

ACADEMIC PROSPECTIVE IN EXACT SCIENCES AND ENGINEERING USING COMPUTER ALGEBRA

Francisco Javier Triveño Vargas, francisco.vargas@embraer.com.br

Empresa Brasileira de Aeronáutica - EMBRAER - Brazil

Hans Christian Muller Santa Cruz, hans.muller.santacruz@gmail.com

Univesidade Mayor de San Simon - UMSS - Bolivia

Abstract. *The present paper has as objective to show the effort that being doing by some universities in Bolivia for implementation of computational algebraic tools (Ex. Matlab, Mathematica®) in its analytical contents, looking for the academic improvement in exact sciences and engineering. To reach this objective, it is systematically described the methodology presented to students of course **Modelling and Simulation of Systems Using Mathematica®** actually offered in San Simon University. This tool is used as a didactic help to study mechanical and aerospace systems as the inverted pendulum and two degree of freedom satellite. Its application is extended to obtain the nonlinear models, determination of equilibrium points, linear models derivation, numerical solution and simulation.*

Keywords: Algebraic tools; Mathematica; Modelling; Simulation

1. INTRODUCTION

Due to the constant technological advance and looking for a high level of competitiveness in the professional market, many universities in the world have introduced in their analytical programs, computational algebraic tools such as *Matlab* and *Mathematica®* [5],[2]. These tools allow to solve a diverse range of problems in exact sciences and engineering also in other areas [7] [3].

Considering the given reasons, this paper has as objective to show the effort that is being done by some universities in Bolivia [9], [10], [11] for implementation of computational algebraic tools in its analytical contents, looking for the academic improvement in exact sciences and engineering.

The first experience with the use of computational algebraic tools in Bolivia, specifically with *Mathematica®*, was taken ahead at 2006, when the course of *Variable Structure Control with Sliding Modes* was prepared in the San Simon University (Cochabamba-Bolivia). In this course, *Mathematica®* was used to obtain the nonlinear models, equilibrium points determination and linear models derivation for its later application in *Variable Structure Control* algorithms. In 2007 the course of *Modelling, Simulation and Control of Systems* was developed at Valle University (Cochabamba-Bolivia). Finally, in 2008, the course of *Modelling and Simulation of Systems using Matlab-Simulink and Mathematica®* was prepared in the Technical University of Oruro-Bolivia.

During the execution of the courses, it was observed the lack of familiarity of the students and some professionals with the tool, because several of them did not have knowledge or practice with the tool in basic courses of its professional training. Nevertheless, it was also possible to verify that the predisposition to learn from students increased, being mentioned as the main cause the use of the *Mathematica®* as didactic help, whose strictly symbolic language offer better understanding of different kind of problems [4].

Based in the prepared courses, actually the San Simon University offers the course of *Modelling and Simulation of Systems using Mathematica®*, where mechanical and aerospace systems as the inverted pendulum and two degree of freedom satellite are studied. In this sense the application of *Mathematica®* is extended to obtain the nonlinear models, determination of equilibrium points, linear models derivation, numerical solution and finally its simulation.

Following the methodology presented to the students, this paper describes in systematic form the development of a classroom to model, solve and simulate the mechanical and aerospace systems.

2. MATHEMATICAL MODELS

In the course of *Modelling and Simulation of Systems using Mathematica®*, the mathematical models are described through differential equations [1]. These equations are obtained using laws such as Newton's second law, energy conservation law and Kirchoff's law. In the course, some nonlinear models are obtained through Lagrange's equation [12].

$$L = T - V \tag{1}$$

where T is kinetic energy and V is potential energy. Given equation (1), the next function is defined.

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_i} \right) - \left(\frac{\partial L}{\partial x_i} \right) = u_i \tag{2}$$

where x_i and \dot{x}_i are the position and velocity variables and the variable u_i corresponds to the external forces applied on the system. In the following subsection, basic equations of inverted pendulum and two degrees of freedom satellite are described.

2.1 Inverted pendulum

The system consists of a pendulum in upright position mounted on top of cart moving back and forth along a straight line in the horizontal axes in response to external force u . The position of the cart is denoted by x and the angle of deflection of the pendulum from the vertical is denoted by θ . The masses of the cart and the pendulum bob are M_1 and M_2 . The length of the pendulum arm is l . The figure 1 illustrates the problem.

