

概説

この Maple のワークシートでは、MapleSim で作成した物理モデルをインポートして、等価な数式モデルの自動生成機能により、さらに、数式ベースでモデルについて、解析やパラメータの調整やシミュレーションなど行われています。

この事例の使用には、Maple および MapleSim が必要となります。

MapleSim という製品は、Maple の数式処理を計算エンジンとして開発された直観的な複合領域の物理モデリングツールです。

製品詳細は[こちら](#)よりご参照ください。

Maple および MapleSim を組み合わせて活用することで、プラントモデルのシステムを直観的にモデリングし、シミュレーションすることおよび数式ベースで詳細にシステムの特性を解析することを両方実現することができます。

数式モデルによる解析：

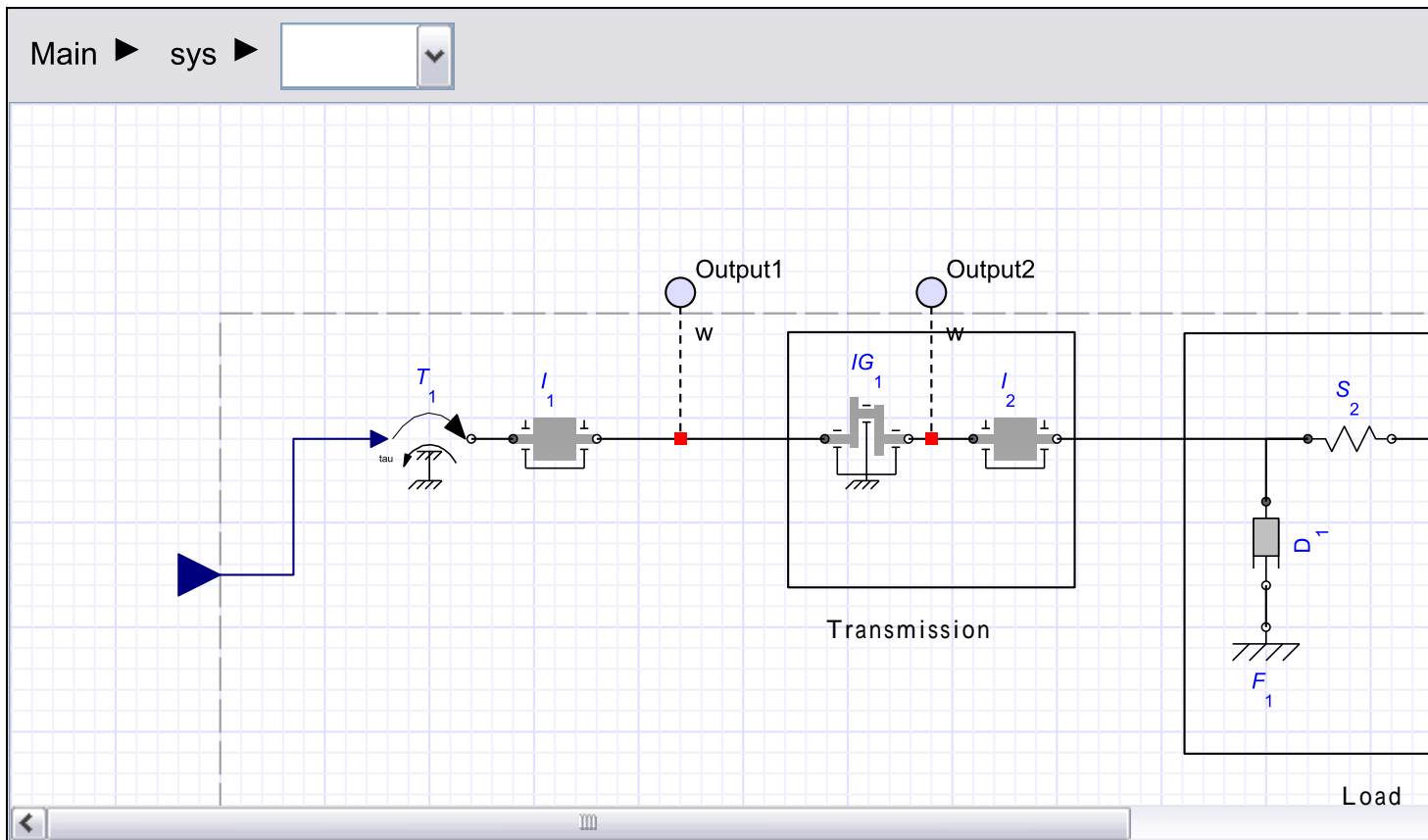
モデルのインポート方法：

1. 以下ウィンドウの上でマウス右ボタンをクリックし、「コンポーネントプロパティ」を選択します。
2. 「ファイル」の「参照」ボタンをクリックし、該当の MapleSim モデルファイルを添付します。(今回は「SimpleDriveline.msim」が該当の MapleSim モデルファイルになります。)
3. コンボボックスより、「sys」を選択し、対象のサブシステムを表示させます。

