

Maple には一連の制御システム設計ツールが用意されており、これによって線形時不変動的システムで分析的に作業することができます。DynamicSystems パッケージは、線形時不変システムモデルの作成、操作、シミュレーション、およびプロットを行うためのプロシージャの集合です。Maple 18 には、新しいコマンドに加え、いくつかの既存のコマンドの更新も含まれます。

- すべてのシステムコンストラクタ、StateSpace コマンド、および ToDiscrete コマンドの更新オプション
- 相互接続システム、離散系での動作、およびノルムと共分散行列の新しいコマンド

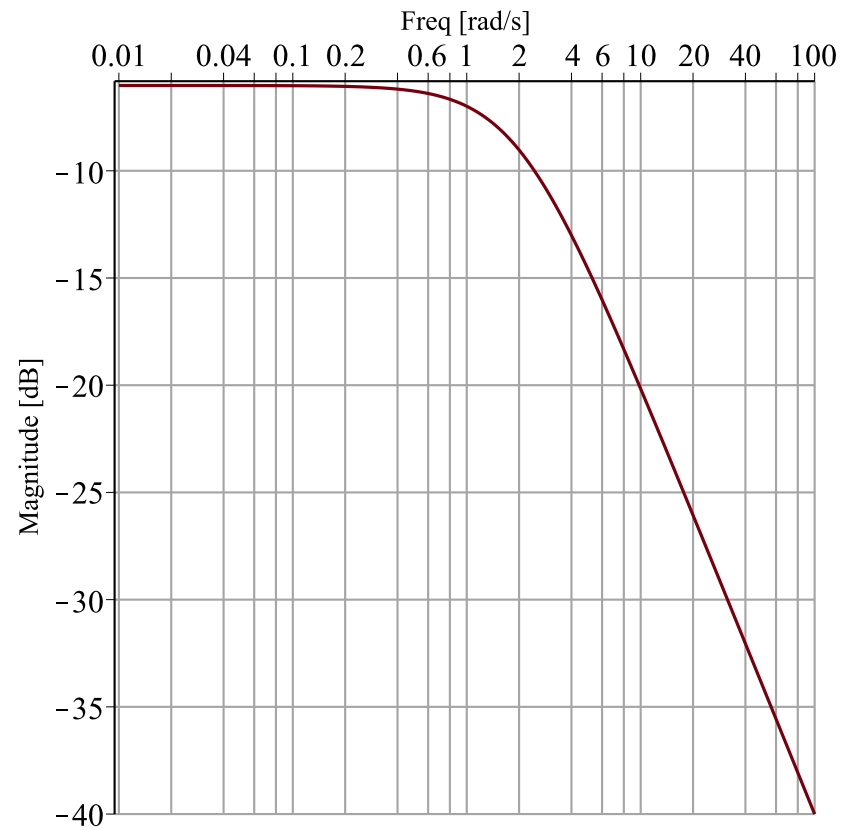
> with(DynamicSystems) :

▼ Parameters オプション

TransferFunction や DiffEquation などのすべてのシステムコンストラクタで、パラメータ名や対応する値を指定する方程式のリストから成る parameters オプションを使用することができます。これらのパラメータは、プロットルーチンなどの数値を必要とするプロシージャで使用されます。数値システムを必要とするほとんどのプロシージャでは、システムに割り当てられたパラメータをオーバーライドする parameters オプションを提供しています。

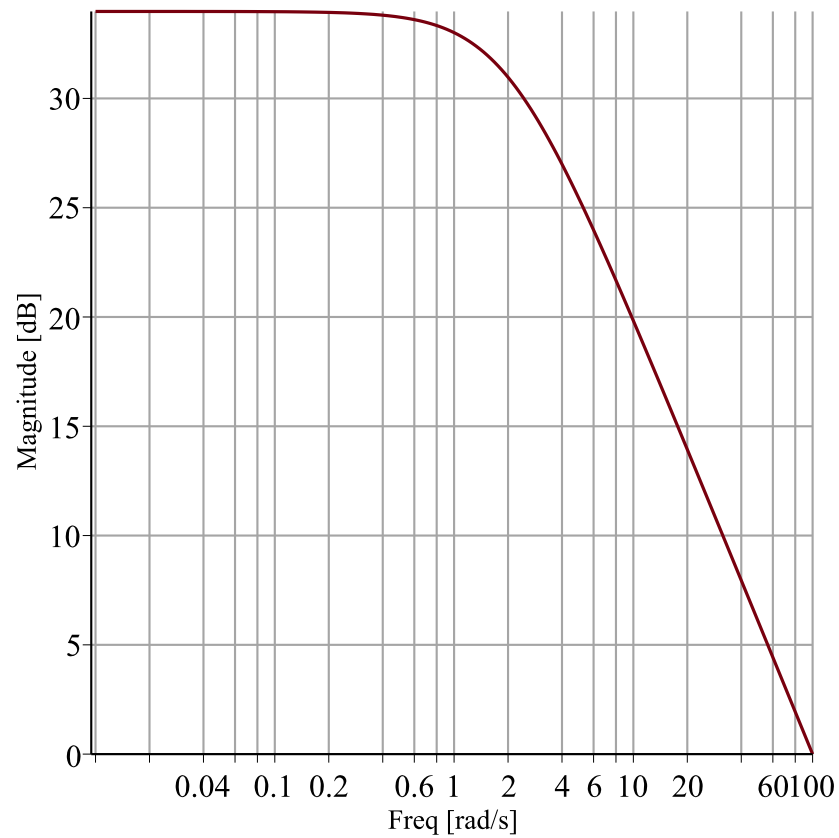
> sys := TransferFunction($\frac{A}{s + \omega}$, parameters = [A = 1, $\omega = 2$]) :