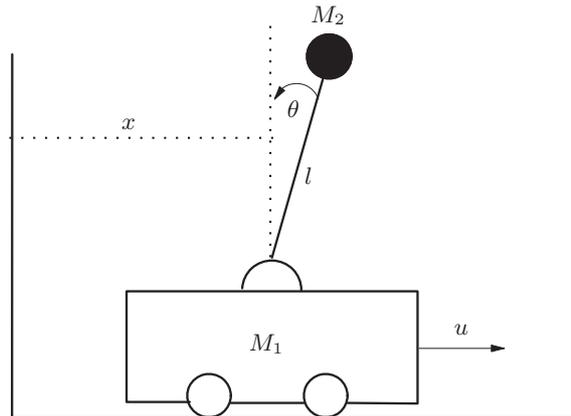


Figure 1. Inverted pendulum

The kinetics energies due M_1 and M_2 , are

$$T_1 = \frac{1}{2} M_1^2 \dot{x}^2 \tag{3}$$

$$T_2 = \frac{1}{2} M_2^2 v_2^2 \tag{4}$$

In equation (4), v_2 is defined considering the horizontal $p_h = x + l \sin \theta$ and vertical $p_v = l \cos \theta$ positions of M_2 given by:

$$v_2^2 = \left(\frac{dp_h}{dt} \right)^2 + \left(\frac{dp_v}{dt} \right)^2 \tag{5}$$

2.2 Two degree of freedom satellite

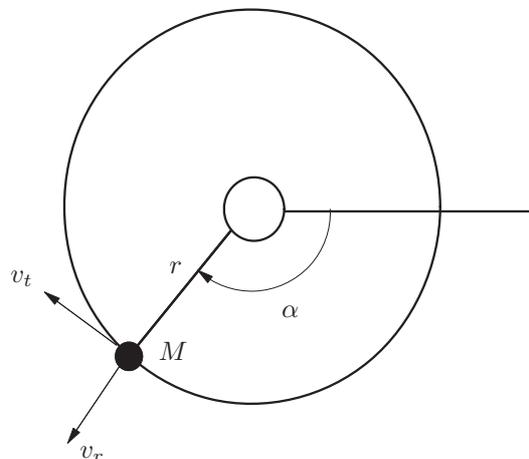


Figure 2. Two degree of freedom satellite

The orbit of a two degree of freedom satellite with mass M can be described by two nonlinear coupled equations, which are obtained considering the sum of kinetic energy due tangential $v_t = r(t)\dot{\alpha}(t)$ and orthogonal velocities $v_r = \dot{r}(t)$. It

is assumed that the satellite is capable to thrusting in both directions, with force magnitudes F_t , and F_r respectively. The problem is illustrated in figure 2 and lagrange's equation is given by:

$$L = \frac{1}{2}M (v_t^2 + v_r^2) \quad (6)$$

3. MODELLING WITH MATHEMATICA®

In the following subsections the nonlinear models of examples presented are obtained.

3.1 Inverted pendulum model

The following code shown how to obtain the nonlinear model of inverted pendulum using the equations (1), (2) (3) and (4):

```

T1[t] = 1/2 M1 x[t]^2
Ph[t] = x[t] + l Sin[theta[t]]
Pv[t] = l Cos[theta[t]]
aux01[t] = Dt Ph[t]
aux02[t] = Dt Pv[t]
v2[t] = Simplify[aux01[t]^2 + aux02[t]^2]
T2[t] = 1/2 M2 v2[t]
L[t] = T1[t] + T2[t]
aux03[t] = Simplify[Dx[t] L[t]]
aux04[t] = Simplify[Dx[t] L[t]]
aux05[t] = FullSimplify[Dt aux03[t]]
resp1 = FullSimplify[Solve[aux05[t] == u, {theta[t]}]]
aux06[t] = Simplify[Dtheta[t] L[t]]
aux07[t] = Simplify[Dtheta[t] L[t]]
aux08[t] = FullSimplify[Dt aux06[t] - aux07[t]]
Tv[t] = M2 l g Sin[theta[t]]
resp2 = FullSimplify[Solve[aux08[t] == Tv[t], {x[t]}]]
resp1 /. {x -> g Tan[theta] - l Sec[theta][t] theta[t]}
ddtheta[t] = Sec[theta[t]] / (l M2) (u + l Sin[theta[t]] M2 theta[t]^2 - (M1 + M2) (g Tan[theta[t]] - l Sec[theta[t]] theta[t]))
ddthetaF = FullSimplify[Solve[ddtheta[t] - theta[t] == 0, {theta[t]}]]
ddxF = resp2 /. {theta -> Sec[theta[t]] (-u + g (M1 + M2) (Tan[theta[t]] - l M2 Tan[theta[t]] theta[t]^2)) / (l (Sec[theta[t]]^2 M1 + M2 Tan[theta[t]]^2))}