4. ワークシート全体を簡単に実行するには、 Maple 上のコンテキストバ

ーより枠内のボタンを押します。Mapleワークシート上で各行を確認しながら実行するには、Enter キーを押します。

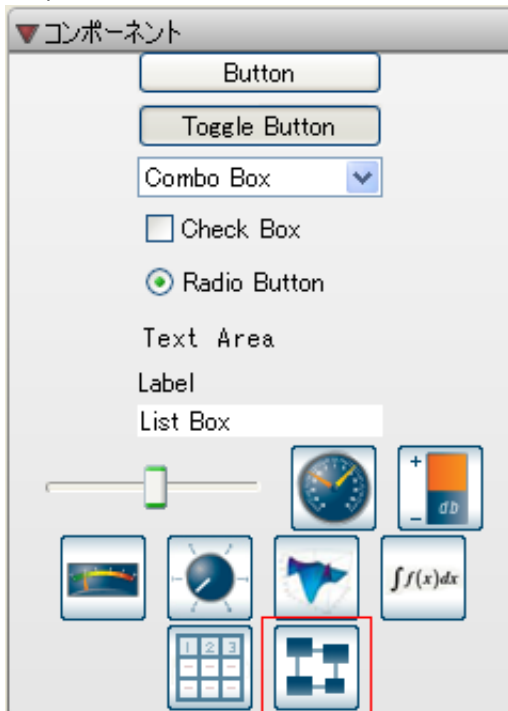
MapleSim で作った物理モデルをインポート：



参考：上記のMapleSim コンポーネントをワークシート上でゼロから作成する場合は、以下方法となります。

1. 「MapleSim コンポーネント」を挿入：

Maple 上の左側のパレット：「コンポーネント」パレットより挿入可能です。



2. 対象モデルのインポート :

「MapleSim コンポーネント」の上でマウス右ボタンをクリックし、「コンポーネントプロパティ」を選択し、「参照」ボタンをクリックして、該当の MapleSim のモデルファイル(拡張子が .msim)を指定して、インポートします。

3. 解析対象となるサブシステムを選択します。

MapleSim 上で事前に解析対象となる部分をサブシステム化する必要があります。サブシステム化の詳細については、[こちら](#)よりご確認ください。

自動生成された数式モデルの抽出 :

1. [MapleSim](#) パッケージの [LinkModel](#) コマンドにより、対象のサブシステムの数式や入出力変数やパラメータなどモジュールとして生成 :

```
> A:=MapleSim:-LinkModel();
```

2. 対象のプラントモデルの数式(パラメータを残したまま)を抽出 :

MapleSim でパラメータ化しておいた c(バネ定数)、d(ダンパー定数)、J(イナーシャのモーメント) に値が代入されずに数式を生成しています。

sys_SprC c, sys_DamC d sys_J Jに該当し、sys というのは、サブシステム名となります。
それは、params オプションにより、明確に指定することができます。

```
DAEs:=A:-GetEquations('output'='daes','simplify'=true,filter={},params=[sys_J,sys_SprC,sys_DamC]);
```

$$\begin{aligned} DAEs := & \left\{ -\frac{1}{10} \frac{1}{sys_J} (10 sys_IG1_flange_b_tau(t) + 10 sys_DamC sys_I2_w(t) \right. & (2.1.1) \\ & - 10 sys_SprC sys_I3_phi(t) + 10 sys_SprC sys_I2_phi(t) + 10 sys_u1(t) sys_J \\ & + sys_IG1_flange_b_tau(t) sys_J) = 0, \frac{d}{dt} sys_I2_phi(t) = sys_I2_w(t), \\ & \frac{d}{dt} sys_I2_w(t) = -\frac{1}{sys_J} (sys_IG1_flange_b_tau(t) + sys_DamC sys_I2_w(t) \\ & - sys_SprC sys_I3_phi(t) + sys_SprC sys_I2_phi(t)), \frac{d}{dt} sys_I3_phi(t) \\ & = sys_I3_w(t), \frac{d}{dt} sys_I3_w(t) = \frac{1}{sys_J} (-2 sys_SprC sys_I3_phi(t) \\ & + sys_SprC sys_I2_phi(t) - sys_DamC sys_I3_w(t) + sys_SprC sys_I4_phi(t)), \\ & \frac{d}{dt} sys_I4_phi(t) = sys_I4_w(t), \frac{d}{dt} sys_I4_w(t) = \\ & \left. \frac{- sys_SprC sys_I4_phi(t) - sys_SprC sys_I3_phi(t)}{sys_J} \right\} \end{aligned}$$

