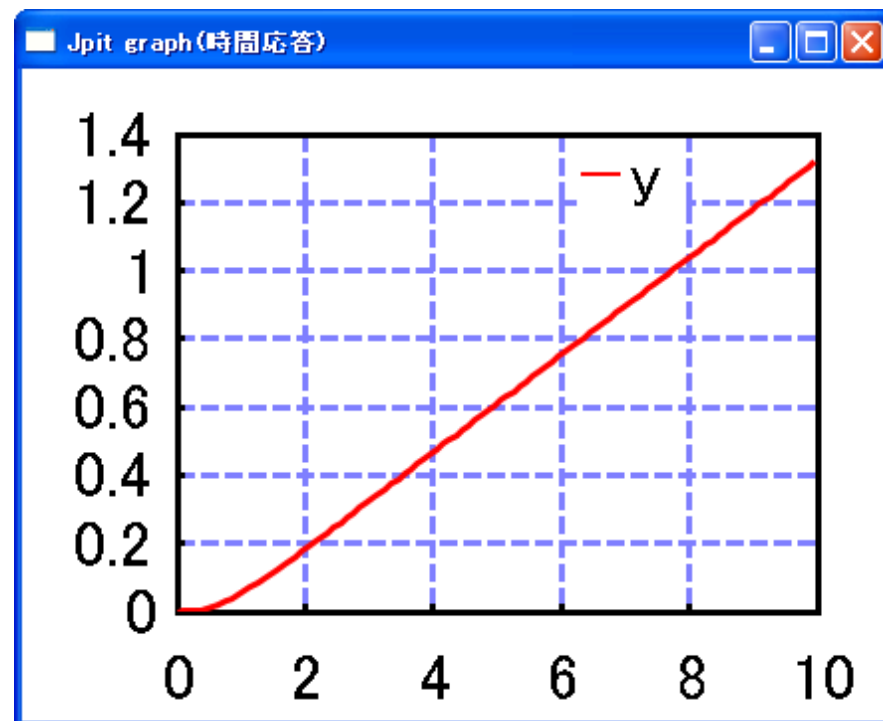
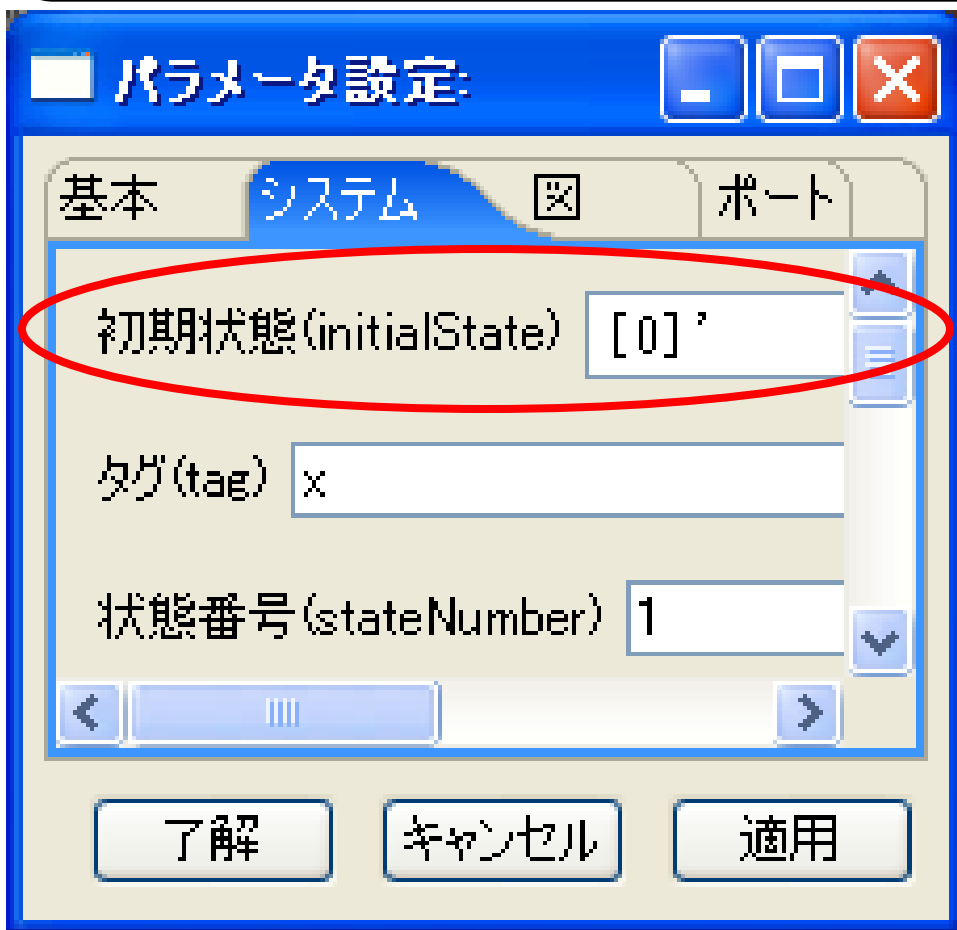
$$\dot{i}(t) = (-R i(t) - K_1 \dot{y}(t) + u(t)) / L$$