```

Looking the last two lines of command, the equations for $\ddot{\theta}$ and \ddot{x} are:

$$\ddot{\theta} = \frac{\text{Sec}[\theta[t]](-u + g(M_1 + M_2)(\text{Tan}[\theta[t]] - l M_2 \text{Tan}[\theta[t]] \dot{\theta}^2))}{l(\text{Sec}[\theta[t]]^2 M_1 + M_2 \text{Tan}[\theta[t]]^2)} \quad (7)$$

$$\ddot{x} = g \text{Tan}[\theta[t]] - \frac{\text{Sec}[\theta[t]](-u + g(M_1 + M_2)(\text{Tan}[\theta[t]] - l M_2 \text{Tan}[\theta[t]] \dot{\theta}^2))}{\text{Sec}[\theta[t]]^2 M_1 + M_2 \text{Tan}[\theta[t]]^2} \quad (8)$$

Doing $x_1 = \theta$, $x_2 = \dot{\theta}$, $x_3 = x$ and $x_4 = \dot{x}$, equation of inverted pendulum is write in state variables as:

$$\begin{pmatrix} \dot{x}_1 = & x_2 \\ \dot{x}_2 = & \frac{\text{Sec}[\theta[t]](-u + g(M_1 + M_2)(\text{Tan}[\theta[t]] - l M_2 \text{Tan}[\theta[t]] \dot{\theta}^2))}{l(\text{Sec}[\theta[t]]^2 M_1 + M_2 \text{Tan}[\theta[t]]^2)} \\ \dot{x}_3 = & x_4 \\ \dot{x}_4 = & g \text{Tan}[\theta[t]] - \frac{\text{Sec}[\theta[t]](-u + g(M_1 + M_2)(\text{Tan}[\theta[t]] - l M_2 \text{Tan}[\theta[t]] \dot{\theta}^2))}{\text{Sec}[\theta[t]]^2 M_1 + M_2 \text{Tan}[\theta[t]]^2} \end{pmatrix} \quad (9)$$

3.2 Two degree of freedom satellite model

The next code is used to obtain the nonlinear model of two degree of freedom satellite:

$$L = \frac{1}{2}M (r[t]^2 \dot{\alpha}^2 + \dot{r}[t]^2)$$

$$\begin{aligned}
 aux03[t] &= \text{Simplify}[\partial_{r[t]} L[t]] \\
 aux04[t] &= \text{Simplify}[\partial_{\dot{r}[t]} L[t]] \\
 aux05[t] &= \text{FullSimplify}[\partial_t aux04[t] - aux03[t]] \\
 resp1 &= \text{FullSimplify}[\text{Solve}[aux05[t] == F_r[t] - \frac{kM}{r[t]^2}, \{\ddot{r}[t]\}]] \\
 aux06[t] &= \text{Simplify}[\partial_{\alpha[t]} L[t]] \\
 aux07[t] &= \text{Simplify}[\partial_{\dot{\alpha}[t]} L[t]] \\
 aux08[t] &= \text{FullSimplify}[\partial_t aux06[t] - aux07[t]] \\
 resp2 &= \text{FullSimplify}[\text{Solve}[aux08[t] == F_t[t], \{\ddot{\alpha}[t]\}]]
 \end{aligned}$$

$$\ddot{r}[t] = -\frac{k}{r[t]^2} + \frac{F_r[t]}{M} + r[t]\dot{\alpha}[t]^2 \quad (10)$$

$$\ddot{\alpha}[t] = -\frac{F_t}{Mr[t]^2} - \frac{2\dot{r}[t]\dot{\alpha}[t]^2}{Mr[t]} \quad (11)$$

Doing $x_1 = r$, $x_2 = \dot{r}$, $x_3 = \alpha$ and $x_4 = \dot{\alpha}$, equations (10) and (11) are write in state variables as:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} x_2 \\ -\frac{k}{x_1^2} + \frac{F_r}{M} + x_1 x_4^2 \\ x_4 \\ \frac{F_t}{M x_1^2} - \frac{2x_2 x_4}{x_1} \end{pmatrix} \quad (12)$$

4. EQUILIBRIUM POINTS

An equilibrium point for the system represented by $\dot{x} = f(x, u)$ is a point (x_{eq}, u_{eq}) where \dot{x} is equal to zero.

4.1 Equilibrium points of inverted pendulum

The equation (9) is represented by *eqs* and equilibrium points are obtained doing *eqs* = 0, for this purpose the next code is used:

```

eqs={x2[t], aux01, x4[t], aux02}/. {x1[t]->x1, x2[t]->x2, x3[t]->x3, x4[t]->x4}/. {x2->0, x4->0}
Solve[eqs[[2]]==0, x1]{{x1->-3.13649}, {x1->3.13649}, {x1->0.005102}, {x1->0.005102}}
Solve[eqs[[4]]==0, x1]{{x1->0.0102048}, {x1->1.56059}}

```

one equilibrium point correspond to (0, 0, 0, 0). Additionally another two points are determined.

4.2 Equilibrium points of two degree of freedom satellite

With the satellite mass M normalized to 1 in equation (12), it can be verified that the rotation of the satellite at constant radial distance $d = r[t]$ and a constant angular velocity $\frac{d\theta}{dt} = w$ ($\alpha(t) = wt$) represents a particular nominal solution.