3. プラントモデルの変数を抽出 :

```
> Vars:=A:-GetVariables();
Vars := [sys_I2_phi(t), sys_I2_w(t), sys_I3_phi(t), sys_I3_w(t), sys_I4_phi(t),
          sys_I4_w(t), sys_IG1_flange_b_tau(t), sys_u1(t)] (2.1.2)
```

4. プラントモデルのパラメータを抽出し、Params1 とする：

```
> Params1:=A:-GetParameters(allparams);
Params1 := [sys_D1_phi_nominal = 1/10000, sys_D2_phi_nominal = 1/10000, sys_DamC
            = 2, sys_F1_phi0 = 0, sys_F2_phi0 = 0, sys_I1_J = 1/10, sys_IG1_ratio = 10, sys_J = 2,
            sys_S2_phi_rel0 = 0, sys_S3_phi_rel0 = 0, sys_SprC = 10000] (2.1.3)
```

▼ DynamicSystems パッケージによるシステム特性の解析：

[DynamicSystems](#) パッケージは Maple 本体に標準で提供されている動力学解析機能をまとめたものです。

このパッケージ内に、微分方程式や伝達関数や状態空間によるシステムの表現や、非線形システムの線形化や、ボード線図 / ナイキスト線図 / ZeroPole線図などの描画や、シミュレーションなど実現するためのコマンド群が標準で実装されています。

1. 上記システムの数式モデル DAEs よりシステムオブジェクトを生成：

システムの入力変数が $sys_u1(t)$ ，出力変数が $sys_I4_phi(t)$ (角度センサーの出力) となります。
 そういったような情報を以下のシステムオブジェクトの出力結果より確認できます。

```
> with(DynamicSystems);
> SYSOBJ:=NewSystem(convert(DAEs,list),[sys_u1(t)],[sys_I4_phi(t)]);
SYSOBJ:=
Diff. Equation
continuous
1 output(s); 1 input(s)
inputvariable = [sys_u1(t)]
outputvariable = [sys_I4_phi(t)] (2.2.1)
```

システムを微分方程式表現から等価な伝達関数表現に変換することが可能です。

```
> SYSOBJT:=TransferFunction(SYSOBJ);
SYSOBJT:=-tf[1,1];
(10 sys_SprC^2) / ((10 sys_J^2 + sys_J^3) s^6 + (2 sys_J^2 sys_DamC + 10 sys_DamC sys_J) s^5
+ (4 sys_J^2 sys_SprC + sys_DamC^2 sys_J + 30 sys_J sys_SprC) s^4
+ (10 sys_DamC sys_SprC + 5 sys_J sys_SprC sys_DamC) s^3 + (3 sys_SprC^2 sys_J
```

$$+ \text{sys_DamC}^2 \text{sys_SprC} + 10 \text{sys_SprC}^2) s^2 + 2 \text{sys_DamC} \text{sys_SprC}^2 s)$$

同様のようにならば等価な状態空間に変換することも可能です。

```
> SYSOBJSS:=StateSpace(SYSOBJ):
PrintSystem(SYSOBJSS);
```

State Space

continuous

1 output(s); 1 input(s); 6 state(s)

inputvariable = [sys_u1(t)]

outputvariable = [sys_I4_phi(t)]

statevariable = [sys_I2_phi(t), sys_I2_w(t), sys_I3_phi(t), sys_I3_w(t), sys_I4_w(t), sys_IG1_flange_b_t

$$a = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{\text{sys_SprC}}{10 + \text{sys_J}} & -\frac{\text{sys_DamC}}{10 + \text{sys_J}} & \frac{\text{sys_SprC}}{10 + \text{sys_J}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{\text{sys_SprC}}{\text{sys_J}} & 0 & -\frac{2 \text{sys_SprC}}{\text{sys_J}} & -\frac{\text{sys_DamC}}{\text{sys_J}} & \frac{\text{sys_SprC}}{\text{sys_J}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{\text{sys_SprC}}{\text{sys_J}} & 0 & -\frac{\text{sys_SprC}}{\text{sys_J}} & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ \frac{10}{10 + \text{sys_J}} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$c = [0 \ 0 \ 0 \ 0 \ 1 \ 0]$$