5. 加減算を並列・フィードバック結合

# DCサーボモータのステップ応答

初期条件： $y(0) = 0, \dot{y}(0) = 0, i(0) = 0$   
積分器のパラメータ設定





# ラプラス変換

ラプラス変換:

$$F(s) = \mathcal{L}[f(t)] := \int_0^{\infty} f(t)e^{-st} dt$$

時間微分のラプラス変換:

$$\mathcal{L}\left[\frac{d^n f(t)}{dt^n}\right] = s^n F(s) \quad \left( f(0) = \dot{f}(0) = \dots = f^{(n-1)}(0) = 0 \right. \\ \left. \text{のとき} \right)$$

時間積分のラプラス変換:

$$\mathcal{L}\left[\int_0^t f(\tau) d\tau\right] = \frac{1}{s} F(s)$$

# ラプラス変換表(1)

$f(t)$	$F(s) = \mathcal{L}[f(t)]$
$\delta(t)$	1
$u_s(t)(= 1)$	$\frac{1}{s}$
$t$	$\frac{1}{s^2}$
$\sin \omega t$	$\frac{\omega}{s^2 + \omega^2}$
$\cos \omega t$	$\frac{s}{s^2 + \omega^2}$

# ラプラス変換表(2)

$f(t)$	$F(s) = \mathcal{L}[f(t)]$
$e^{-at}$	$\frac{1}{s+a}$
$e^{-at} \sin \omega t$	$\frac{\omega}{(s+a)^2 + \omega^2}$
$e^{-at} \cos \omega t$	$\frac{s+a}{(s+a)^2 + \omega^2}$

# 伝達関数

$$\text{伝達関数} := \frac{\text{出力のラプラス変換}}{\text{入力 of ラプラス変換}}$$

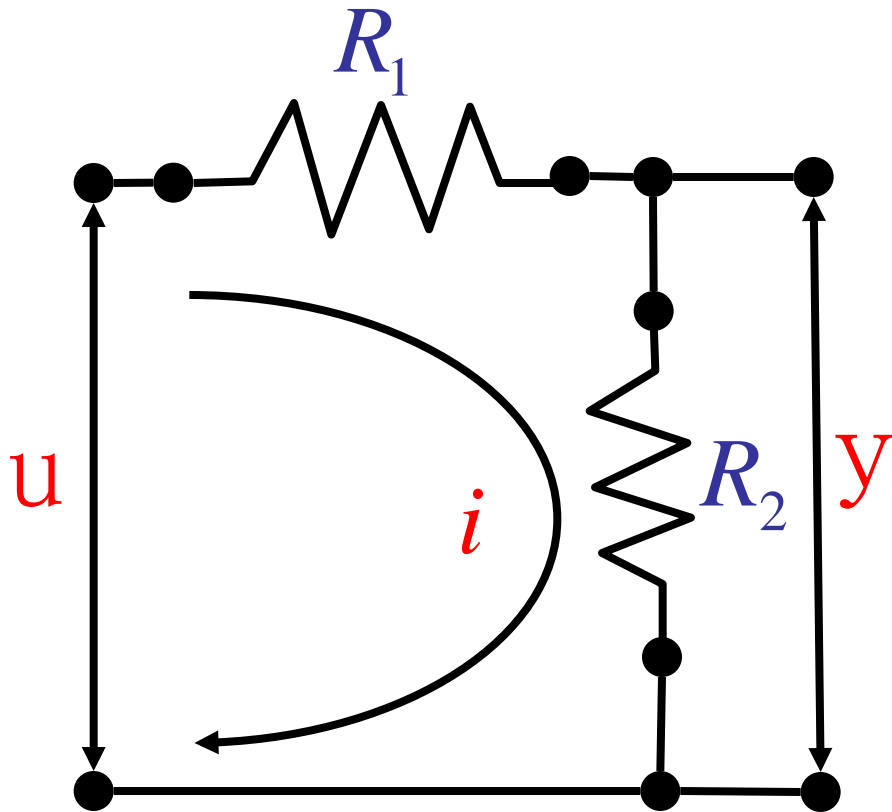
「すべての初期値を0とする」

$$G(s) := \frac{Y(s)}{U(s)} \quad (\text{入力: } u(t), \text{出力: } y(t))$$



$$Y(s) = G(s)U(s)$$

# 抵抗回路の伝達関数



関係式:

$$u(t) = i(t) \times (R_1 + R_2)$$

$$y(t) = i(t) \times R_2$$

⇩ ラプラス変換

$$Y(s) = I(s) \times (R_1 + R_2)$$

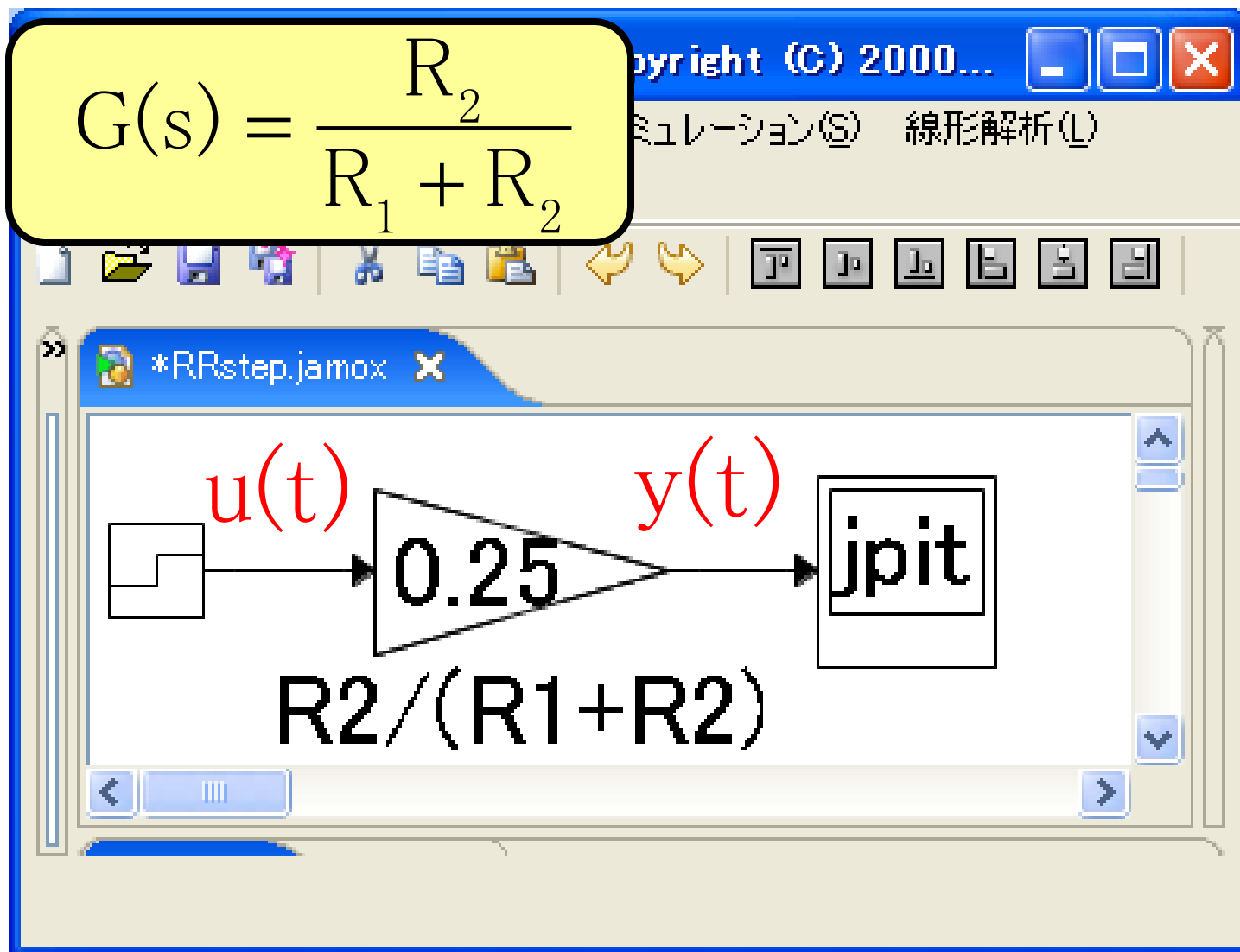
$$Y(s) = I(s) \times R_2$$

⇩

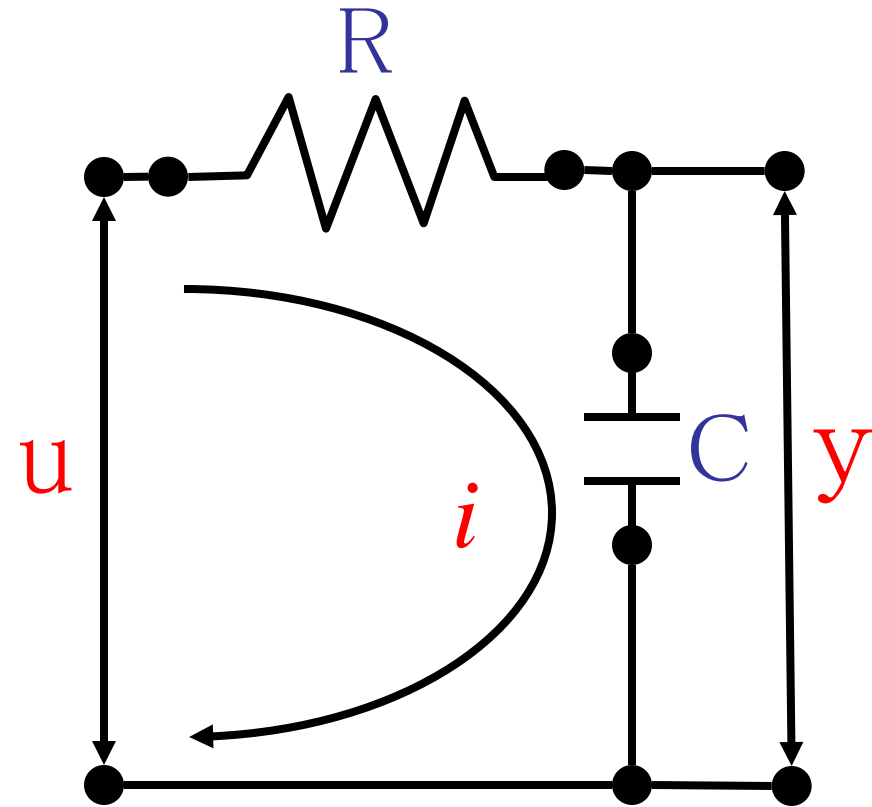
$$G(s) := \frac{Y(s)}{U(s)} = \frac{R_2}{R_1 + R_2}$$

# 抵抗回路のブロック線図

$$G(s) = \frac{R_2}{R_1 + R_2}$$



# RC回路の伝達関数



関係式:

$$i(t) = C\dot{y}(t)$$

$$u(t) = Ri(t) + y(t)$$

ラプラス変換

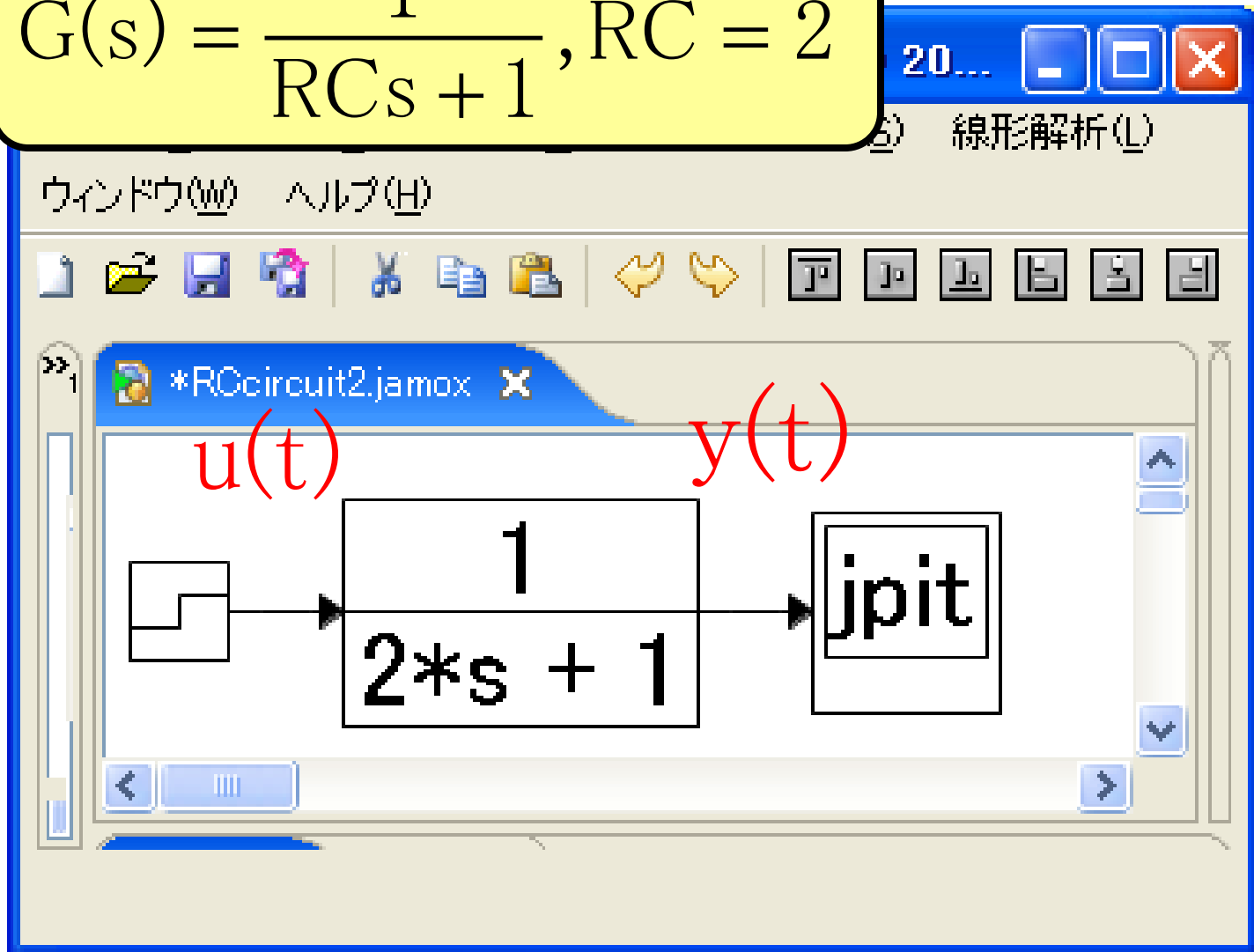
$$I(s) = CsY(s)$$

$$U(s) = RI(s) + Y(s)$$

$$G(s) := \frac{Y(s)}{U(s)} = \frac{1}{RCs + 1}$$

# RC回路のブロック線図

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





# 伝達関数ブロックのパラメータ設定

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

連続時間システム

- 積分器
- 伝達関数表現
- 零点・極表現
- 状態空間表現

初期条件:  $y(0) = 0$

基本 システム ポート

分子係数 (numerator) [1]

分母係数 (denominator) [2 1]

初期状態 (initialState) [0]

タグ (tag) 1

了解 キャンセル 適用

# 質量・ばね・ダンパ系の伝達関数

関係式:

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

⇩ ラプラス変換

$$Ms^2Y(s) = -CsY(s) - KY(s) + U(s)$$

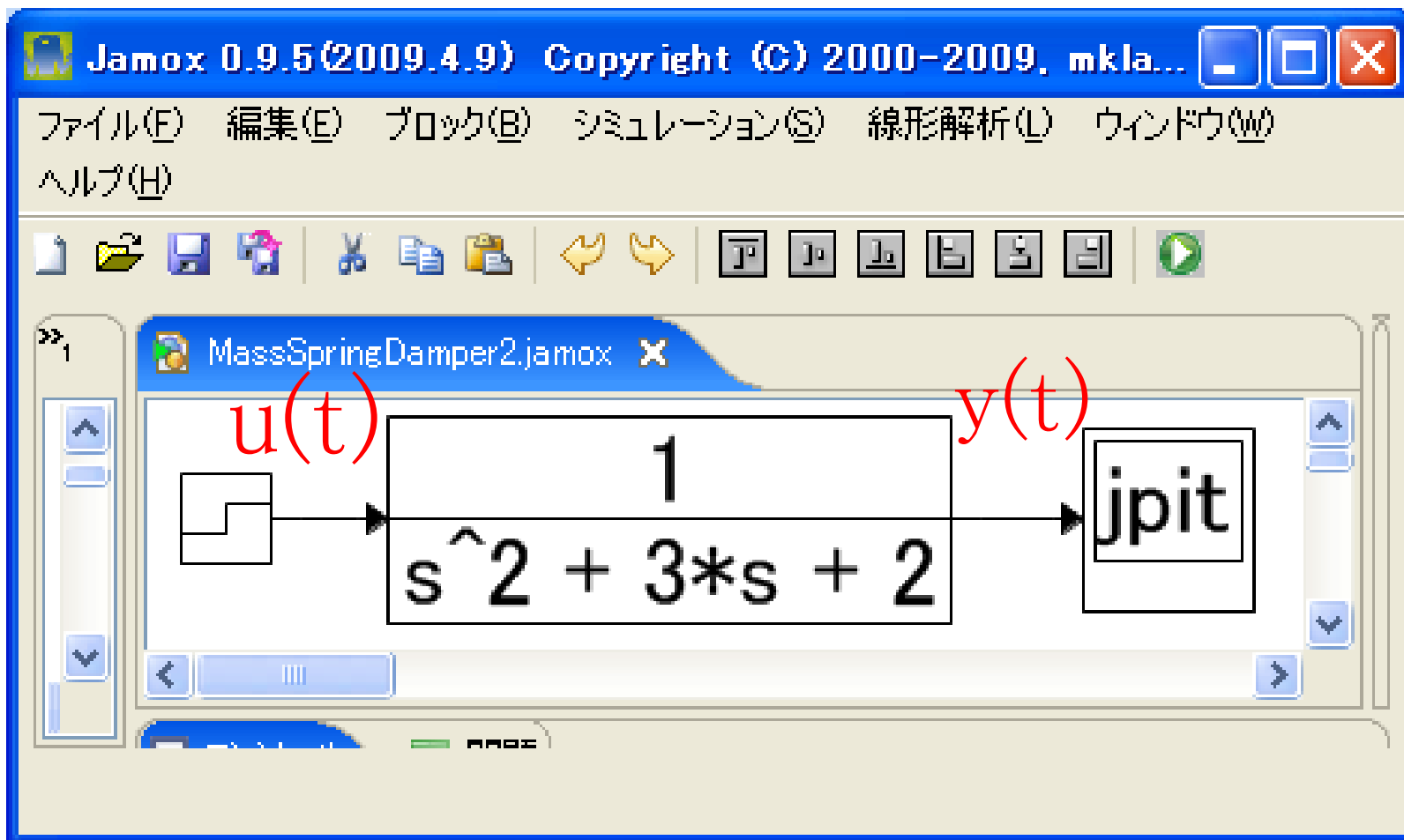
⇩

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

# 質量・ばね・ダンパ系のブロック線図

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

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



# 伝達関数ブロックのパラメータ設定

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

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

初期条件:  
 $y(0) = 0, \dot{y}(0) = 0$



積分器の数だけ設定

(=分母多項式の次数)