5. MODEL LINEARIZATION

Different techniques and approaches can be applied on the nonlinear models to obtain their equivalent linear. The most common method corresponds to Taylor's series expansion, quite effective for small variations in relation to equilibrium point [5], [8]. To follow a summary of technique is presented.

5.1 Expansion by Taylor's series

The Taylor's series is defined as:

$$f(x) = \sum_{k=0}^{\infty} \frac{1}{k!} \left. \frac{d^k f(x)}{dx^k} \right|_{x_0} \quad (13)$$

where, x_0 corresponds to equilibrium point. Expanding the series to (x_{eq}, u_{eq}) and defining the little variations of state vector and input as:

$$\Delta x = x - x_{eq} \quad \Delta u = u - u_{eq} \quad (14)$$

The right side of equation (13) can be expanded in the following form:

$$f(x, u) = f(x_{eq}, u_{eq}) + \left. \left(\frac{\partial f}{\partial x} \right) \right|_{x_{eq}} \Delta x + \left. \left(\frac{\partial f}{\partial u} \right) \right|_{u_{eq}} \Delta u + \text{h.o.t} \quad (15)$$

High-order terms **h.o.t** contain at least quadratic quantities of Δx and Δu . Since Δx and Δu are small their squares are even smaller, and hence the **h.o.t** can be neglected. With this consideration the next approximation is obtained.

$$\Delta \dot{x} = \left(\frac{\partial f}{\partial x} \right) \Big|_{\substack{x_{eq} \\ u_{eq}}} \Delta x + \left(\frac{\partial f}{\partial u} \right) \Big|_{\substack{x_{eq} \\ u_{eq}}} \Delta u \quad (16)$$

The partial derivatives of (16), correspond to *Jacobian* matrices represented by $A^{n \times n}$ and $B^{n \times r}$, where n define the number of state variables and r the inputs number.

Finally doing $\Delta u = u_\delta$ and $\Delta x = x_\delta$, the linearized equation (16) has de form:

$$\dot{x}_\delta = Ax_\delta + Bu_\delta \quad (17)$$

With this considerations the linear models can be obtained.

5.2 Linear model of inverted pendulum

With the next code the linear model is obtained:

```
A = 0 IdentityMatrix[4]
Do[A[[i, j]] = FullSimplify
[ $\partial_{x[[j]]} eqs[[i]]$ , {j, 1, 4}, {i, 1, 4}]
MatrixForm[A]
Ae = FullSimplify[A/.{x1- > 0, x2- > 0}]
B = {{0}, {0}, {0}, {0}}
F = {u} Do[B[[i, j]] = FullSimplify
[ $\partial_{u[[j]]} eqs[[j]]$ , {j, 1, 1}, {i, 1, 4}]
MatrixForm[B]
Be = FullSimplify[B/.{x1- > 0, x2- > 0}]
```

The *Jacobian* matrices A and B are given by:

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{g(M_1+M_2)}{lM_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{gM_2}{M_1} & 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ -\frac{1}{lM_1} \\ 0 \\ \frac{1}{M_1} \end{pmatrix} u \quad (18)$$

5.3 Linear model of two degree of freedom satellite

With the following code de linear model is obtained:

```
A = 0 IdentityMatrix[4]
Do[A[[i, j]] = FullSimplify
[ $\partial_{x[[j]]} eqs[[i]]$ , {j, 1, 4}, {i, 1, 4}]
MatrixForm[A]
Ae = FullSimplify[A/.{x1- > d, x2- > 0, x3- > wt, x4- > w, k- > w^2 d^3, u1- > 0, u2- > 0}]
B = {{00}, {00}, {00}, {00}}
F = {u1, u2} Do[B[[i, j]] = FullSimplify
[ $\partial_{F[[j]]} eqs[[j]]$ , {j, 1, 2}, {i, 1, 4}]
MatrixForm[B]
Be = FullSimplify[B/.{x1- > d, x2- > 0, x3- > wt, x4- > w, k- > w^2 d^3}]
```

The *Jacobian* matrices A and B are given by:

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 3w^2 & 0 & 0 & 2dw \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2w}{d} & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & \frac{1}{d^2} \end{pmatrix} [u_1 u_2]^T \quad (19)$$

6. NONLINEAR AND LINEAR SIMULATION

The numerical analysis and simulation are done using equations (9), (12), (18) and (19), for this purpose simulation parameters are presented in table 1.