> MagnitudePlot(sys)



ここで、プロットプロシージャの `parameters` オプションは、オブジェクトのパラメータ `A` をオーバーライドします。

> `MagnitudePlot(sys, parameters = [A = 100])`



▼ StateSpace オプション

▼ ランダムシステム

[StateSpace](#) コンストラクタは、`randomtest` や `genbound` など、ランダムかつ安定した状態空間システムを生成するために役立つ、新しいオプションに対応しています。

```
> ss1 := StateSpace('randomtest', 'numstates' = 3, 'numinputs' = 2, 'numoutputs' = 1, 'genbound' = 5.) :
```

```
> PrintSystem(ss1)
```

State Space

continuous

1 output(s); 2 input(s); 3 state(s)

inputvariable = [$u1(t)$, $u2(t)$]

outputvariable = [$y1(t)$]

statevariable = [$x1(t)$, $x2(t)$, $x3(t)$]

$$a = \begin{bmatrix} -4.424931645657024 & -4.024595950005905 & -3.7301318370649392 \\ 0.46881519204983846 & -7.676407537745905 & 4.057919370756192 \\ -2.215017811329516 & 4.133758561390193 & -5.85276313606821 \end{bmatrix}$$

(2.1.1)

$$b = \begin{bmatrix} 3.00280468888800200 & 4.70592781760615608 \\ -0.146243512771587980 & -3.42386918322451717 \\ 4.57166948242945637 & 4.64888535199276554 \end{bmatrix}$$

$$c = [4.15735525189067090 \quad -0.782387173737250308 \quad -3.58113661372784664]$$

$$d = [4.59492426392903042 \quad 2.92207329559554374]$$

▼ 数式システム

StateSpace コンストラクタは、数式要素で状態行列を作成するために使用できる `usesymbols` キーワードオプションに対応しています。

```
> ss2 := StateSpace('numstates'=3, 'numinputs'=2, 'numoutputs'=1, 'usesymbols')
```

```
> PrintSystem(ss2)
```

State Space

continuous

1 output(s); 2 input(s); 3 state(s)

inputvariable = [$u1(t)$, $u2(t)$]

outputvariable = [$y1(t)$]

statevariable = [$x1(t)$, $x2(t)$, $x3(t)$]

$$a = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

$$b = \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \\ b_{3,1} & b_{3,2} \end{bmatrix}$$

$$c = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \end{bmatrix}$$

$$d = \begin{bmatrix} d_{1,1} & d_{1,2} \end{bmatrix}$$

(2.2.1)

▼ ToDiscrete 法

離散化の新しい方法である 1 次ホールド (foh) が [ToDiscrete](#) コマンドに追加されています。0 次ホールド法と同様に、この方法は状態空間、伝達関数、および係数システムタイプに適用できます。

▼ Connection コマンド

相互接続システム用に、いくつかの新しいコマンドが追加されています。

- [AppendConnect](#) : 入出力を付加することで統合された 2 つ以上のシステムオブジェクトについて、同等のシステム表示を作成します。
- [FeedbackConnect](#) : 正または負のフィードバック接続のある 1 つまたは 2 つのシステムオブジェクトについて、同等のシステム表示を作成します。
- [ParallelConnect](#) : 並列接続された 2 つ以上のシステムオブジェクトについて、同等のシステム表示を作成します。

- [SeriesConnect](#): 直列接続された 2 つ以上のシステムオブジェクトについて、同等のシステム表示を作成します。

> `sys1 := StateSpace(usesymbols, numstates = 2, numinputs = 1, numoutputs = 1, symbols = [A1, B1, C1, D1]) :`

> `PrintSystem(sys1)`

State Space

continuous

1 output(s); 1 input(s); 2 state(s)

inputvariable = $[u1(t)]$

outputvariable = $[y1(t)]$

statevariable = $[x1(t), x2(t)]$

$$\mathbf{a} = \begin{bmatrix} A1_{1,1} & A1_{1,2} \\ A1_{2,1} & A1_{2,2} \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} B1_{1,1} \\ B1_{2,1} \end{bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} C1_{1,1} & C1_{1,2} \end{bmatrix}$$

$$\mathbf{d} = \begin{bmatrix} D1_{1,1} \end{bmatrix}$$

> `sys2 := StateSpace(usesymbols, numstates = 2, numinputs = 1, numoutputs = 1, symbols = [A2, B2, C2, D2]) :`