$$d = [0]$$

ここまでは、パラメータ化したパラメータを残したまま等価な数式や状態空間を生成することを紹介しています。

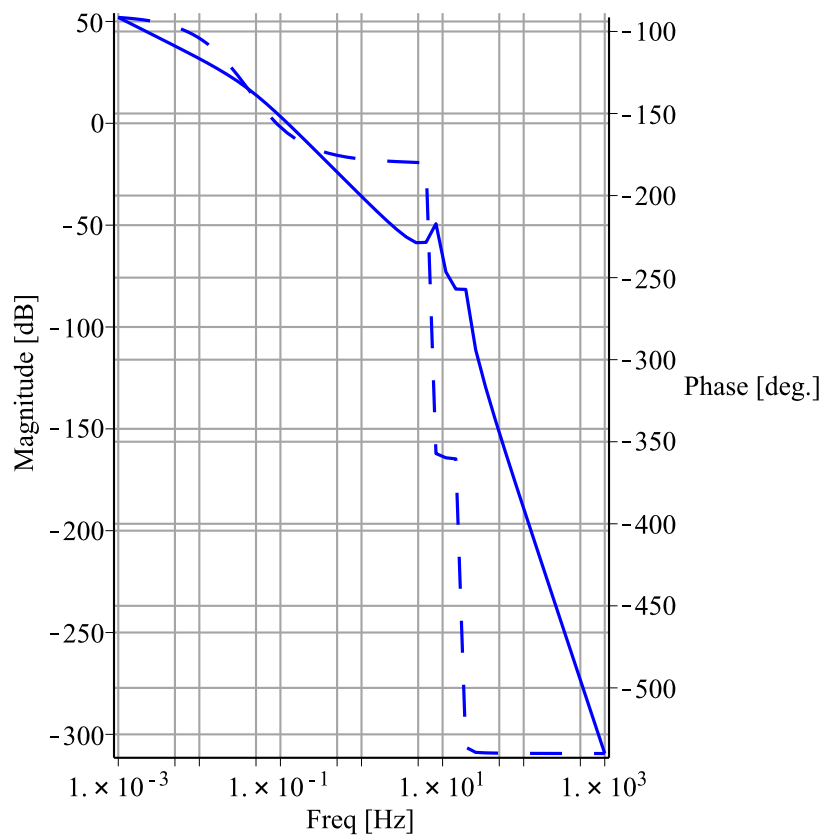
2. 数値を与えてボード線図を描画：

ボード線図により、システムの周波数特性を解析することができます。

確認しやすいように、振幅および位相を同一のグラフに表示させます。

```
> BodePlot(SYSOBJ,parameters=Params1,hertz=true,output=dualaxis,color=
```

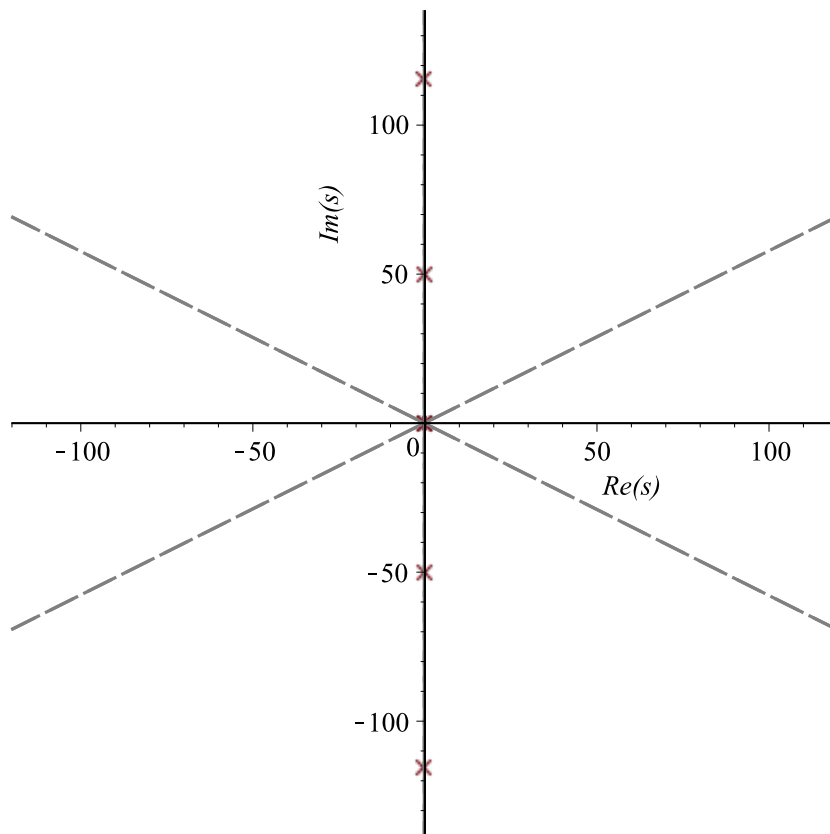
blue);