Table 1. Simulation parameters

Parameters	Value
Integration Method	ExplicitRungeKutta
Work Precision	Machine
MaxSteps	1000

6.1 Inverted pendulum

To obtain the simulation results with nonlinear and linear models and compare both, it is necessary to write the following code:

```
x1p = x2[t]
x2p = aux01
x3p = x4[t]
x4p = aux02
x5p = 0-x3[t]
KI = -10
KR = {-158.6130,-48.3707,-25.2051,-26.7648};
control=-KR.{x1[t],x2[t],x3[t],x4[t]}+KI*x5[t];
eqs={x1p,x2p,x3p,x4p,x5p} /.{u->control,g-> 9.8,1->1, M2->1,M1->1}

z1p = z2[t]
z2p = -u/1M1+(g(M1+m2)z1[t])/(1M1)
z3p = z4[t]
z4p = u/M1-gM2z1[t]/M1
z5p = 0-z3[t]
KI = -10
ctkr=-KR.{z1[t],z2[t],z3[t],z4[t]}+KI*z5[t];
eqL={z1p,z2p,z3p,z4p,z5p} /.{u->control,g-> 9.8,1->1, M2->1,M1->1}
```

Numerical solution is written as:

```
tf = 2 pi
sol = NDSolve[x1'[t]=eqs[[1]], x2'[t]=eqs[[2]],x3'[t]=eqs[[3]], x4'[t]=eqs[[4]],x5'[t]=eqs[[5]],
x1[0]==0.5,x2[0]==0,x3[0]==0.5,x4[0]==0,x5[0]==0},{x1,x2,x3,x4,x5},{t,0,tf},
WorkPrecision-> MachinePrecision,Method ->ExplicitRungeKutta,MaxSteps -> 1000]

solin =NDSolve[z1'[t]=eqL[[1]], z2'[t]=eqL[[2]],z3'[t]=eqL[[3]], z4'[t]=eqL[[4]],z5'[t]=eqL[[5]],
z1[0]==0.5,z2[0]==0,z3[0]==0.5,z4[0]==0,z5[0]==0},{z1,z2,z3,z4,z5},{t,0,tf},
WorkPrecision-> MachinePrecision,Method ->ExplicitRungeKutta, MaxSteps -> 1000]
```

Finally the results can be plotted as:

```
Plot[{158.613 z1[t]+48.3707 z2[t]+25.2051 z3[t] + 26.7648 z4[t]-10z5[t]/.solin},
{t,0,tf},AxesLabel -> {Sec,Magnitude},PlotStyle -> Hue[0.7]];
Plot[{z1[t]/.solin[[1, 1]],x1[t]/.sol[[1, 1]]},
{t,0,tf},AxesLabel -> {Sec,Magnitude},PlotStyle -> {GrayLevel[0],Dashing[{.03]}}];
Plot[{z2[t]/.solin[[1, 2]], x2[t]/.sol[[1, 2]]},
{t,0,tf},AxesLabel -> {Sec,Magnitude},PlotStyle -> {GrayLevel[0],Dashing[{.03]}}];
Plot[{z3[t]/.solin[[1, 3]], x3[t] /.sol[[1, 3]]},
{t,0,tf},AxesLabel -> {Sec,Magnitude},PlotStyle -> {GrayLevel[0],Dashing[{.03]}}];
Plot[{z4[t]/.solin[[1, 4]], x4[t]/.sol[[1, 4]]},
{t,0,tf},AxesLabel -> {Sec,Magnitude},PlotStyle -> {GrayLevel[0],Dashing[{.03]}}];
```

The obtained results allow to observe the difference between the nonlinear and linear models. This difference is minimal considering small variations at equilibrium point. At the same time it is possible to modify the simulation parameters such as: initial conditions, pendulum parameters or integration method. Some of these characteristics allows verify the matrix conditioning for example or another issues from numerical analysis.

6.2 Two degree of freedom satellite

In this case, the code is:

```
KI = -1.{-0.7228, 6.8761}, {-0.3090, -3.2165}}
KR = {{6.3186, 4.5003, -0.0688, -1.8195}, {1.4525, -0.3639, 0.0322,2.1400}};
{Aux1, Aux2} = KI.{x5[t], x6[t]}
{Aux3, Aux4} = -KR.{x1[t], x2[t], x3[t], x4[t]}
{cktra, cktrb} = {Aux1, Aux2} + {Aux3, Aux4}
```