# DCサーボモータの伝達関数

$$u(t) = Ri(t) + Li\dot{i}(t) + v_1(t), v_1(t) = K_1\dot{y}(t)$$

$$J\ddot{y}(t) = \tau(t) - C\dot{y}(t), \tau(t) = K_2i(t)$$

 ラプラス変換

$$U(s) = RI(s) + LsI(s) + V_1(s), V_1(s) = K_1sY(s)$$

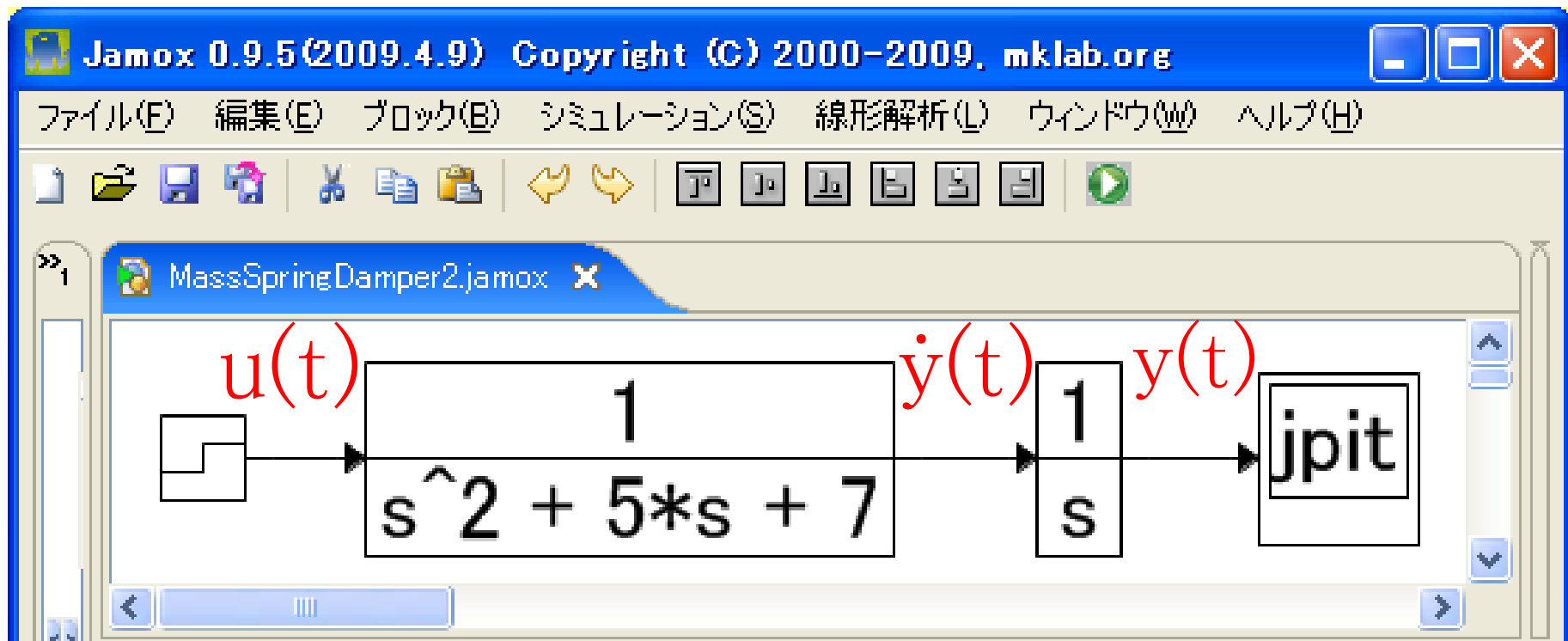
$$Js^2Y(s) = T(s) - CsY(s), T(s) = K_2I(s)$$



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

# DCサーボモータのブロック線図

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



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

# 伝達関数ブロックのパラメータ設定

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

初期条件:  $i(0) = 0, \dot{y}(0) = 0$

基本

システム

図

ポート

分子係数(numerator) [1]

分母係数(denominator) [1\*1 (1\*2+1\*3) (2\*3+1\*1)]