3. 根軌跡を描画：

info = 'data' オプションを使うことで、根軌跡の情報を data に割り当てることができます。
さらに、data:-Kbranches() を実行すると、分岐する K の値が求められます。

```
> RootLocusPlot(SYSOBJ,parameters=Params1,info = 'data');
```



```
> data:=-Kbranches();
```

```
[0.02499997070]
```

(2.2.4)

```
>
```

(2.2.5)

▼ パラメータの再設定によるシステム解析：

MapleSim パッケージの [SetParameters](#) コマンドは、オリジナルのパラメータを再設定することができます。

▼ ボード線図の確認：

1. オリジナルのパラメータによるボード線図を BP1 とし、出力：色をブルーにしています。

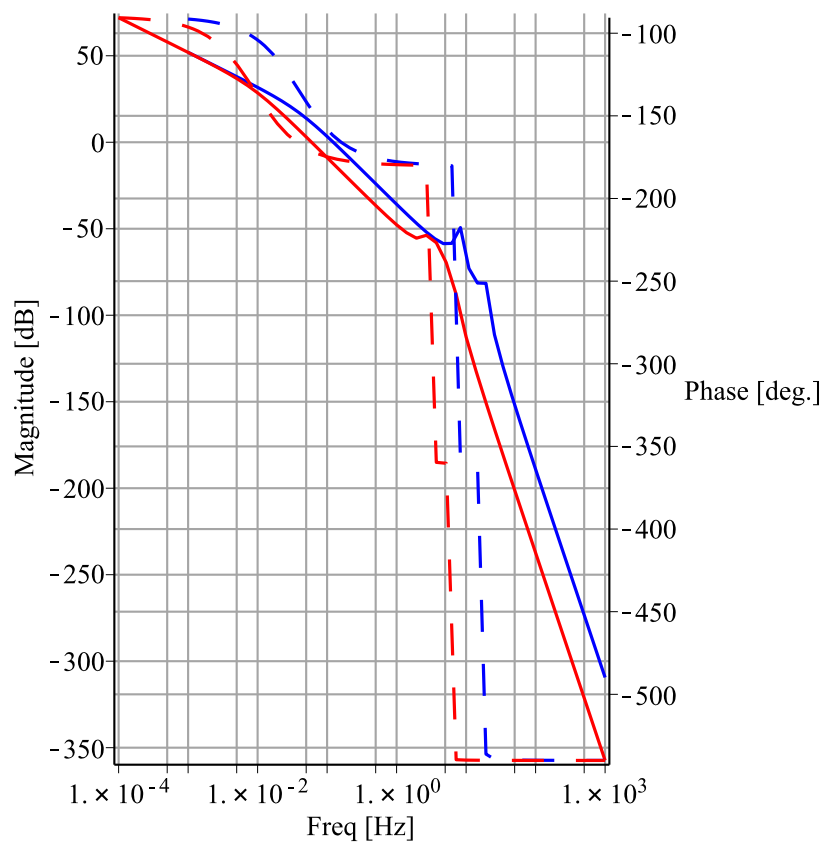
```
> BP1:=BodePlot(SYSOBJ,parameters=Params1,hertz=true,output=dualaxis,
color=blue):
```

2. イナーシャJの値を 2 から 20 に変更し Params2 に割り当て、ボード線図を BP2 として描画 :
色を赤にしています。

```
> A:=-SetParameters([sys_J=20]):  
> Params2:=A:-GetParameters(allparams):  
> BP2:=BodePlot(SYSOBJ,parameters=Params2,hertz=true,output=dualaxis,  
color=red):
```

3. イナーシャJの値を変更する前後のボード線図を同一のグラフに表示し、変更によるシステム特性への影響を確認 :

```
> with(plots):  
> display(BP1,BP2);
```