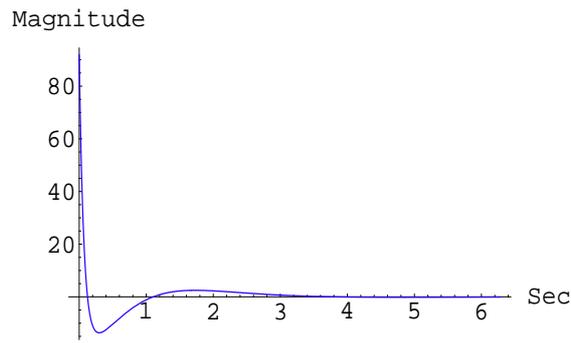


Figure 3. Variable u

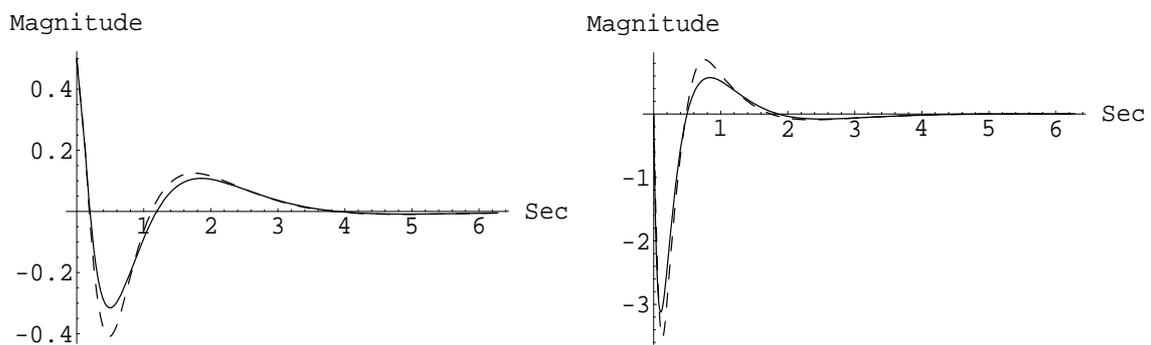


Figure 4. Variables θ and $\dot{\theta}$: nonlinear (dashed line) and linear (solid line) models

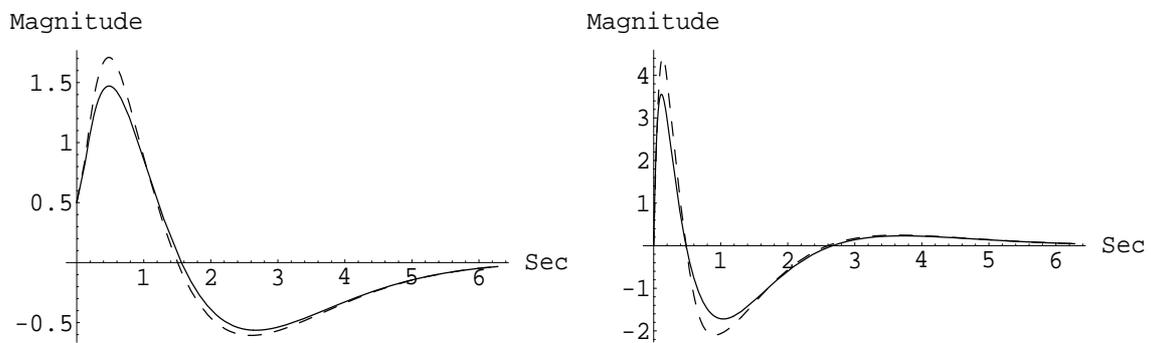


Figure 5. Variables x and \dot{x} : nonlinear (dashed line) and linear (solid line) models

```
x1p = x2[t]
x2p = u1 + 3 w^2 x1[t] + 2 d w x4[t]
x3p = x4[t]
x4p = u2/d^2 - 2 w x2[t] /d
x5p = pi/6-x1[t]
x6p = pi/6-x4[t]
eqL = FullSimplify[{x1p[[1]],x2p[[1]],x3p[[1]],x4p[[1]],x5p[[1]],x6p[[1]]}/.{d-> 1, w -> 1, u1 -> cktra, u2 -> 0}
```

```
{Aux5, Aux6} = KI.{z5[t], z6[t]}
{Aux7, Aux8} = -KR.{z1[t], z2[t], z3[t], z4[t]}
{ca, cb} = {Aux5, Aux6} + {Aux7, Aux8}
```

```
z1p = z2[t]
z2p = u1 - k/z1[t]^2+ z1[t] z4[t]^2
z3p = z4[t]
z4p = u2/z1[t]^2 - 2 z1[t] z2[t] z4[t]/z1[t]^2
z5p = pi/6-z1[t]
z6p = pi/6-z4[t]
eqs = FullSimplify[{x1p[[1]],x2p[[1]],x3p[[1]],x4p[[1]],x5p[[1]],x6p[[1]]}/.{k -> 1, u1 -> ca, u2 -> 0}
```