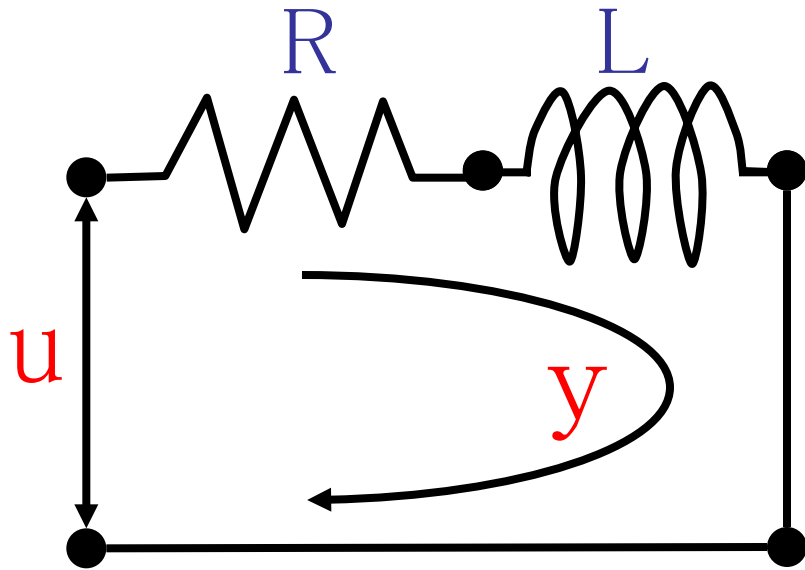
初期状態(initialState) [0 0]'

積分器の数だけ設定

(=分母多項式の次数)

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

# 演習1: RL回路の応答



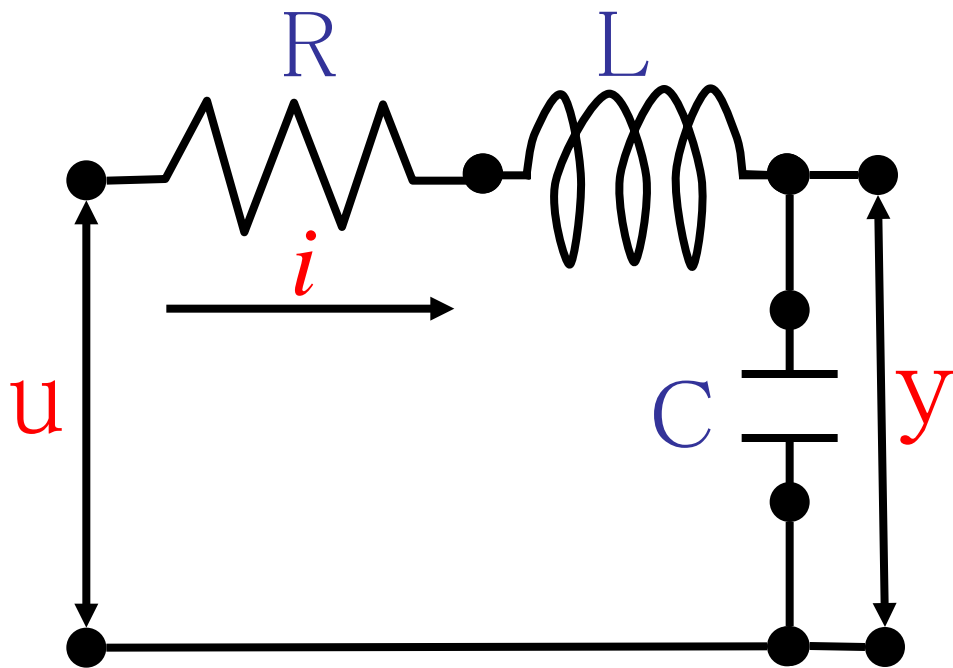
関係式:

$$u(t) = Ry(t) + L\dot{y}(t)$$

- 微分方程式表現からブロック線図を作成し、ステップ応答を求めよ。  
(適当にパラメータ設定)
- 伝達関数を求めよ。
- 伝達関数からブロック線図を作成し、ステップ応答を求めよ。  
(適当にパラメータ設定)



# 演習2: RLC回路の応答



関係式:

$$i(t) = C\dot{y}(t)$$

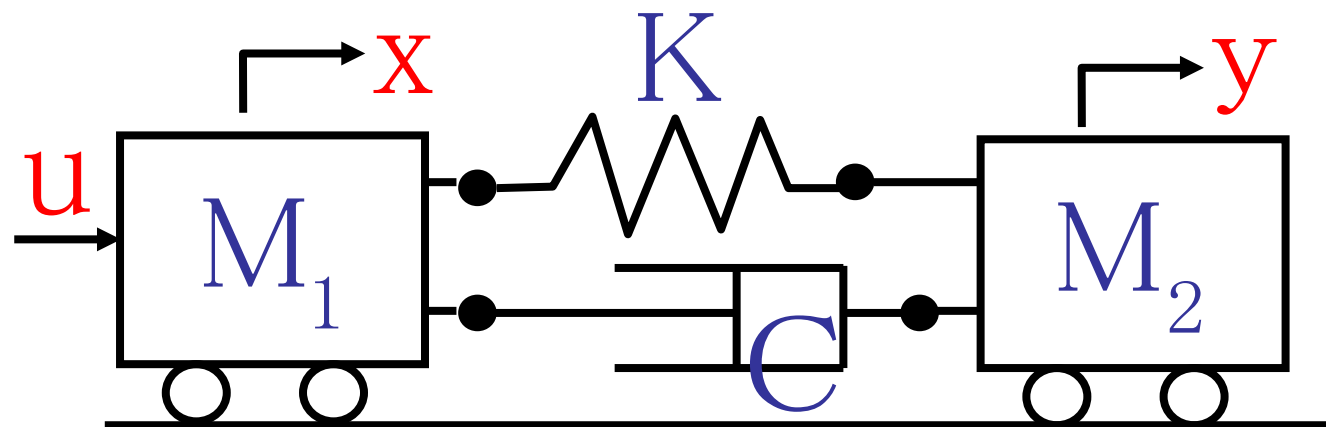
$$u(t) = Ri(t) + L \frac{d}{dt} i(t) + y(t)$$

- 微分方程式表現からブロック線図を作成し、ステップ応答を求めよ。  
(適当にパラメータ設定)
- 伝達関数を求めよ。
- 伝達関数からブロック線図を作成し、ステップ応答を求めよ。  
(適当にパラメータ設定)

# 演習3：初期状態の変更

- テキスト中の各システムについて、初期状態を変更しながらステップ応答を求め、初期状態について考察せよ。

# 演習4: 質量×2・ばね・ダンパ系の応答

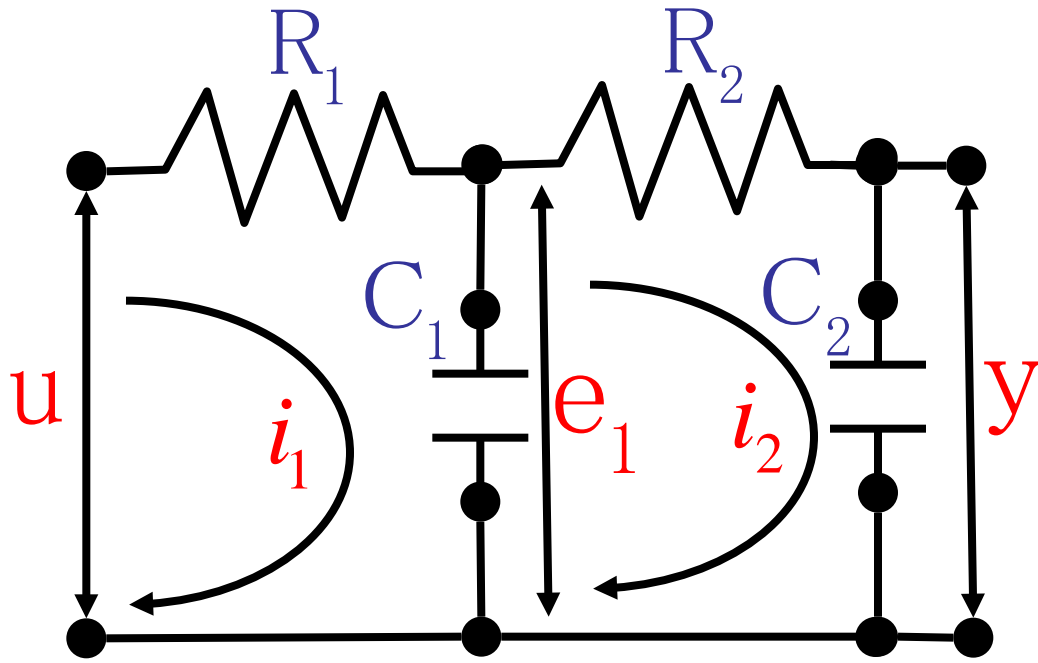


関係式:

$$M_1 \ddot{x}(t) = u(t) + C(\dot{y}(t) - \dot{x}(t)) + K(y(t) - x(t))$$
$$M_2 \ddot{y}(t) = -C(\dot{y}(t) - \dot{x}(t)) - K(y(t) - x(t))$$

- 微分方程式表現からブロック線図を作成し、ステップ応答を求めよ。
- 伝達関数を求めよ。
- 伝達関数からブロック線図を作成し、ステップ応答を求めよ。

# 演習5: RCRC型回路の応答



- 微分方程式表現からブロック線図を作成し、ステップ応答を求めよ。
- 伝達関数を求めよ。
- 伝達関数からブロック線図を作成し、ステップ応答を求めよ。

$$u(t) = R_1 i_1(t) + e_1(t)$$

$$R_2 i_2(t) + y(t) - e_1(t) = 0 \quad y(t) = \frac{1}{C_2} \int_0^t i_2(\tau) d\tau$$

$$e_1(t) = \frac{1}{C_1} \int_0^t (i_1(\tau) - i_2(\tau)) d\tau$$