> `PrintSystem(sys2)`

(4.1

State Space

continuous

1 output(s); 1 input(s); 2 state(s)

inputvariable = $[uI(t)]$

outputvariable = $[yI(t)]$

statevariable = $[x1(t), x2(t)]$

$$\mathbf{a} = \begin{bmatrix} A2_{1,1} & A2_{1,2} \\ A2_{2,1} & A2_{2,2} \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} B2_{1,1} \\ B2_{2,1} \end{bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} C2_{1,1} & C2_{1,2} \end{bmatrix}$$

$$\mathbf{d} = \begin{bmatrix} D2_{1,1} \end{bmatrix}$$

(4.2)

> *sys_series* := *SeriesConnect*([*sys1*, *sys2*]) :

> *PrintSystem*(*sys_series*)

State Space

continuous

1 output(s); 1 input(s); 4 state(s)

inputvariable = $[uI(t)]$

outputvariable = $[yI(t)]$

statevariable = $[x1(t), x2(t), x3(t), x4(t)]$

$$a = \begin{bmatrix} A1_{1,1} & A1_{1,2} & 0 & 0 \\ A1_{2,1} & A1_{2,2} & 0 & 0 \\ B2_{1,1} & C1_{1,1} & B2_{1,1} & C1_{1,2} & A2_{1,1} & A2_{1,2} \\ B2_{2,1} & C1_{1,1} & B2_{2,1} & C1_{1,2} & A2_{2,1} & A2_{2,2} \end{bmatrix}$$

$$b = \begin{bmatrix} B1_{1,1} \\ B1_{2,1} \\ B2_{1,1} & D1_{1,1} \\ B2_{2,1} & D1_{1,1} \end{bmatrix}$$

$$c = \begin{bmatrix} D2_{1,1} & C1_{1,1} & D2_{1,1} & C1_{1,2} & C2_{1,1} & C2_{1,2} \end{bmatrix}$$

$$d = \begin{bmatrix} D2_{1,1} & D1_{1,1} \end{bmatrix}$$

(4.3)

> *sys_parallel* := *ParallelConnect*([*sys1*, *sys2*]) :

> *PrintSystem*(*sys_parallel*)

State Space

continuous

1 output(s); 1 input(s); 4 state(s)

inputvariable = $[u1(t)]$

outputvariable = $[y1(t)]$

statevariable = $[x1(t), x2(t), x3(t), x4(t)]$

$$a = \begin{bmatrix} A1_{1,1} & A1_{1,2} & 0 & 0 \\ A1_{2,1} & A1_{2,2} & 0 & 0 \\ 0 & 0 & A2_{1,1} & A2_{1,2} \\ 0 & 0 & A2_{2,1} & A2_{2,2} \end{bmatrix}$$

$$b = \begin{bmatrix} B1_{1,1} \\ B1_{2,1} \\ B2_{1,1} \\ B2_{2,1} \end{bmatrix}$$

$$c = [C1_{1,1} \quad C1_{1,2} \quad C2_{1,1} \quad C2_{1,2}]$$

$$d = [D1_{1,1} + D2_{1,1}]$$

(4.4)

▼ ToContinuous および Resample コマンド

[ToContinuous](#) コマンドは、指定の変換方法を使用して、離散系を連続系に変換します。

> `sysd := TransferFunction(1/(z+a), discrete, parameters = [a=3]) :`

> `sysc1 := ToContinuous(sysd, method = forward) :`

> `PrintSystem(sysc1)`

Transfer Function

continuous

1 output(s); 1 input(s)

inputvariable = [$uI(s)$]

outputvariable = [$yI(s)$]

$$tf_{1,1} = \frac{1}{s + 1 + a}$$

(5.1)

> `sysc2 := ToContinuous(sysd, method = prewarp)` :

> `PrintSystem(sysc2)`

Transfer Function

continuous

1 output(s); 1 input(s)

inputvariable = [$uI(s)$]

outputvariable = [$yI(s)$]

$$tf_{1,1} = \frac{s - 2.}{a s - s - 2. a - 2.}$$

(5.2)

Resample コマンドは、新しいサンプリング時間で離散系のリサンプリングを行います。

> `sysd2 := Resample(sysd, 2, method = prewarp)` :

> `PrintSystem(sysd2)`

Transfer Function

discrete; samptime = 2

1 output(s); 1 input(s)

inputvariable = [$uI(z)$]

outputvariable = [$yI(z)$]

$$tf_{1,1} = \frac{z + 3.}{a z + 3. z + 3. a + 1}$$

(5.3)

▼ ノルムおよびノイズ解析

[NormH2](#) および [NormHinf](#) コマンドは、システムの H_2 および H_∞ ノルムをそれぞれ計算します。

```
> sys := TransferFunction( $\frac{1}{(s+1)^2}$ ):
```

```
> NormH2(sys)
```

0.5000000000000000

(6.1)

```
> NormHinf(sys)
```

1.000001000

(6.2)

[Covariance](#) コマンドは、白色ガウスノイズから得られる入力を使用して、出力共分散行列を計算します。

```
> Covariance(sys, <<1>>)
```

[0.2500000000000000]

(6.3)

▼ 参照

[DynamicSystems](#)