Numerical solution is written as:

```
tf = 6 pi
solin = NDSolve[x1'[t]=eqL[[1]], x2'[t]=eqL[[2]], x3'[t]=eqL[[3]], x4'[t]=eqL[[4]], x5'[t]=eqL[[5]], x6'[t]=eqL[[6]]
x1[0]==1, x2[0]==0, x3[0]==0, x4[0]==1, x5[0]==pi-1, x6[0]==pi-1}, {x1, x2, x3, x4, x5, x6}, {t, 0, tf}],
Method -> ExplicitRungeKutta, WorkPrecision->MachinePrecision, MaxSteps -> 1000]

sol = NDSolve[z1'[t]=eqt[[1]], z2'[t]=eqt[[2]], z3'[t]=eqt[[3]], z4'[t]=eqt[[4]], z5'[t]=eqt[[5]], z6'[t]=eqt[[6]]
z1[0]==1, z2[0]==0, z3[0]==0, z4[0]==1, z5[0]==pi-1, z6[0]==pi-1}, {z1, z2, z3, z4, z5, z6}, {t, 0, tf}],
Method -> ExplicitRungeKutta, WorkPrecision->MachinePrecision, MaxSteps -> 1000]
```

Finally the results can be plotted as:

```
Plot[{-6.3186 x1[t] - 4.5003 x2[t] + 0.0688 x3[t] + 1.8195 x4[t] + 0.7228 x5[t] - 6.8761 x6[t] /. solin},
{t, 0, tf}, AxesLabel -> {ut, Magnitude}, PlotStyle -> Hue[0.6], PlotRange -> All];
Plot[{x1[t] /. solin[[1, 1]], z1[t] /. sol[[1, 1]]}, {t, 0, tf},
AxesLabel -> {ut, Magnitude}, PlotStyle -> {GrayLevel[0], Dashing[{.03]}], PlotRange -> All];
Plot[{x2[t] /. solin[[1, 2]], z2[t] /. sol[[1, 2]]}, {t, 0, tf},
AxesLabel -> {ut, Magnitude}, PlotStyle -> {GrayLevel[0], Dashing[{.03]}], PlotRange -> All];
Plot[{x3[t] /. solin[[1, 3]], z3[t] /. sol[[1, 3]]}, {t, 0, tf},
AxesLabel -> {ut, Magnitude}, PlotStyle -> {GrayLevel[0], Dashing[{.03]}], PlotRange -> All];
Plot[{x4[t] /. solin[[1, 4]], z4[t] /. sol[[1, 4]]}, {t, 0, tf},
AxesLabel -> {ut, Magnitude}, PlotStyle -> {GrayLevel[0], Dashing[{.03]}], PlotRange -> All];
```

where **ut** correspond to *time unit*.

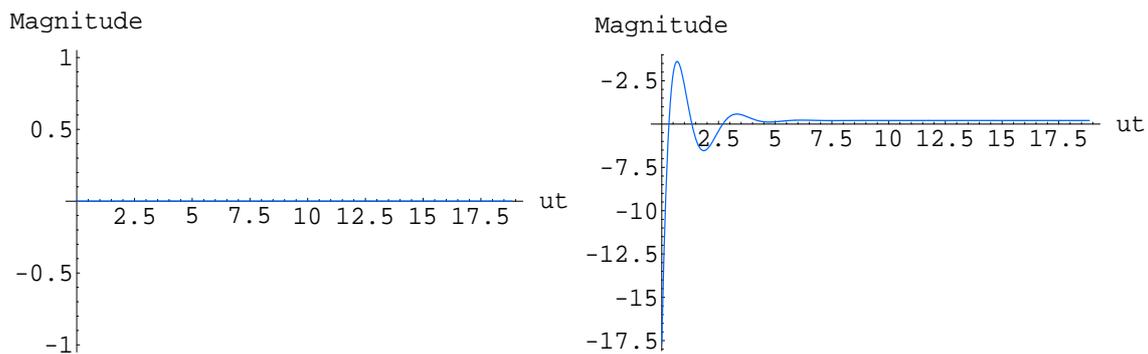


Figure 6. Variables u_1 and u_2

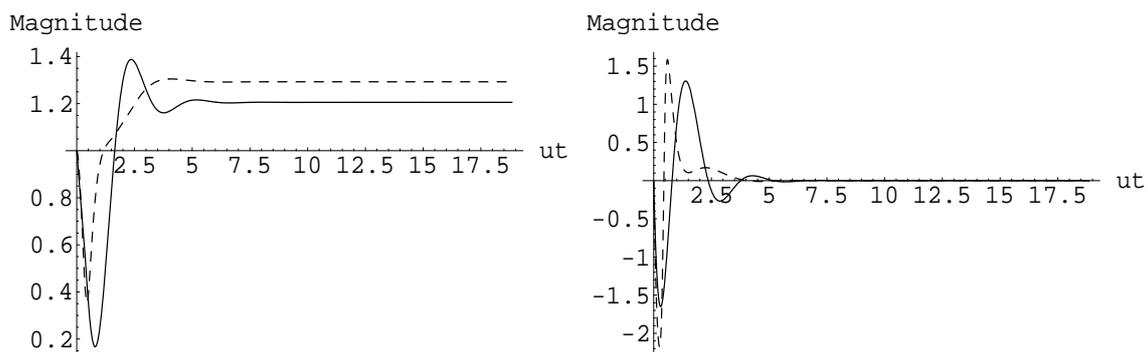


Figure 7. Variables r and \dot{r} : nonlinear (dashed line) and linear (solid line) models

From the obtained results, it is possible to observe that the difference between the nonlinear and linear model in this case is greater and not only is presented in transitory state but in the steady state also. Additionally is possible to verified that the variation of α confirm a constant angular velocity. With these examples is possible to confirm the flexibility and robustness that Mathematica® offers as algebraic tool.