▼ シミュレーションの確認 :

[DynamicSystems\[Simulate\]](#) コマンドにより、システムをシミュレーションすることができます。

ここまでは、パラメータの再設定により、Params (オリジナル), Params2 (イナーシャJ) を生成してい

ます。さらに、ダンパー d の値を再設定し、Params3 にしておきます。
最後上記変更によるシステムのシミュレーション結果を比較してみます。

1 . ダンパー d の値を 2 から 10に変更し、Params3 に割り当て :

```
[ > A:-SetParameters([sys_DamC=10]):  
[ > Params3:=A:-GetParameters(allparams):
```

2 . 3 組のパラメータによる回転角速度のシミュレーション結果について、自動処理するための簡単なプログラム(for ループ)を用意 :

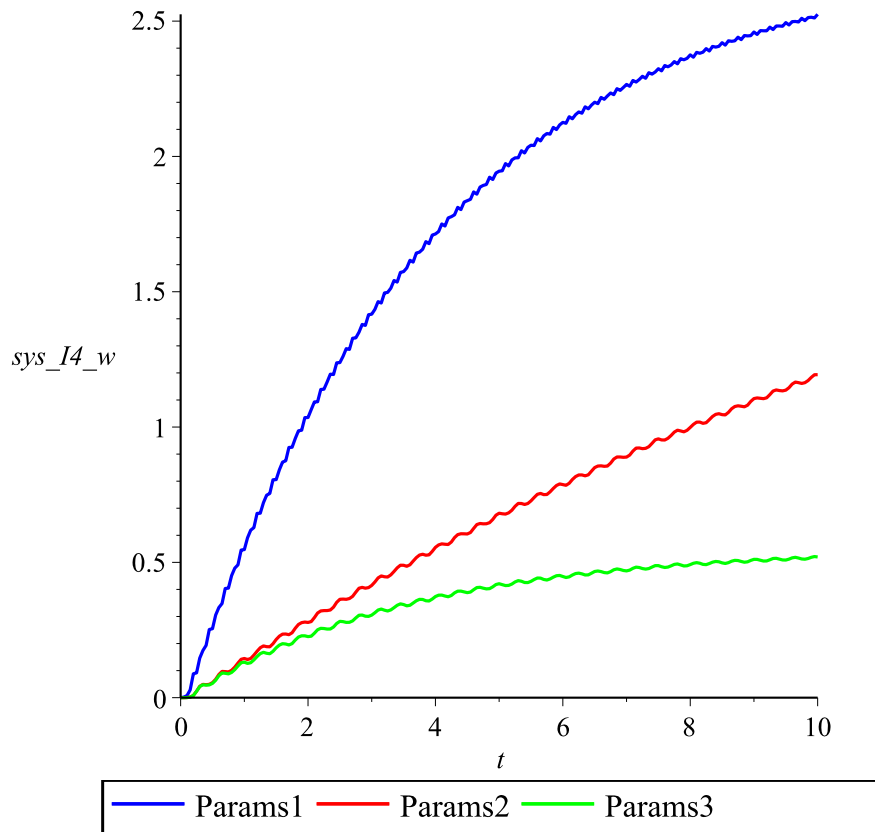
システムにステップ入力を与えて、シミュレーション時間を 0~10s にしています。

CorL では、各グラフの色をリストとして用意し、LegL では、各グラフのレジェンドを用意していません。

```
[ > i:=NULL:  
[ > CorL:=[blue,red,green]:  
  LegL:=["Params1","Params2","Params3"]:  
[ > for i from 1 to 3 do  
  Sim || i := Simulate(SYSOBJ, Step(1,0.1,0.1), parameters = Params ||  
  i );  
  SimP || i := odeplot(Sim || i, [t, sys_I4_w(t)], 0 .. 10,color=CorL  
  [i],legend=LegL[i]):  
[ end do:
```

3 . パラメータの再設定によるシミュレーションを同一のグラフに表示 :

```
[ > display(SimP1,SimP2,SimP3);
```



A:-SetParameters(Params1):

以上で、DynamicSystems パッケージを用いて、MapleSim で作成した物理モデルについて、ダイナミックシステムの解析やパラメータ変更によるシミュレーションなどの方法を解説いたしました。

さらに、GUI コンポーネントを適宜に利用することでマウス操作による対話型アプリケーションの作成が可能となります。

なお、Maple のワークシートを MapleSim のモデルに添付できるので、MapleSim と連動を実現できます。