7. CONCLUSIONS

In order to reach the objective proposed in this work, mathematica® has been used as didactic help to solve Mechanical and Aerospace systems such as the inverted pendulum and two degree of freedom satellite studied in *Modelling and*

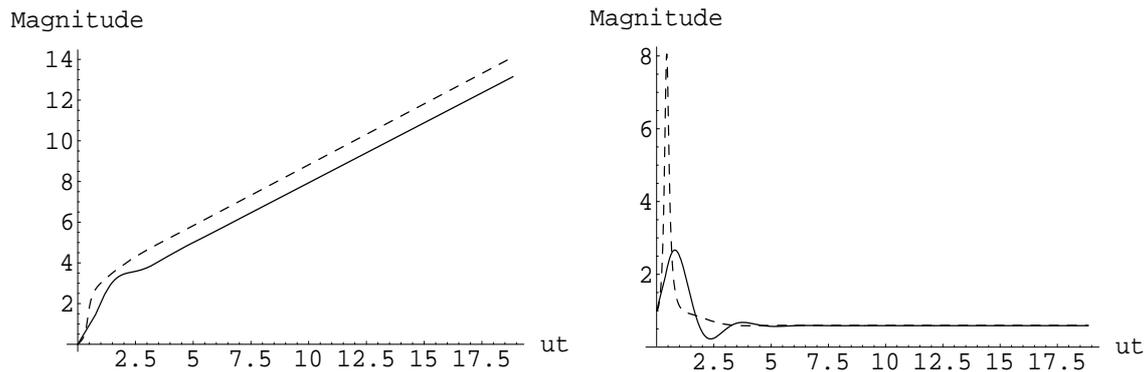


Figure 8. Variables α and $\dot{\alpha}$: nonlinear (dashed line) and linear (solid line) models

Simulation of Systems Using Mathematica course. Its application was extended to obtain nonlinear models, equilibrium point's determination, linear models derivation, numerical solution and simulation.

The presented methodology has allowed the academic improvement in exact sciences and engineering of some universities in Bolivia, exploring the robustness and versatility that mathematica® offers.

Finally to give continuity to this effort, other courses are incorporating Mathematica® in their analytical contents.

8. ACKNOWLEDGEMENTS

Authors would like to thanks to **EMBRAER** in Brazil and **San Simon University** in Bolivia.

9. REFERENCES

- Hans C. Muller Santa Cruz, 1998, "Calculo III - Ecuaciones Diferenciales", Universidad Mayor de San Simon.
- Denis da Cruz M Maciel, Delx D. Lunardi, Idelfonso Bianchi, Jose C. da Silva, 2006, "Utilização do Mathematica no Ensino de Antenas de Microfita", Revista de Ensino de Engenharia, Vol.25, No. 1, pp. 41-48.
- Jose Manuel Gutierrez, Andres Iglesias, 1998, "Mathematica package for analysis and control of chaos in nonlinear systems", Computers in Physics, Vol.12, No. 6, pp. 608-618.
- Wolfram, 2008, <http://www.wolfram.com>.
- A. J. Jordan, 2006, "Linearization of nonlinear state equation", Bulletin of the polish Academy of Sciences, Vol.54, No. 1, pp. 63-73.
- Hassan K. Khalil, 1996, Nonlinear Systems, Prentice Hall.
- B. Palancz, L. Kovacs, B. Benyo and Z. Benyo, 2004, "A fully Symbolic Design and Modelling of Nonlinear Glucose Control System Profesional Suite of Mathematica", Acta Physiologica Hungarica, Vol.91, No. 2, pp. 147-156.
- Katsuhito Ogata, 1997, "Engenharia de Controle Moderno", Prentice Hall.
- Francisco J. Triveño Vargas, 2006, "Control a Estructura Variable y Modos Deslizantes", Universidad Mayor de San Simon.
- Francisco J. Triveño Vargas, 2007, "Modelado, Simulación y Control de Sistemas", Universidad del Valle.
- Francisco J. Triveño Vargas, 2008, "Modelado y Simulación de Sistemas Empleando Matlab-Simulink y Mathematica", Universidad Tecnica de Oruro.
- William A. Wolovich, 1994, "Automatic Control Systems: Basic Analysis and Design", Oxford University Press.

10. Responsibility notice

The authors are the only responsible for the printed material included in this paper.